

TECHNISCHE UNIVERSITÄT MÜNCHEN

Professur für Mechanik auf Höchstleistungsrechnern

A Monolithic Solver for Fluid-Structure Interaction with Adaptive Time Stepping and a Hybrid Preconditioner

Matthias Mayr

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. ir. Daniel J. Rixen

Prüfer der Dissertation: 1. Univ.-Prof. Dr.-Ing. Michael W. Gee
2. Univ.-Prof. Dr.-Ing. Wolfgang A. Wall
3. MER Dr. ès Sc. Simone Deparis

Ecole Polytechnique Fédérale de Lausanne, Switzerland

Die Dissertation wurde am 04. Juli 2016 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 18. November 2016 angenommen.

Abstract

Many scientific and engineering problems involve the coupling of several physical effects or models. One class of coupled problems, that has interested many scientists and engineers for decades, is the interaction of fluid flow and solid bodies. Possible applications range from aeroelasticity over civil engineering to biomedical problems like the analysis of blood flow in the human vascular system. Application-wise and from a numerical point of view, the interaction of an incompressible fluid flow with solid bodies undergoing finite deformation is of particular interest. Although many researchers addressed this class of problems for decades, solving fluid-structure interaction (FSI) problems numerically still poses a challenging task.

In this thesis, a finite-element-based monolithic framework for the solution of FSI problems is presented. It is particularly tailored to the demands that arise when an incompressible fluid flow interacts with solid bodies undergoing finite deformation. Several key aspects are addressed, in particular questions related to the formulation of the monolithic solver, non-matching interface discretizations, time integration, and preconditioning of the system of linear equations. For complex practical applications, computational effort has been reduced by 75% while at the same time guaranteeing a user-given level of accuracy.

As FSI problems almost exclusively exhibit transient behavior, temporal discretization and time integration play an essential role when it comes to accuracy and efficiency of the solver. Hence, the proposed formulation allows for choosing the time integration schemes in the solid and the fluid field freely and independently from each other by introducing a temporal interpolation of the interface traction fields in a temporally consistent manner. To the author's best knowledge, the present work poses the first development of an adaptive time stepping scheme for monolithic FSI solvers where control over the temporal accuracy is enabled by adapting the time step size based on *a posteriori* error estimation. Therefore, the fluid field, the structure field, and the fluid-structure interface are taken into account.

The nonlinearity of the coupled problem is treated by a NEWTON–KRYLOV method. For an efficient solution process, preconditioning techniques tailored to the FSI problem are crucial. To further leverage efficiency and robustness of existing FSI-specific physics-based block preconditioners based on algebraic multigrid methods, a novel *hybrid additive/multiplicative* SCHWARZ preconditioner is proposed, that combines the existing physics-based block preconditioners with an additional additive SCHWARZ preconditioner. The latter one is specifically designed to tackle error accumulation at the fluid-structure interface, that stems from the physics-based block preconditioning. Therefore, an overlapping domain decomposition with subdomains, that purposely span across the interface, is generated. Subdomain solvers, that are insensitive to the separation of physical fields by the interface, are used to smooth the solution across the interface.

Various numerical examples demonstrate the outstanding robustness and efficiency of the proposed monolithic solver. The adaptive time stepping algorithm guarantees temporal accuracy up to a user-given level while simultaneously computational cost is limited to the amount necessary. Significant savings of up to 75% have been demonstrated. Furthermore, all described preconditioning approaches are compared to each other. Remarkable savings of linear iterations and of linear solver time can be achieved by the novel hybrid preconditioner. Finally, the pulsatile blood flow through a patient-specific abdominal aortic aneurysm is simulated, which highlights the huge potential of the proposed solver framework in real-world applications with complex geometries.

Zusammenfassung

Viele Fragestellungen in Forschung und Entwicklung beinhalten die Kopplung mehrerer physikalischer Effekte oder Modelle. Eine Klasse gekoppelter Probleme, welche seit Jahrzehnten im Fokus von Wissenschaftlern und Ingenieuren steht, umfasst die Wechselwirkung von Fluidströmungen und deformierbaren Festkörpern. Zahlreiche Anwendungen aus Bereichen wie der Aeroelastizität oder dem Bauingenieurwesen bis hin zu biomedizinischen Fragestellungen bezüglich des Blutflusses im menschlichen Körper fallen in diese Problemklasse. Aus Anwendungs- wie auch aus numerischer Sicht ist die Wechselwirkung inkompressibler Strömungen mit Festkörpern, welche großen Deformationen unterliegen, von besonderem Interesse. Obwohl seit Jahrzehnten Gegenstand äußerst intensiver Forschung, ist die numerische Behandlung von Problemen der Fluid-Struktur-Interaktion (FSI) noch immer eine ausgenommen anspruchsvolle Aufgabe.

Basierend auf der Methode der Finiten Elemente wird in dieser Arbeit ein monolithisches Verfahren zur Lösung solcher Fluid-Struktur-Interaktionsprobleme vorgestellt. Dieses wird auf die Besonderheiten der Interaktion von inkompressiblen Strömungen mit deformierbaren Festkörpern zugeschnitten. Insbesondere werden entscheidende Aspekte betreffend die Formulierung des monolithischen Lösers, die Herausforderung nicht-passender Netze am Interface, sowie Aspekte der Zeitintegration und der Vorkonditionierung des linearen Gleichungssystems behandelt. Die Rechenzeit für praktische Anwendungen mit komplexen Geometrien konnte dadurch um bis zu 75% reduziert werden.

Da Probleme der Fluid-Struktur-Interaktion fast immer zeitabhängig sind, kommen der Zeitdiskretisierung und -integration eine entscheidende Rolle zu, sobald Genauigkeit und Effizienz des Lösers wichtig werden. Daher erlaubt die vorgeschlagene Formulierung eine freie und gänzlich unabhängige Wahl der Zeitintegrationsverfahren für das Struktur- und Fluidfeld. Dies wird durch eine Interpolation der Interfacekräfte in der Zeit in einer zeitlich konsistenten Art und Weise ermöglicht. Des Weiteren stellt diese Arbeit die erste Beschreibung einer adaptiven Zeitschrittweitensteuerung für monolithische FSI-Löser dar, in der eine *a posteriori* Fehlerschätzung die Kontrolle der Genauigkeit der Zeitdiskretisierung unter Berücksichtigung der Beiträge des Fluid- und des Strukturgebiets wie auch des Interfaces zwischen den beiden Feldern ermöglicht.

Der Nichtlinearität des betrachteten Problems wird durch ein NEWTON-KRYLOV-Verfahren Rechnung getragen. Mit dem angestrebten Ziel eines effizienten Lösungsverfahrens kommen Vorkonditionierungstechniken, welche auf die Besonderheiten der FSI zugeschnitten sind, eine zentrale Bedeutung zu. Zwei existierende Vorkonditionierungsverfahren, die sich die physikalisch motivierte Blockstruktur des Gleichungssystems zu Nutze machen und auf algebraischen Mehrgittermethoden basieren, werden eingeführt. Zur weiteren Steigerung der Effizienz und Robustheit wird ein neuartiges hybrides Vorkonditionierungsverfahren vorgeschlagen, welches additive und multiplikative SCHWARZ'sche Methoden miteinander kombiniert. Der Einsatz der additiven SCHWARZ'schen Methode soll dabei zielgerichtet die Anhäufung des Fehlers am Interface beseitigen, welche im Rahmen der physikalisch motivierten Block-Vorkonditionierung entsteht. Dazu wird eine überlappende Gebietszerlegung erzeugt, in der sich Teilgebiete über das Interface hinweg erstrecken. Durch die Anwendung bestimmter Lösungsverfahren auf diesen Teilgebieten, welche gegenüber der unterschiedlichen Physik der beiden Felder insensitiv sein müssen, wird die Lösung am Interface geglättet.

Eine Vielzahl anspruchsvoller numerischer Beispiele demonstriert die herausragende Robustheit und Effizienz des vorgeschlagenen monolithischen Lösungsverfahrens. Die adaptive Zeitschrittweitensteuerung ermöglicht dabei eine garantierte Kontrolle der Genauigkeit der Zeitintegration bei gleichzeitiger Begrenzung der Rechenkosten auf das minimal notwendige Maß. Dadurch können erhebliche Einsparungen von bis zu 75% erreicht werden. Des Weiteren werden die vorgestellten Vorkonditionierungsverfahren miteinander verglichen. Durch die Anwendung der hybriden Vorkonditionierung können sowohl die Anzahl der Iterationen des linearen Gleichungslösers als auch die Löserzeit erheblich reduziert werden. Abschließend wird der pulsatile Blutfluß durch eine patientenspezifische Geometrie eines abdominalen Aortenaneurysmas simuliert, um die Anwendbarkeit des vorgestellten Lösungsverfahrens auf komplexe Geometrien und realitätsnahe Problemstellungen zu demonstrieren.

Contents

1	Introduction	1
2	Governing Equations of Fluid-Structure Interaction	5
2.1	Fluid Field on a Deforming Domain	5
2.2	Structure Field	7
2.3	Coupling Conditions at the Fluid-Structure Interface	8
3	Finite Element Formulation and Monolithic System of Equations	11
3.1	Weak Form of the FSI Problem	11
3.2	Discretization of the FSI Problem	13
3.2.1	Fluid field	15
3.2.2	Structure field	17
3.2.3	LAGRANGE Multiplier Field	18
3.2.4	Fluid-Structure Interface	19
3.3	Monolithic System of Equations	27
3.3.1	Fluid-handled Interface Motion	28
3.3.2	Structure-handled Interface Motion	31
3.4	Discussion of the Formulation using a Pseudo One-Dimensional FSI Example	33
3.4.1	Temporal Convergence Study	33
3.4.2	LAGRANGE Multiplier Field and Choice of Master and Slave Side	35
3.5	Summary	38
4	An Adaptive Time Stepping Procedure for Monolithic FSI Solvers	41
4.1	Fundamentals	42
4.1.1	Preliminaries	43
4.1.2	Errors and the Order of Convergence	43
4.1.3	Adapting the Time Step Size	45
4.1.4	Estimation of the Local Discretization Error	48
4.2	Application to Monolithic Fluid-Structure Interaction Solvers	51
4.2.1	Estimation of the Local Discretization Error	52
4.2.2	Adapting the Time Step Size	53
4.2.3	Practical Considerations	54
4.3	Numerical Examples	58
4.3.1	Buckling of a Solid Cylindrical Shell Submerged into Fluid	58
4.3.2	An Elastic Wall in a Channel Flow	67
4.3.3	Snap-Through of a Gasket	73
4.4	Summary	80

5	Solution Methods for the Monolithic System of Equations	81
5.1	Physics-Based Multilevel Block Preconditioning for FSI	82
5.1.1	Algebraic Multigrid Methods in a Nutshell	82
5.1.2	A Block-Iterative Approach with an Internal Algebraic Multigrid Preconditioner	84
5.1.3	A Fully Coupled Algebraic Multigrid Preconditioner	85
5.2	Hybrid Additive/Multiplicative SCHWARZ Preconditioning for Surface-Coupled Problems	89
5.2.1	Overlapping Domain Decomposition and SCHWARZ Methods	90
5.2.2	Requirements for the Domain Decomposition	90
5.2.3	Setup and Application of the Preconditioner	94
5.2.4	Proof of Concept and Demonstration of Improved Error Reduction	96
5.3	Practical Aspects of Monolithic Linear and Nonlinear Solvers for Coupled Problems	97
5.3.1	Convergence Check for the Nonlinear Solver	99
5.3.2	Convergence Check for the Linear Solver	100
5.4	Summary	102
6	Numerical Examples	103
6.1	Pressure Wave through an Elastic Tube	103
6.1.1	On the Influence of Time Integration Schemes	105
6.1.2	Discussion of FSI Preconditioners	109
6.2	Pulsatile Blood Flow through a Patient-specific Abdominal Aortic Aneurysm	116
7	Concluding Remarks and Outlook	125
A	Mesh Motion Algorithms for the ALE Grid	129
A.1	Requirements	129
A.2	Mesh Motion Algorithms	130
A.2.1	LAPLACEian Smoothing	130
A.2.2	Springs Model	131
A.2.3	Elasticity Model	133
A.3	A Comparison of ALE Mesh Motion Algorithms	133
B	Hardware and Computational Resources	139
	Bibliography	141

List of Figures

2.1	Sketch of the FSI problem and the involved domains and boundaries	6
3.1	Illustration of discrete kinematic interface coupling conditions	20
3.2	Temporal interpolation of the interface traction field	25
3.3	Geometry of a pseudo one-dimensional FSI problem with analytical solution . .	33
3.4	Temporal convergence study for pseudo one-dimensional example	36
3.5	DIRICHLET boundary conditions for a pseudo one-dimensional example	37
3.6	Comparison of LAGRANGE multiplier fields for pseudo one-dimensional example	39
4.1	Illustration of local and global error	44
4.2	Adaptive time stepping algorithm based on <i>a posteriori</i> error estimation	47
4.3	Error estimation by RICHARDSON extrapolation	49
4.4	Rectangular fluid domain with submerged thin-walled solid cylinder segment .	59
4.5	Mesh for solid cylindrical shell under fluid loading	60
4.6	Snapshots of the solution of a solid cylindrical shell under fluid loading	62
4.7	Solution at the location A of a solid cylindrical shell under fluid loading	63
4.8	Monitoring of the estimated local error in a solid cylindrical shell under fluid loading	64
4.9	Averaging of increasing time step sizes for a solid cylindrical shell under fluid loading	65
4.10	Evolution of time step size for solid cylindrical shell under fluid loading	66
4.11	Geometry and boundary conditions of an elastic wall in a channel flow	68
4.12	Inflow velocity and tip displacement of an elastic wall in a channel flow	70
4.13	State of maximum deformation of an elastic wall in a channel flow	71
4.14	Evolution of time step size Δt for an elastic wall in a channel flow	71
4.15	Necessary number of time step repetitions for an elastic wall in a channel flow .	72
4.16	Geometry and boundary conditions of gasket-like example	73
4.17	Views including the mesh of a gasket-like example	74
4.18	Temporal evolution of the inflow velocity for a gasket-like example	75
4.19	Snapshots of the solution of a gasket-like example using a fine mesh	77
4.20	Displacement and velocity in a gasket-like example	78
4.21	Evolution of the time step size for a gasket-like example	78
4.22	Influence of the mesh on the evolution of the time step size for a gasket-like example	79
4.23	Energies in gasket-like example	79
5.1	Sketch of a 3-level multigrid V-cycle	83
5.2	Block GAUSS–SEIDEL preconditioner with AMG-based approximate block in- verses	85

5.3	Fully coupled AMG preconditioner for FSI	86
5.4	Fully coupled multigrid hierarchy with different numbers of levels per field	88
5.5	Physics-based overlapping domain decomposition of a FSI problem	91
5.6	Overlapping domain decomposition based on a monolithic graph of a FSI problem	92
5.7	Overlapping domain decompositions for FSI preconditioners	93
5.8	Comparison of the error reduction by BGS(LU) and H-BGS(LU)	98
5.9	Screen output of nonlinear convergence test for monolithic FSI in Baci	101
6.1	Geometry of pressure wave through an elastic tube	104
6.2	Exemplary mesh for pressure wave example	106
6.3	Displacement and pressure field in pressure wave	106
6.4	Various combinations of time integration schemes for the pressure wave example	108
6.5	Iteration counts and timings for physics-based block preconditioners	111
6.6	Domain decompositions for physics-based and hybrid preconditioners	113
6.7	Iteration counts and timings for hybrid preconditioners on 256 cores	115
6.8	Geometry of a patient-specific Abdominal Aortic Aneurysm	118
6.9	Physiological time curves for inflow velocity and outflow pressure of AAA	119
6.10	Mesh of a patient-specific Abdominal Aortic Aneurysm	120
6.11	Snapshots of the solution of a pulsatile blood flow through an AAA	121
6.12	Displacement of the artery throughout the heart beat	122
6.13	Evolution of time step size for pulsatile flow through AAA	123
A.1	Springs analogy in two dimensions for ALE mesh motion	132
A.2	Configuration of rectilinear and torsional springs in a triangular ALE element	132
A.3	Square domain with rotating rectangular slot for ALE mesh motion tests	136
A.4	Final deformations of different ALE mesh motion schemes in a rotation test	137

List of Tables

3.1	Material parameters of a pseudo one-dimensional FSI example	37
4.1	Error norms for fluid-handled and structure-handled interface motion	53
4.2	Material parameters for solid cylindrical shell under fluid loading	59
4.3	Averaging strategies for solid cylindrical shell under fluid loading	63
4.4	Computational savings for a solid cylindrical shell under fluid loading	67
4.5	Material parameters for elastic wall in channel flow	67
4.6	Computational savings for an elastic wall in a channel flow	72
4.7	Material parameters for gasket-like example	74
4.8	Mesh overview for a gasket-like example	75
6.1	Material parameters for the pressure wave example	104
6.2	Matching meshes for the pressure wave example	105
6.3	Settings for multigrid hierarchies for pressure wave example	110
6.4	Averages of iteration counts and solver time per time step	112
6.5	Settings for multigrid hierarchies embedded in the hybrid preconditioner	114
6.6	Averages of iteration counts and time measurements per time step	117
6.7	Settings for fully-coupled AMG-based preconditioner for solution of AAA	120
A.1	Methods involved in the comparison of ALE mesh motion algorithms	135

1 Introduction

Coupled physical phenomena, that include interactions between various physical fields and effects, are omnipresent in our world. They play an important role not only in traditional engineering applications but also in various types of biomechanical and biomedical problems. One prominent representative of coupled problems is the interaction of a fluid flow with solid bodies undergoing finite deformation, commonly referred to as *fluid-structure interaction (FSI)*. Engineering applications range from the design of moving objects like an aircraft, where aeroelasticity [70] is of huge interest, to the analysis of large building structures in civil engineering [257]. In biomechanics, especially the analysis of blood flow in the human vascular system, i.e. the interaction of the incompressible pulsatile blood flow with the arterial wall, is of great interest to many scientists and medical doctors. Several particular phenomena have been in the focus of research during the past years, among them the simulation of the pumping motion of the human heart [164], the assessment of rupture risk of abdominal aortic aneurysms, the blood flow through healthy [12, 41, 230, 258] as well as stented arteries [171], or the development and progression of atherosclerosis [140]. Numerical simulations have become a key component for the understanding of these phenomena but also to assess their risk and to develop treatment strategies. Although many researchers have addressed this class of problems for decades, solving such problems robustly, accurately, and efficiently still poses a great challenge for numerical algorithms. One very promising approach are monolithic solvers for FSI problems as they were found to be superior in such applications [143].

In this thesis, some of the difficulties are addressed, namely those arising from complex geometries, the choice of time integration schemes, accuracy requirements of the time discretization, and questions related to the robustness and the efficiency of solving FSI problems monolithically in a parallel computing environment. In practical applications with complex geometries, a speed up of up to 75% could be achieved while concurrently guaranteeing a user-given level of accuracy.

A Brief Classification of Methods and Algorithms for FSI

In the numerical solution of FSI problems, two basic types of solution approaches are distinguished, namely *monolithic* and *partitioned* schemes. In partitioned schemes, a sequence of single field solutions is required, where coupling information is exchanged between the fluid and the structure field. They have been and nowadays are still very popular for several reasons. On the one hand, solving the solid and the fluid problem one after the other is possible even in case of memory limitations since only one of the problems needs to fit into memory at a time. On the other hand, it is often claimed that existing codes specifically tailored to the solution of solid or fluid problems can be coupled together quite easily since no invasive changes in the solid or the fluid solver are necessary. This is of particular importance if well-established legacy codes or commercial codes are to be included into the coupling algorithm. Finally, the realization of

the coupling procedures is straightforward. Complicated tasks like solving and preconditioning of large systems of linear equations are left to the single field codes. The exchange of coupling information is often based on a DIRICHLET–NEUMANN scheme [145, 153] or ROBIN transmission conditions [7, 8, 79, 176]. Especially in biomedical applications, partitioned schemes lack efficiency as discussed in [33, 82, 91] and demonstrated numerically in [9, 113, 143]. Acceleration techniques [51, 144] might help to alleviate these problems. A detailed overview, classification, and comparison of partitioned approaches has been given in the thesis by KÜTTLER [142].

By contrast, monolithic procedures as the one proposed by MAYR *et al.* [163] solve both the fluid and the structural equations simultaneously within one global system of nonlinear equations. For some challenging numerical problems like channels with flexible walls [112], thin-walled structures in the human respiratory or hemodynamic system [143] or for balloon-type problems like human red blood cells [139], monolithic schemes outperform partitioned procedures by far in terms of computational costs or are even the only feasible schemes to address such problems. Although the implementation of a monolithic solver requires more effort, especially for the preconditioner, great improvements in robustness and performance can be achieved. Detailed performance analyses and comparisons to partitioned schemes have been carried out by BADIA *et al.* [9], HEIL *et al.* [113], and KÜTTLER *et al.* [143], for example. A variety of preconditioners for the monolithic system of equations is available in the literature. An approach based on block-triangular approximations of the Jacobian matrix has been proposed by HEIL [112] and extended by MUDDLE *et al.* [172]. In GEE *et al.* [89], efficient preconditioners based on algebraic multigrid techniques have been developed. VERDUGO and WALL [240] extended these ideas to the monolithic coupling of an arbitrary number of physical fields. Further preconditioning strategies can be found in [10, 40, 149] and are discussed in the respective chapter of this thesis.

Further distinction of methods is based on the observer, that is used to describe the fluid flow field. When an *Arbitrary LAGRANGEan–EULERian (ALE)* description of the fluid field is employed, the fluid grid needs to be deformed to follow the motion of the fluid-structure interface. Such approaches date back to the 1970's and 1980's [18, 19, 60, 117, 125] and have since spread widely into scientific and commercial codes. Limitations of an ALE approach may come from too extensive mesh distortion or the necessity of topological changes. When solving the fluid field on a fixed EULERian grid, the main challenge is to capture the interface location. As the interface moves, it does not coincide with element edges and, thus, intersects the fluid mesh. Prominent approaches to impose the coupling conditions in case of an EULERian fluid mesh are *immersed boundary methods* [168, 181] or approaches based on the *eXtended Finite Element Method (XFEM)* [92, 245]. An overview of both ALE-based and fixed-grid approaches has been given by WALL *et al.* [244]. A combination of both methods to a hybrid fixed-grid/ALE approach is sketched by WALL *et al.* [244] and detailed in the thesis by SHAHMIRI [218]. Closely related is the differentiation between *interface-tracking* methods, where the position of the fluid-structure interface is explicitly resolved by the deformable computational grid, and *interface-capturing* methods, that use a fixed grid for the fluid field [226]. The present work falls into the category of interface-tracking methods, since the interface is explicitly meshed, the fluid field is described by an ALE observer, and the fluid grid is deformed to follow the motion of the fluid-structure interface.

Goals

This thesis aims at providing a framework for monolithic solution schemes for FSI problems where an incompressible fluid flow interacts with a solid body undergoing finite deformation. To deal with the deformation of the fluid domain, an *arbitrary* LAGRANGEan–EULERian (ALE) observer will be used for the fluid field, while the solid domain is treated in a purely LAGRANGEan description. To allow for freedom in mesh creation, the proposed algorithm shall exhibit capabilities to deal with non-matching interface discretizations. This can be achieved by enforcing the interfacial constraints with a LAGRANGE multiplier field, that is then discretized with a dual mortar method.

Questions related to time integration are of particular interest. On the one hand, it is aimed at a free and independent choice of time integration schemes in the solid and the fluid field. This allows for tailoring the time integration schemes to the needs of the solid and the fluid field without being limited by the other field. To achieve this goal, a temporal interpolation of interface traction fields needs to be incorporated into the time-discrete formulation in a temporally consistent manner. Furthermore, control over the accuracy of the time-discrete solution is desired. This can be achieved by introducing a novel adaptive time stepping scheme for monolithic FSI solvers, that modifies the time step size throughout the entire simulation to match the required level of accuracy. For efficiency reasons, such an approach will be based on *a posteriori* error estimation. Of course, the fluid and the structure field, but also the fluid-structure interface need to be taken into account for that purpose.

To allow for an efficient solution of large-scale problems, suitable preconditioning techniques are required. As the monolithic system of equations exhibits a block structure, that is closely related to the the involved solid, fluid, and ALE fields, physics-based preconditioners can be designed that exploit this particular block structure of the monolithic system matrix. Powerful approaches based on algebraic multigrid methods are available. However, it is known that physics-based block preconditioning of surface-coupled problems exhibits an accumulation of the error at the interface. To overcome this drawback, a novel *hybrid additive/multiplicative* SCHWARZ *preconditioner* shall be developed that combines the powerful multigrid performance of existing physics-based block preconditioners with an additional additive SCHWARZ preconditioner. The latter one needs to be constructed based on an overlapping domain decomposition, whose subdomains span across the fluid-structure interface on purpose. By using subdomain solvers, that are insensitive to the separation of physics by the interface, a high-quality solution can be obtained. In combination with the physics-based block preconditioners, the error accumulation at the interface can be reduced effectively such that gains in efficiency are expected.

Both the adaptive time stepping scheme as well as the proposed preconditioners will be evaluated by means of a series of numerical examples. Special focus will be put on the assessment of the accuracy and the efficiency of the proposed solution scheme. Therefore, detailed comparisons to existing approaches will be drawn. Finally, the proposed monolithic solver shall be used to simulate the pulsatile blood flow through a patient-specific abdominal aortic aneurysm to prove its applicability to complex geometries and real-world problems.

Outline

The remainder of this thesis is organized as follows: In Chapter 2, the physical model is introduced briefly by means of its governing equations.

In Chapter 3, discretization of all fields in space and time is performed. Starting from the weak form presented in Section 3.1, discrete representations of the fluid and the solid problems are shown in Section 3.2. Furthermore, discrete versions of the interface coupling conditions are derived. A detailed discussion of the treatment of non-matching grids at the fluid-structure interface by means of the dual mortar method is included and the freedom of choosing time integration schemes in the solid and the fluid field independently is incorporated. After the assembly of the monolithic system of equations, two variants of static condensation are detailed, that lead to the final systems of equations to be implemented, cf. Section 3.3. A pseudo one-dimensional example is proposed in Section 3.4 to demonstrate and discuss properties of the proposed monolithic FSI solver, among them temporal convergence rates, a physical interpretation of the LAGRANGE multiplier field at the interface, and the interplay of essential boundary conditions with the LAGRANGE multiplier field.

Temporal accuracy is addressed in Chapter 4, which details the first proposition of an adaptive time stepping scheme for monolithic FSI solvers. Theoretical background as well as strategies for *a posteriori* error estimation and adaptation of the time step size in the context of single field problems are briefly summarized in Section 4.1 and extended to monolithic FSI solvers in Section 4.2. Several examples are used to study accuracy and performance of the proposed adaptive time stepping scheme in Section 4.3.

To allow for an efficient solution of the large monolithic systems of equations, powerful preconditioning techniques are discussed in Chapter 5. After a brief review of existing physics-based FSI preconditioners based on algebraic multigrid techniques in Section 5.1, a novel hybrid additive/multiplicative SCHWARZ preconditioner for surface-coupled problems is proposed in Section 5.2. Finally, some general remarks on iterative linear and nonlinear solvers in the context of coupled multi-physics problems are given in Section 5.3.

Chapter 6 presents large-scale numerical examples. First, a pressure wave through an elastic tube is used to study features of the FSI formulation related to time integration as well as to demonstrate solver performance and efficiency of the proposed preconditioning techniques. Second, the pulsatile blood flow through a patient-specific abdominal aortic aneurysm is studied to show applicability of the proposed solver framework to complex geometries and real-world problems.

Finally, some concluding remarks and an outlook for future research are given in Chapter 7.

2 Governing Equations of Fluid-Structure Interaction

The physical model and the set of governing equations for fluid-structure interaction problems used in this thesis consists of the incompressible NAVIER–STOKES equations on a deforming domain and a nonlinear elastic solid body. This is a common approach in literature and has already been used for previous implementations in the in-house research code `Baci`, cf. the theses by KÜTTLER [142] and KLÖPPEL [137]. Hence, only a brief introduction is given. For a comprehensive presentation of continuum mechanics, the reader is referred to literature, e.g. to the monographs by GURTIN [103] or TRUESDELL and NOLL [231]. Among others, GRESHO and SANI [102] or DONEA and HUERTA [61] devoted their textbooks to fluid mechanics only, while solid mechanics is covered by BONET and WOOD [24] or HOLZAPFEL [118] for example. A sound and comprehensive introduction of the FSI problem has been given in the thesis by KLÖPPEL [137], which was used as a starting point for the present work. The physical model used in this thesis has previously been presented by MAYR *et al.* [163].

A sketch of the FSI problem is depicted in Figure 2.1. Two physical domains, namely the solid body Ω^s and the fluid field Ω^f , interact with each other through a shared surface, the *fluid-structure interface* Γ_{FSI} . Interface tractions $\underline{\mathbf{h}}_{\Gamma_{\text{FSI}}}^s$ and $\underline{\mathbf{h}}_{\Gamma_{\text{FSI}}}^f$ are exchanged at the interface. To deal with the deformation of the fluid domain, an ALE approach is used for the fluid field, while the solid is treated with a purely LAGRANGEan description.

The governing equations of the fluid field on a deforming domain are briefly summarized in Section 2.1, followed by the introduction of the solid mechanics problem in Section 2.2. Finally, Section 2.3 details the coupling conditions, that need to be satisfied at the fluid-structure interface.

2.1 Fluid Field on a Deforming Domain

The fluid field is assumed to be governed by the instationary, incompressible NAVIER–STOKES equations for a NEWTONian fluid on a deformable domain Ω^f using an ALE observer. In the following, the notation $\Omega^f \times (0, T)$ is used to indicate $\{\underline{\mathbf{x}} \in \Omega^f, t \in (0, T)\}$. Using an underline to indicate continuum vector or tensor valued quantities, the unknown deformation $\underline{\mathbf{d}}^g(\underline{\mathbf{x}}, t)$ of the deformable fluid domain Ω^f is defined by the transformation $\underline{\psi}^g$ given as

$$\underline{\mathbf{d}}^g(\underline{\mathbf{x}}, t) = \underline{\psi}^g(\underline{\mathbf{d}}_{\Gamma}^g, \underline{\mathbf{x}}, t) \quad \text{in } \Omega^f \times (0, T), \quad (2.1)$$

which extends the effect of a prescribed boundary motion $\underline{\mathbf{d}}_{\Gamma}^g$ into the interior of the fluid domain resulting in a deformation $\underline{\mathbf{d}}^g(\underline{\mathbf{x}}, t)$ of the fluid domain. The mesh deformation in the interior of the fluid domain is calculated by a mesh moving algorithm purely based on the boundary

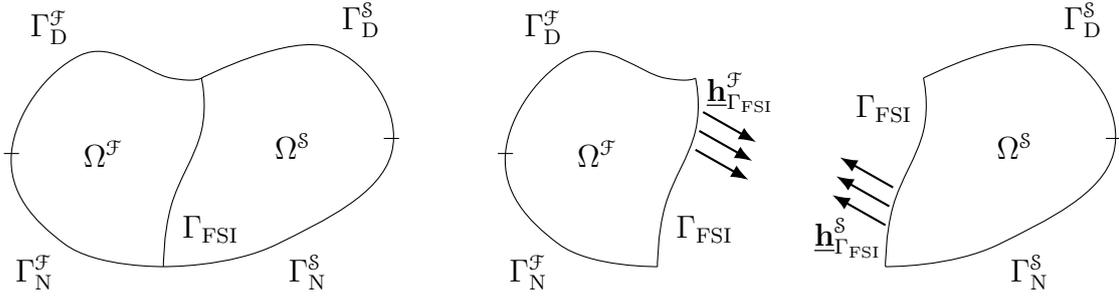


Figure 2.1: Problem statement — *Left*: The domain Ω is subdivided into a fluid domain Ω^f and a structural domain Ω^s by the fluid-structure interface Γ_{FSI} . Both subdomains are bounded by DIRICHLET boundaries Γ_D^f and Γ_D^s , NEUMANN boundaries Γ_N^f and Γ_N^s , and the common fluid-structure interface Γ_{FSI} , respectively. *Right*: At the interface, kinematic continuity as well as equilibrium of interface traction fields $\underline{h}_{\Gamma_{FSI}}^f$ and $\underline{h}_{\Gamma_{FSI}}^s$ are required.

deformation \underline{d}_Γ^g . Then, the domain velocity $\underline{u}^g(\underline{x}, t)$ is given by

$$\underline{u}^g(\underline{x}, t) = \frac{\partial \psi^g(\underline{d}_\Gamma^g, \underline{x}, t)}{\partial t} \quad \text{in } \Omega^f \times (0, T). \quad (2.2)$$

In order to prevent fluid flow across the interface, it has to match the fluid velocity $\underline{u}_{\Gamma_{FSI}}^f$ at the fluid-structure interface Γ_{FSI} , reading

$$\underline{u}_{\Gamma_{FSI}}^f = \underline{u}_{\Gamma_{FSI}}^g = \frac{\partial \underline{d}_{\Gamma_{FSI}}^g}{\partial t} \quad \text{on } \Gamma_{FSI} \times (0, T). \quad (2.3)$$

The velocity of the fluid relative to the moving background mesh is given by the ALE convective velocity $\underline{c} = \underline{u}^f - \underline{u}^g$. Using the ALE time derivative, the incompressible NAVIER–STOKES equations governing the fluid field on a deforming domain then read

$$\rho^f \frac{\partial \underline{u}^f}{\partial t} + \rho^f \underline{c} \cdot \nabla \underline{u}^f - 2\mu_{\text{dyn}}^f \nabla \cdot \underline{\underline{\epsilon}}(\underline{u}^f) + \nabla p^f = \rho^f \underline{b}^f, \quad (2.4a)$$

$$\nabla \cdot \underline{u}^f = 0, \quad (2.4b)$$

both valid in $\Omega^f \times (0, T)$, where *fluid velocity* \underline{u}^f and *dynamic fluid pressure* p^f are unknown. The *body force* is denoted by \underline{b}^f , the *strain rate tensor* by

$$\underline{\underline{\epsilon}}(\underline{u}^f) = \frac{1}{2} \left(\nabla \underline{u}^f + (\nabla \underline{u}^f)^T \right),$$

and the constant *dynamic viscosity* by μ_{dyn}^f , respectively. The *fluid density* ρ^f is assumed to be constant.

Given velocities \underline{u} are prescribed at the DIRICHLET boundary Γ_D^f . At the NEUMANN boundary Γ_N^f , the fluid domain is loaded with external tractions \underline{h}^f . Additional tractions $\underline{h}_{\Gamma_{FSI}}^f$ arising from the fluid-structure coupling act onto the interface portion Γ_{FSI} of the boundary of the fluid subdomain Ω^f . These boundary conditions read

$$\underline{u}^f = \underline{u} \quad \text{on } \Gamma_D^f \times (0, T), \quad (2.5a)$$

$$\underline{\underline{\boldsymbol{\sigma}}}^{\mathcal{F}} \cdot \underline{\mathbf{n}}^{\mathcal{F}} = \underline{\mathbf{h}}^{\mathcal{F}} \quad \text{on } \Gamma_{\text{N}}^{\mathcal{F}} \times (0, T), \quad (2.5b)$$

$$\underline{\underline{\boldsymbol{\sigma}}}^{\mathcal{F}} \cdot \underline{\mathbf{n}}^{\mathcal{F}} = \underline{\mathbf{h}}_{\Gamma_{\text{FSI}}}^{\mathcal{F}} \quad \text{on } \Gamma_{\text{FSI}} \times (0, T), \quad (2.5c)$$

where the CAUCHY stress tensor $\underline{\underline{\boldsymbol{\sigma}}}^{\mathcal{F}}$ is defined as

$$\underline{\underline{\boldsymbol{\sigma}}}^{\mathcal{F}} = -p^{\mathcal{F}} \underline{\underline{\mathbf{I}}} + 2\mu_{\text{dyn}}^{\mathcal{F}} \underline{\underline{\boldsymbol{\varepsilon}}}(\underline{\mathbf{u}}^{\mathcal{F}}) \quad (2.6)$$

with the second order identity tensor $\underline{\underline{\mathbf{I}}}$. The role of the interface traction $\underline{\mathbf{h}}_{\Gamma_{\text{FSI}}}^{\mathcal{F}}$ in (2.5c) will be detailed in Section 2.3, when the coupling conditions will be discussed. As initial condition, a divergence free velocity field

$$\underline{\mathbf{u}}^{\mathcal{F}}(\underline{\mathbf{x}}, 0) = \underline{\mathbf{u}}_0^{\mathcal{F}}(\underline{\mathbf{x}}) \quad \text{with } \nabla \cdot \underline{\mathbf{u}}_0^{\mathcal{F}}(\underline{\mathbf{x}}) = 0 \quad \text{for } \underline{\mathbf{x}} \in \Omega^{\mathcal{F}} \quad (2.7)$$

has to be given.

2.2 Structure Field

For the solid domain Ω^{S} , a description in material coordinates $\underline{\mathbf{X}}$ is convenient. In contrast to the fluid field, a LAGRANGEan observer is utilized. Without loss of generality, the structure field is assumed to exhibit a nonlinear elastic behavior, while for convenience it is restricted to compressible and nearly incompressible solid bodies. The dynamic equilibrium of forces of inertia, internal forces, and an external *body force* $\underline{\mathbf{b}}_0^{\text{S}}$ per unit undeformed volume in the undeformed structural domain Ω^{S} is given by the nonlinear elastodynamics equation

$$\rho_0^{\text{S}} \frac{d^2 \underline{\mathbf{d}}^{\text{S}}}{dt^2} = \nabla_0 \cdot \underline{\underline{\mathbf{P}}} + \underline{\mathbf{b}}_0^{\text{S}} \quad \text{in } \Omega^{\text{S}} \times (0, T) \quad (2.8)$$

with the *solid displacement field* $\underline{\mathbf{d}}^{\text{S}}(\underline{\mathbf{X}}, t)$ as the primary field of unknowns. The *solid density per unit undeformed volume* is denoted by ρ_0^{S} . Internal forces are expressed in terms of the *first* PIOLA-KIRCHHOFF stress tensor

$$\underline{\underline{\mathbf{P}}} = \underline{\underline{\mathbf{F}}} \underline{\underline{\mathbf{S}}}. \quad (2.9)$$

It is computed based on the *deformation gradient* $\underline{\underline{\mathbf{F}}} = \nabla_0 \underline{\mathbf{x}}$ and the *second* PIOLA-KIRCHHOFF stress tensor $\underline{\underline{\mathbf{S}}}$. For the sake of simplicity hyperelastic material behavior with a *strain energy function* Ψ is assumed, such that the second PIOLA-KIRCHHOFF stress tensor is given as

$$\underline{\underline{\mathbf{S}}} = 2 \frac{\partial \Psi}{\partial \underline{\underline{\mathbf{C}}}}$$

with $\underline{\underline{\mathbf{C}}} = \underline{\underline{\mathbf{F}}}^{\text{T}} \underline{\underline{\mathbf{F}}}$ denoting the *right* CAUCHY-GREEN tensor. The constitutive behavior is characterized by the material parameters E^{S} and ν^{S} denoting YOUNG's modulus and POISSON's ratio, respectively.

As indicated in Figure 2.1, boundary conditions

$$\underline{\mathbf{d}}^{\text{S}} = \underline{\bar{\mathbf{d}}}^{\text{S}} \quad \text{on } \Gamma_{\text{D}}^{\text{S}} \times (0, T), \quad (2.10a)$$

$$\underline{\underline{\mathbf{P}}}^s \underline{\underline{\mathbf{n}}}_0^s = \underline{\underline{\mathbf{h}}}^s \quad \text{on } \Gamma_N^s \times (0, T), \quad (2.10b)$$

$$\underline{\underline{\mathbf{P}}}^s \underline{\underline{\mathbf{n}}}_0^s = \underline{\underline{\mathbf{h}}}_{\Gamma_{\text{FSI}}}^s \quad \text{on } \Gamma_{\text{FSI}} \times (0, T) \quad (2.10c)$$

are imposed on the solid domain with $\underline{\underline{\mathbf{n}}}_0^s$ denoting the outward pointing normal vector in the material configuration. On the DIRICHLET portion Γ_D^s of the solid boundary, displacements $\underline{\underline{\mathbf{d}}}^s$ are prescribed. External tractions $\underline{\underline{\mathbf{h}}}^s$ act onto the NEUMANN boundary Γ_N^s , while the fluid-structure interface Γ_{FSI} is subject to interface tractions $\underline{\underline{\mathbf{h}}}_{\Gamma_{\text{FSI}}}^s$. In addition, initial displacement and velocity fields are provided, reading

$$\underline{\underline{\mathbf{d}}}^s(\underline{\underline{\mathbf{X}}}, 0) = \underline{\underline{\mathbf{d}}}_0^s(\underline{\underline{\mathbf{X}}}) \quad \forall \underline{\underline{\mathbf{X}}} \in \Omega_0^s, \quad (2.11a)$$

$$\frac{d\underline{\underline{\mathbf{d}}}^s}{dt}(\underline{\underline{\mathbf{X}}}, 0) = \underline{\underline{\dot{\mathbf{d}}}}_0^s(\underline{\underline{\mathbf{X}}}) \quad \forall \underline{\underline{\mathbf{X}}} \in \Omega_0^s. \quad (2.11b)$$

2.3 Coupling Conditions at the Fluid-Structure Interface

Finally, the coupling of fluid field and structure field are addressed in this section. Both fields are coupled to each other through enforcement of kinematic and dynamic continuity conditions at the fluid-structure interface Γ_{FSI} . Physically motivated, a *no-slip condition*

$$\frac{\partial \underline{\underline{\mathbf{d}}}_{\Gamma_{\text{FSI}}}^s}{\partial t}(\underline{\underline{\mathbf{X}}}, t) = \underline{\underline{\mathbf{u}}}_{\Gamma_{\text{FSI}}}^f(\underline{\underline{\mathbf{x}}}, t) \quad \text{on } \Gamma_{\text{FSI}} \times (0, T) \quad (2.12a)$$

is assumed for kinematic continuity at the interface. Thus, both fluid flow across the fluid-structure interface and relative tangential movement of fluid and structure at the fluid-structure interface are prohibited. Condition (2.12a) couples the physical fields, i.e. fluid velocity field and structural displacement field. From (2.3) it is known that fluid velocity and grid velocity coincide at the fluid-structure interface. Thus, condition (2.3) can be rewritten as

$$\frac{\partial \underline{\underline{\mathbf{d}}}_{\Gamma_{\text{FSI}}}^s}{\partial t} = \frac{\partial \underline{\underline{\mathbf{d}}}_{\Gamma_{\text{FSI}}}^g}{\partial t} \quad \text{on } \Gamma_{\text{FSI}} \times (0, T). \quad (2.12b)$$

Integration w.r.t. time finally leads to the equivalent coupling condition

$$\underline{\underline{\mathbf{d}}}_{\Gamma_{\text{FSI}}}^s = \underline{\underline{\mathbf{d}}}_{\Gamma_{\text{FSI}}}^g \quad \text{on } \Gamma_{\text{FSI}} \times (0, T) \quad (2.12c)$$

that will be used as starting point when deriving the discrete set of equations of the coupled problem. This approach is favored in this thesis, since it allows for an interpretation as a meshtying problem that couples two displacement fields and, thus, enables a straightforward use of mortar methods for meshtying of non-matching discretizations as it will be applied to the FSI problem in Chapter 3.

Note that thanks to (2.3) all three conditions (2.12) are equivalent in the time-continuous regime. They might differ after temporal discretization depending on the choice of time integration schemes in fluid and structure field. In this thesis, condition (2.12c) will be enforced in the discrete regime.

By application of the method of sections, the interface traction fields $\underline{\mathbf{h}}_{\Gamma_{\text{FSI}}}^{\text{S}}$ and $\underline{\mathbf{h}}_{\Gamma_{\text{FSI}}}^{\text{F}}$ of the structure and the fluid field, respectively, become apparent at the interface as depicted in Figure 2.1. NEWTON's third law [175] — *actio = reactio* — requires the interface tractions to be equal, reading

$$\underline{\mathbf{h}}_{\Gamma_{\text{FSI}}}^{\text{S}} = -\underline{\mathbf{h}}_{\Gamma_{\text{FSI}}}^{\text{F}} \quad \text{on } \Gamma_{\text{FSI}} \times (0, T). \quad (2.13)$$

From a mathematical point of view, the interface traction can be interpreted as a LAGRANGE multiplier, that is used to enforce the kinematic constraint (2.12c). Without loss of generality, the LAGRANGE multiplier field is assumed to equal the interface traction field acting onto the structure side of the interface. This physical interpretation of the LAGRANGE multiplier field will be discussed in detail in Section 3.4.2, where its relation to the enforcement of essential boundary conditions and interface coupling conditions will be studied.

Remark 2.3.1 *The choice of interpreting the LAGRANGE multiplier field as the interface traction that acts onto the structure side of the interface is arbitrary, but will be kept consistently throughout this thesis. The opposite choice has been done in the thesis by KLÖPPEL [137]. In solid meshtying or contact mechanics problems, the LAGRANGE multiplier field is often associated with the negative interface traction of the slave side (cf. [95, 96, 119, 152, 184–186, 191, 256] among others).*

3 Finite Element Formulation and Monolithic System of Equations

Starting from the governing equations as presented in Chapter 2 of this thesis, the focus is now put on the discrete representation of the coupled problem as previously reported in MAYR *et al.* [163]. Numerical models for the physical fields of fluid and solid as well as the artificial ALE field need to be established. Additionally, discrete versions of the interface coupling conditions have to be derived.

For the spatial discretization of the involved *partial differential equations (PDEs)*, the *Finite Element Method (FEM)* is used. A detailed introduction to the FEM is omitted for the sake of brevity. For a detailed introduction to the FEM and its theoretical foundation, the monographs by HUGHES [124], STRANG and FIX [224], LARSON and BENGZON [150], and the first volume of the textbook series by ZIENKIEWICZ *et al.* [263] can be consulted among others.

A variety of numerical algorithms for the solution of time-dependent problems is covered by the books by BUTCHER [31], DEUFLHARD and BORNEMANN [58], HAIRER *et al.* [110], and HAIRER and WANNER [111] to name a few. In this thesis, integration in time will be performed by the *generalized- α method* [39, 129] or the *one-step- θ scheme* [67]. An adaptive time stepping scheme for monolithic FSI solvers will be proposed in Chapter 4.

The present monolithic solver has been implemented in the in-house research code *BACI* [243], a C++-based multi-physics code jointly developed by the Institute for Computational Mechanics¹ and the Mechanics & High Performance Computing Group² both at Technical University of Munich. Among others, finite element solvers for solid dynamics and fluid flow problems on deforming domains as well as different coupling algorithms are available. Earlier approaches to monolithic solvers for FSI have been developed and implemented in *BACI* and are documented in the theses by KÜTTLER [145] and KLÖPPEL [137].

After stating the weak form of the coupled FSI problem in Section 3.1, space- and time-discrete expressions for all fields and coupling conditions are presented in Section 3.2. The assembly and two variants of static condensation of the monolithic system of equations are detailed in Section 3.3, before a very simple example is studied to discuss and demonstrate basic properties of the proposed algorithms in Section 3.4. This chapter is concluded with an intermediate summary in Section 3.5.

3.1 Weak Form of the FSI Problem

To obtain the starting point for a finite element formulation, a weak form needs to be established. Denoting the collection of *trial solutions* by \mathcal{S} and the collection of *weighting functions* by \mathcal{T} ,

¹www.lnm.mw.tum.de

²www.mhpc.mw.tum.de

the solution spaces and test spaces for the fluid field [61] and ALE field [124]

$$\mathcal{S}_{\underline{\mathbf{u}}^{\mathcal{F}}} := \{ \underline{\mathbf{u}}^{\mathcal{F}} \in \mathcal{H}^1(\Omega^{\mathcal{F}}) \mid \underline{\mathbf{u}}^{\mathcal{F}} = \bar{\underline{\mathbf{u}}}^{\mathcal{F}} \text{ on } \Gamma_{\mathcal{D}}^{\mathcal{F}} \} \quad (3.1a)$$

$$\mathcal{S}_{p^{\mathcal{F}}} := \{ p^{\mathcal{F}} \in \mathcal{L}^2(\Omega^{\mathcal{F}}) \} \quad (3.1b)$$

$$\mathcal{S}_{\underline{\mathbf{d}}^{\mathcal{S}}} := \{ \underline{\mathbf{d}}^{\mathcal{S}} \in \mathcal{H}^1(\Omega^{\mathcal{S}}) \mid \underline{\mathbf{d}}^{\mathcal{S}} = \bar{\underline{\mathbf{d}}}^{\mathcal{S}} \text{ on } \Gamma_{\mathcal{D}}^{\mathcal{S}} \} \quad (3.1c)$$

$$\mathcal{T}_{\underline{\mathbf{v}}^{\mathcal{F}}} := \{ \underline{\mathbf{v}}^{\mathcal{F}} \in \mathcal{H}^1(\Omega^{\mathcal{F}}) \mid \underline{\mathbf{v}}^{\mathcal{F}} = \underline{\mathbf{0}} \text{ on } \Gamma_{\mathcal{D}}^{\mathcal{F}} \} \quad (3.1d)$$

$$\mathcal{T}_{q^{\mathcal{F}}} := \{ q^{\mathcal{F}} \in \mathcal{L}^2(\Omega^{\mathcal{F}}) \} \quad (3.1e)$$

$$\mathcal{T}_{\underline{\mathbf{w}}^{\mathcal{S}}} := \{ \underline{\mathbf{w}}^{\mathcal{S}} \in \mathcal{H}^1(\Omega^{\mathcal{S}}) \mid \underline{\mathbf{w}}^{\mathcal{S}} = \underline{\mathbf{0}} \text{ on } \Gamma_{\mathcal{D}}^{\mathcal{S}} \} \quad (3.1f)$$

as well as the structure field [124]

$$\mathcal{S}_{\underline{\mathbf{d}}^{\mathcal{S}}} := \{ \underline{\mathbf{d}}^{\mathcal{S}} \in \mathcal{H}^1(\Omega^{\mathcal{S}}) \mid \underline{\mathbf{d}}^{\mathcal{S}} = \bar{\underline{\mathbf{d}}}^{\mathcal{S}} \text{ on } \Gamma_{\mathcal{D}}^{\mathcal{S}} \} \quad (3.2a)$$

$$\mathcal{T}_{\underline{\mathbf{w}}^{\mathcal{S}}} := \{ \underline{\mathbf{w}}^{\mathcal{S}} \in \mathcal{H}^1(\Omega^{\mathcal{S}}) \mid \underline{\mathbf{w}}^{\mathcal{S}} = \underline{\mathbf{0}} \text{ on } \Gamma_{\mathcal{D}}^{\mathcal{S}} \} \quad (3.2b)$$

are defined. Finally, the respective dual spaces for the Lagrange multiplier field $\underline{\boldsymbol{\lambda}}$ are given as

$$\mathcal{S}_{\underline{\boldsymbol{\lambda}}} := \{ \underline{\boldsymbol{\lambda}} \in \mathcal{H}^{-\frac{1}{2}}(\Gamma_{\text{FSI}}) \} \quad (3.3a)$$

$$\mathcal{T}_{\underline{\boldsymbol{\mu}}} := \{ \underline{\boldsymbol{\mu}} \in \mathcal{H}^{-\frac{1}{2}}(\Gamma_{\text{FSI}}) \}. \quad (3.3b)$$

as detailed by WOHLMUTH [253] as a LAGRANGE multiplier field is used to enforce the kinematic interface constraints weakly via a dual mortar method.

The *method of weighted residuals* is applied to the strong form of the fluid, solid, and interface problems detailed in Sections 2.1, 2.2, and 2.3, respectively. This is done by multiplication with the respective test functions $\underline{\mathbf{v}}$, q , $\underline{\mathbf{w}}$, and $\underline{\boldsymbol{\mu}}$ for the fluid velocity, the fluid pressure, the solid displacement, and the LAGRANGE multiplier field. Subsequent integration by parts as well as exploitation of the boundary conditions gives rise to the weak problem: Find $\underline{\mathbf{d}}^{\mathcal{S}} \in \mathcal{S}_{\underline{\mathbf{d}}^{\mathcal{S}}}$, $\underline{\mathbf{u}}^{\mathcal{F}} \in \mathcal{S}_{\underline{\mathbf{u}}^{\mathcal{F}}}$, $p^{\mathcal{F}} \in \mathcal{S}_{p^{\mathcal{F}}}$, $\underline{\mathbf{d}}^{\mathcal{S}} \in \mathcal{S}_{\underline{\mathbf{d}}^{\mathcal{S}}}$ and $\underline{\boldsymbol{\lambda}} \in \mathcal{S}_{\underline{\boldsymbol{\lambda}}}$ for all $\underline{\mathbf{w}}^{\mathcal{S}} \in \mathcal{T}_{\underline{\mathbf{w}}^{\mathcal{S}}}$, $\underline{\mathbf{v}}^{\mathcal{F}} \in \mathcal{T}_{\underline{\mathbf{v}}^{\mathcal{F}}}$, $q^{\mathcal{F}} \in \mathcal{T}_{q^{\mathcal{F}}}$, $\underline{\mathbf{w}}^{\mathcal{S}} \in \mathcal{T}_{\underline{\mathbf{w}}^{\mathcal{S}}}$ and $\underline{\boldsymbol{\mu}} \in \mathcal{T}_{\underline{\boldsymbol{\mu}}}$ such that

$$\begin{aligned} 0 &= \left(\underline{\mathbf{v}}^{\mathcal{F}}, \rho^{\mathcal{F}} \frac{\partial \underline{\mathbf{u}}^{\mathcal{F}}}{\partial t} \right)_{\Omega^{\mathcal{F}}} + \left(\underline{\mathbf{v}}^{\mathcal{F}}, \rho^{\mathcal{F}} \underline{\mathbf{c}} \cdot \nabla \underline{\mathbf{u}}^{\mathcal{F}} \right)_{\Omega^{\mathcal{F}}} - \left(\nabla \cdot \underline{\mathbf{v}}^{\mathcal{F}}, p^{\mathcal{F}} \right)_{\Omega^{\mathcal{F}}} \\ &+ \left(\nabla \underline{\mathbf{v}}^{\mathcal{F}}, 2\mu_{\text{dyn}}^{\mathcal{F}} \underline{\boldsymbol{\epsilon}}(\underline{\mathbf{u}}^{\mathcal{F}}) \right)_{\Omega^{\mathcal{F}}} - \left(q^{\mathcal{F}}, \nabla \cdot \underline{\mathbf{u}}^{\mathcal{F}} \right)_{\Omega^{\mathcal{F}}} - \left(\underline{\mathbf{v}}^{\mathcal{F}}, \rho^{\mathcal{F}} \underline{\mathbf{b}}^{\mathcal{F}} \right)_{\Omega^{\mathcal{F}}} \\ &- \left(\underline{\mathbf{v}}^{\mathcal{F}}, \bar{\underline{\mathbf{h}}}^{\mathcal{F}} \right)_{\Gamma_{\text{N}}^{\mathcal{F}}} + \left(\underline{\mathbf{v}}^{\mathcal{F}}, \underline{\boldsymbol{\lambda}} \right)_{\Gamma_{\text{FSI}}}, \end{aligned} \quad (3.4a)$$

$$0 = \left(\underline{\mathbf{w}}^{\mathcal{S}}, \rho_0^{\mathcal{S}} \frac{d^2 \underline{\mathbf{d}}^{\mathcal{S}}}{dt^2} - \underline{\mathbf{b}}_0^{\mathcal{S}} \right)_{\Omega^{\mathcal{S}}} + \left(\nabla_0 \underline{\mathbf{w}}^{\mathcal{S}}, \underline{\mathbf{P}} \right)_{\Omega^{\mathcal{S}}} - \left(\underline{\mathbf{w}}^{\mathcal{S}}, \bar{\underline{\mathbf{h}}}_0^{\mathcal{S}} \right)_{\Gamma_{\text{N}}^{\mathcal{S}}} - \left(\underline{\mathbf{w}}^{\mathcal{S}}, \underline{\boldsymbol{\lambda}} \right)_{\Gamma_{\text{FSI}}}, \quad (3.4b)$$

$$0 = \left(\underline{\boldsymbol{\mu}}, \underline{\mathbf{d}}_{\Gamma_{\text{FSI}}}^{\mathcal{S}} - \underline{\mathbf{d}}_{\Gamma_{\text{FSI}}}^{\mathcal{S}} \right)_{\Gamma_{\text{FSI}}}. \quad (3.4c)$$

This will be used as the starting point for the finite element discretization.

3.2 Discretization of the FSI Problem

The weak form (3.4) has to be discretized in space and time. For the monolithic approach presented here, the spatial discretization of the fluid and the structure field is done with finite elements. For the spatial discretization of the ALE field, several approaches are available, e.g. finite elements or spring models. An overview of such mesh motion techniques as well as a comparison are given in Appendix A. The constraints at the interface are enforced using a dual mortar method, which has been proposed by KLÖPPEL *et al.* [138] for the FSI case. This results in great freedom during mesh generation to tailor the meshes to the needs of the individual fields, since the nodes of the fluid and structure mesh do not have to match at the interface.

Temporal discretization of the fluid and the structure field is performed independently by finite differencing with the same time step size Δt in both fields. Subsequently, field-specific fully implicit, single-step, and single-stage time integration schemes are applied, which can be tailored to the needs of the individual field. Depending on the actual choices of time integration schemes and their algorithmic parameters, the balance of linear momentum is formulated at an intermediate time instant $t_m \in]t_n, t_{n+1}]$ where the index m indicates a mid-point somewhere in the time interval $]t_n, t_{n+1}]$. In general, the actual time instants for equilibrium in the fluid and the structure field do not coincide, i.e. $t_m^{\mathcal{F}} \neq t_m^{\mathcal{S}}$. MAYR *et al.* [163] allow for the freedom of choosing the time integration schemes for the fluid and structure field independently and still maintaining temporal consistency between both fields, which will be shown in detail in Section 3.2.4. In the sequel, time-discrete quantities are indexed with the subscript $(\bullet)_n$ to indicate the current time step. To prepare the adaptive time stepping scheme detailed in Chapter 4, the time step size is denoted by Δt_n to hint at possible changes of the time step size, even if a constant time step size might be used.

Another new aspect of time integration in the work of MAYR *et al.* [163] is that field-specific predictors are allowed within the monolithic FSI framework. By applying predictors in the solid and the fluid field, one aims at a good initial guess as a starting point for the nonlinear solution procedure and hopes to reduce numerical effort of the nonlinear solution scheme. Due to possible predictors in the structure and fluid field, the solution at the beginning of the nonlinear iteration loop may differ from the converged solution of the previous time step by additional increments, reading

$$\mathbf{d}_{n+1}^{\mathcal{S},0} = \mathbf{d}_n^{\mathcal{S}} + \Delta \mathbf{d}_p^{\mathcal{S}} \quad (3.5a)$$

$$\mathbf{u}_{n+1}^{\mathcal{F},0} = \mathbf{u}_n^{\mathcal{F}} + \Delta \mathbf{u}_p^{\mathcal{F}} \quad (3.5b)$$

with the subscript $(\bullet)_p$ indicating the predictor step. Within the predictors, both fields can evolve independently, leading to a possible violation of the kinematic continuity requirement (2.12) at the fluid-structure interface, i.e. possibly incompatible initial guesses for the structure and the fluid fields. This violation can be measured and will be accounted for when the discrete kinematic coupling conditions are derived in Section 3.2.4. Without any predictor, these additional increments vanish, i.e. $\Delta \mathbf{d}_p^{\mathcal{S}} = \mathbf{0}$ and $\Delta \mathbf{u}_p^{\mathcal{F}} = \mathbf{0}$.

Remark 3.2.1 MAYR *et al.* [163] use very simple extrapolation-based predictions of the solid displacement field. In principle, the proposed solution schemes are able to handle fluid predictors as well. However, if only the velocity field is predicted in a comparably simple way as in the structure field, the pressure field does not match the velocity field after the prediction. An

extrapolation-based prediction of the pressure field is not possible since the pressure field does not provide a time derivative, but rather adapts oneself as instantaneous quantity to guarantee a divergence-free velocity field. Hence, MAYR et al. [163] recommend to just predict the structural solution unless sophisticated fluid predictors that include a pressure projection step are available. In the present implementation, however, only explicit predictors are considered that come with only negligible additional cost and therefore refrain from using a nonconstant fluid predictor step.

Remark 3.2.2 In implementations, where DIRICHLET boundary conditions are applied in the predictor step and then the respective degrees of freedom are excluded from the nonlinear solution process, the predictor strategy also allows for the prescription of inhomogeneous DIRICHLET boundary conditions at the fluid-structure interface.

Discretization of the weak form (3.4) of the coupled FSI problem can be performed independently for each field. Discretization of the fluid contribution (3.4a) results in the fluid residual

$$\mathbf{r}^{\mathcal{F}} = \mathbf{r}_{\mathbf{u}^{\mathcal{F}}}^{\mathcal{F}} + \mathbf{r}_{\lambda}^{\mathcal{F}} = \begin{bmatrix} \mathbf{r}_{\Gamma}^{\mathcal{F}} \\ \mathbf{r}_{\Gamma_{\text{FSI}}}^{\mathcal{F}} \\ \mathbf{r}_{\Gamma}^{\mathcal{G}} \\ \mathbf{r}_{\Gamma_{\text{FSI}}}^{\mathcal{G}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_{\lambda, \Gamma_{\text{FSI}}}^{\mathcal{F}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad (3.6)$$

where the first term $\mathbf{r}_{\mathbf{u}^{\mathcal{F}}}^{\mathcal{F}}$ on the right-hand side contains the standard fluid residual and only the second term $\mathbf{r}_{\lambda}^{\mathcal{F}}$ accounts for the interface coupling of fluid and structure field. Accordingly, the discretization of the structural contribution (3.4b) is written as

$$\mathbf{r}^{\mathcal{S}} = \mathbf{r}_{\mathbf{d}^{\mathcal{S}}}^{\mathcal{S}} + \mathbf{r}_{\lambda}^{\mathcal{S}} = \begin{bmatrix} \mathbf{r}_{\Gamma}^{\mathcal{S}} \\ \mathbf{r}_{\Gamma_{\text{FSI}}}^{\mathcal{S}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{r}_{\lambda, \Gamma_{\text{FSI}}}^{\mathcal{S}} \end{bmatrix} \quad (3.7)$$

with the first term $\mathbf{r}_{\mathbf{d}^{\mathcal{S}}}^{\mathcal{S}}$ accounting for the pure structural problem and the second term $\mathbf{r}_{\lambda}^{\mathcal{S}}$ again representing the interface coupling. Finally, the weak coupling condition (3.4c) is discretized yielding a residual contribution $\mathbf{r}^{\text{coupl}}$. After assembling these three individual residuals into the global FSI residual vector \mathbf{r}^{FSI} , that depends on the structural unknowns, the fluid unknowns, and the unknown LAGRANGE multipliers, the solution of the nonlinear coupled FSI problem is obtained by solving for

$$\mathbf{r}^{\text{FSI}} = \begin{bmatrix} \mathbf{r}^{\mathcal{S}} \\ \mathbf{r}^{\mathcal{F}} \\ \mathbf{r}^{\text{coupl}} \end{bmatrix} = \mathbf{0}. \quad (3.8)$$

To solve the nonlinear problem (3.8), a NEWTON-type method is applied requiring the full linearization of \mathbf{r}^{FSI} and, thus, of all single field residuals. After summarizing all unknowns of the structure field in $\mathbf{x}^{\mathcal{S}}$ and those of the fluid field in $\mathbf{x}^{\mathcal{F}}$ to ease notation, respectively, the resulting linear system in NEWTON iteration step $k \geq 0$ reads

$$\begin{bmatrix} \frac{\partial \mathbf{r}_{\mathbf{d}^{\mathcal{S}}}^{\mathcal{S}}}{\partial \mathbf{x}^{\mathcal{S}}} & \mathbf{0} & \frac{\partial \mathbf{r}_{\lambda}^{\mathcal{S}}}{\partial \lambda} \\ \mathbf{0} & \frac{\partial \mathbf{r}_{\mathbf{u}^{\mathcal{F}}}^{\mathcal{F}}}{\partial \mathbf{x}^{\mathcal{F}}} & \frac{\partial \mathbf{r}_{\lambda}^{\mathcal{F}}}{\partial \lambda} \\ \frac{\partial \mathbf{r}^{\text{coupl}}}{\partial \mathbf{d}_{\Gamma}^{\mathcal{S}}} & \frac{\partial \mathbf{r}^{\text{coupl}}}{\partial \mathbf{u}_{\Gamma}^{\mathcal{F}}} & \mathbf{0} \end{bmatrix}_{n+1}^k \begin{bmatrix} \Delta \mathbf{x}^{\mathcal{S}} \\ \Delta \mathbf{x}^{\mathcal{F}} \\ \Delta \lambda \end{bmatrix}_{n+1}^{k+1} = - \begin{bmatrix} \mathbf{r}^{\mathcal{S}} \\ \mathbf{r}^{\mathcal{F}} \\ \mathbf{r}^{\text{coupl}} \end{bmatrix}_{n+1}^k \quad (3.9)$$

where the subscript $(\bullet)_{\Gamma_{\text{FSI}}}$ denoting the fluid-structure interface has been replaced by $(\bullet)_{\Gamma}$ to shorten the notation. This short notation is used throughout the remainder of this thesis, unless a detailed specification is necessary. The current nonlinear iteration is indicated by the superscript $(\bullet)^k$. In (3.9), the splitting into degrees of freedom that belong to the interior of Ω^{S} or Ω^{F} and those located at the fluid-structure interface Γ_{FSI} is omitted for clarity of presentation. It will be re-introduced when the single field contributions to (3.9) will be derived in the following subsections. The matrix contribution $\partial \mathbf{r}_{\text{d}^{\text{S}}}^{\text{S}} / \partial \mathbf{x}^{\text{S}}$ of the structure field will be discussed in detail in Section 3.2.2. Section 3.2.1 deals with the fluid discretization and will specify the matrix contribution $\partial \mathbf{r}_{\text{u}^{\text{F}}}^{\text{F}} / \partial \mathbf{x}^{\text{F}}$. The remaining matrix contributions that are related to the interface coupling will be addressed in Section 3.2.4.

The solution vector is updated in every iteration via

$$\begin{bmatrix} \mathbf{x}^{\text{S}} \\ \mathbf{x}^{\text{F}} \\ \boldsymbol{\lambda} \end{bmatrix}_{n+1}^{k+1} = \begin{bmatrix} \mathbf{x}^{\text{S}} \\ \mathbf{x}^{\text{F}} \\ \boldsymbol{\lambda} \end{bmatrix}_{n+1}^k + \begin{bmatrix} \Delta \mathbf{x}^{\text{S}} \\ \Delta \mathbf{x}^{\text{F}} \\ \Delta \boldsymbol{\lambda} \end{bmatrix}_{n+1}^{k+1}. \quad (3.10)$$

Note that due to the possibility of field-specific predictors, the initial guess for the nonlinear iterative procedure may differ from the converged solution of the last time step, i.e. $\mathbf{x}_{n+1}^0 \neq \mathbf{x}_n$, cf. (3.5).

In order to obtain the full linearization of the coupled FSI problem, first space and time discretization as well as linearization of the fluid, ALE, and structure field equations need to be performed. As stated above, this will be done field-wise in the subsequent sections. A brief introduction to the mortar method will be given in Section 3.2.3. Afterwards, the coupling at the interface via the dual mortar method is illustrated. Finally, temporal consistent coupling of the fluid and the structure field is introduced. The assembly of the global monolithic system will then be shown in Section 3.3.

3.2.1 Fluid field

In this thesis, stabilized equal-order interpolated finite elements are used for spatial discretization of the fluid field. Specialized FEM formulations for flow problems including detailed analysis of the required stabilization techniques can be found in the textbooks by DONEA and HUERTA [61] or GRESHO and SANI [102], in the second volume of the textbook series by ZIENKIEWICZ *et al.* [262] or in the theses by FÖRSTER [81], GRAVEMEIER [98], or WALL [242].

Assuming a matching grid layout for the ALE mesh motion and for the fluid solution, the ansatz for spatial discretization of ALE displacement, fluid velocity and fluid pressure field read

$$\underline{\mathbf{d}}^{\text{S}} \approx \sum_{j=1}^{n^{\text{nd},\text{F}}} N_j^{\text{S}} \mathbf{d}_j^{\text{S}}, \quad (3.11a)$$

$$\underline{\mathbf{u}}^{\text{F}} \approx \sum_{j=1}^{n^{\text{nd},\text{F}}} N_j^{\text{F}} \mathbf{u}_j^{\text{F}}, \quad (3.11b)$$

$$\mathbf{p}^{\mathcal{F}} \approx \sum_{j=1}^{n^{\text{nd},\mathcal{F}}} N_j^{\mathcal{F}} p_j^{\mathcal{F}} \quad (3.11c)$$

with $n^{\text{nd},\mathcal{F}}$ denoting the number of fluid nodes and $N_j^{\mathcal{G}}$ and $N_j^{\mathcal{F}}$ being the finite element ansatz functions. While space-continuous quantities have been denoted by underlined bold face, the underline character is dropped to indicate vectors of space-discrete nodal values.

Temporal discretization of the fluid velocity field is done by finite differences. Either the generalized- α scheme proposed by JANSEN *et al.* [129] or the one-step- θ scheme [67] are used for time integration. The generalized- α scheme is preferred due to its unconditional stability in the linear regime, second order accuracy, and the possibility of user-controlled dissipation of high frequencies. Numerical dissipation is controlled by choosing a *spectral radius* $\rho_{\infty}^{\mathcal{F}}$, which is used as a starting point to calculate the time integration parameters $\alpha_f^{\mathcal{F}}$, $\alpha_m^{\mathcal{F}}$, and $\gamma^{\mathcal{F}}$ [129]. One-step- θ is used with $0.5 \leq \theta^{\mathcal{F}} \leq 1$, only. Second order accuracy can be achieved if and only if $\theta^{\mathcal{F}} = 0.5$ while $\theta^{\mathcal{F}} \neq 0.5$ results in first order accuracy only. Both schemes formulate the balance of linear momentum at an intermediate time instant $t_m^{\mathcal{F}} \in]t_n, t_{n+1}]$ with

$$t_m^{\mathcal{F}} = \begin{cases} t_{n+\alpha_f^{\mathcal{F}}} & \text{for generalized-}\alpha \\ t_{n+\theta} & \text{for one-step-}\theta \end{cases}$$

which will play an important role for the temporal interpolation of interface traction later in Section 3.2.4. Both schemes are single-step and single-stage schemes, i.e. they require only one previous solution to be known and only need one nonlinear problem per time step to be solved. Their fully implicit character makes them perfectly suitable to solve the incompressible fluid flow problem without too distinctive restrictions on the time step size Δt_n .

To prepare the coupling at the interface, fluid velocity degrees of freedom are divided into two sets. Those associated with an interface node are denoted by $\mathbf{u}_{\Gamma}^{\mathcal{F}}$ while the remaining velocity degrees of freedom are collected in the vector $\hat{\mathbf{u}}_I^{\mathcal{F}}$. For notational convenience, the vector $\mathbf{p}^{\mathcal{F}}$ of all nodal pressure values is merged into the vector of inner velocities, reading

$$\mathbf{u}_I^{\mathcal{F}} = \begin{bmatrix} \hat{\mathbf{u}}_I^{\mathcal{F}} \\ \mathbf{p}^{\mathcal{F}} \end{bmatrix}.$$

The introduced split into quantities belonging either to the interior or the fluid-structure interface of the fluid domain yields the matrix representation

$$\mathcal{F}_{\alpha\beta} = \frac{\partial \mathbf{r}_{\alpha}^{\mathcal{F}}}{\partial \mathbf{u}_{\beta}^{\mathcal{F}}}, \quad \mathcal{F}_{\alpha\beta}^{\mathcal{G}} = \frac{\partial \mathbf{r}_{\alpha}^{\mathcal{G}}}{\partial \mathbf{d}_{\beta}^{\mathcal{G}}}, \quad \alpha, \beta \in \{I, \Gamma\}$$

of the linearization of the discrete nonlinear fluid residual $\mathbf{r}^{\mathcal{F}}$. In order to compute the solution increment $\Delta \mathbf{x}_{k+1}^{\mathcal{F},n+1}$, the linear system

$$\begin{bmatrix} \mathcal{F}_{II} & \mathcal{F}_{I\Gamma} & \mathcal{F}_{II}^{\mathcal{G}} & \mathcal{F}_{I\Gamma}^{\mathcal{G}} \\ \mathcal{F}_{\Gamma I} & \mathcal{F}_{\Gamma\Gamma} & \mathcal{F}_{\Gamma I}^{\mathcal{G}} & \mathcal{F}_{\Gamma\Gamma}^{\mathcal{G}} \end{bmatrix}_{n+1}^k \begin{bmatrix} \Delta \mathbf{u}_I^{\mathcal{F}} \\ \Delta \mathbf{u}_{\Gamma}^{\mathcal{F}} \\ \Delta \mathbf{d}_I^{\mathcal{G}} \\ \Delta \mathbf{d}_{\Gamma}^{\mathcal{G}} \end{bmatrix}_{n+1}^{k+1} = - \begin{bmatrix} \mathbf{r}_I^{\mathcal{F}} \\ \mathbf{r}_{\Gamma}^{\mathcal{F}} \end{bmatrix}_{n+1}^k \quad (3.12)$$

has to be solved in every nonlinear iteration step $k \geq 0$. Considering the mesh motion of the ALE mesh, we assume that discretization and linearization of (2.1) result in an ALE system matrix \mathcal{A} . The linearized version of (2.1) reads

$$\left[\begin{array}{cc} \mathcal{A}_{\text{II}} & \mathcal{A}_{\text{IF}} \end{array} \right]_{n+1}^k \left[\begin{array}{c} \Delta \mathbf{d}_{\text{I}}^{\mathcal{G}} \\ \Delta \mathbf{d}_{\text{I}}^{\mathcal{G}} \end{array} \right]_{n+1}^{k+1} = -\mathbf{r}_{\Gamma, n+1}^{\mathcal{G}, k}. \quad (3.13)$$

Note that the vectors of unknowns in (3.12) and (3.13) both contain the mesh displacements $\mathbf{d}^{\mathcal{G}}$. Hence, both systems can be combined to

$$\left[\begin{array}{cccc} \mathcal{F}_{\text{II}} & \mathcal{F}_{\text{IF}} & \mathcal{F}_{\text{II}}^{\mathcal{G}} & \mathcal{F}_{\text{IF}}^{\mathcal{G}} \\ \mathcal{F}_{\text{FI}} & \mathcal{F}_{\text{FF}} & \mathcal{F}_{\text{FI}}^{\mathcal{G}} & \mathcal{F}_{\text{FF}}^{\mathcal{G}} \\ \mathbf{0} & \mathbf{0} & \mathcal{A}_{\text{II}} & \mathcal{A}_{\text{IF}} \end{array} \right]_{n+1}^k \left[\begin{array}{c} \Delta \mathbf{u}_{\text{I}}^{\mathcal{F}} \\ \Delta \mathbf{u}_{\text{I}}^{\mathcal{F}} \\ \Delta \mathbf{d}_{\text{I}}^{\mathcal{G}} \\ \Delta \mathbf{d}_{\text{I}}^{\mathcal{G}} \end{array} \right]_{n+1}^{k+1} = - \left[\begin{array}{c} \mathbf{r}_{\text{I}}^{\mathcal{F}} \\ \mathbf{r}_{\text{I}}^{\mathcal{F}} \\ \mathbf{r}_{\text{I}}^{\mathcal{G}} \\ \mathbf{r}_{\text{I}}^{\mathcal{G}} \end{array} \right]_{n+1}^k, \quad (3.14)$$

which involves an expression for $\partial \mathbf{r}_{\text{I}}^{\mathcal{F}} / \partial \mathbf{x}^{\mathcal{F}}$ needed for the schematic monolithic system (3.9). Note that an equation for the evolution of the interface displacements $\mathbf{d}_{\text{I}}^{\mathcal{G}}$ is still missing in (3.14), since the interface deformation cannot evolve freely, but has to follow the fluid field's or structure field's interface motion. To close the fluid linear system (3.14), a discrete coupling condition that relates fluid interface velocities $\mathbf{u}_{\text{I}}^{\mathcal{F}}$ to ALE interface displacements $\mathbf{d}_{\text{I}}^{\mathcal{G}}$ is necessary. It will be derived in Section 3.2.4 along with the discrete coupling conditions at the fluid-structure interface.

3.2.2 Structure field

Finite elements are also used here for spatial discretization of the structure field. Detailed presentations of the FEM in the context of solid mechanics can be found in the monographs by BELYTSCHKO *et al.* [16], HUGHES [124], DE BORST *et al.* [45], WRIGGERS [255], or the third volume of the textbook series by ZIENKIEWICZ *et al.* [261].

For the solid displacement field, the ansatz

$$\underline{\mathbf{d}}^{\mathcal{S}} \approx \sum_{j=1}^{n^{\text{nd}, \mathcal{S}}} N_j^{\mathcal{S}} \mathbf{d}_j^{\mathcal{S}} \quad (3.15)$$

is made with $n^{\text{nd}, \mathcal{S}}$ denoting the number of nodes of the structure field, $N_j^{\mathcal{S}}$ being the shape functions and $\mathbf{d}^{\mathcal{S}}$ the vector of nodal displacements, respectively.

For time integration, the generalized- α method by CHUNG and HULBERT [39] is applied due to its unconditional stability in the linear regime, second order accuracy, and the possibility of user-controlled numerical dissipation of high frequencies. For the calculation of the time integration parameters $\alpha_{\text{f}}^{\mathcal{S}}$, $\alpha_{\text{m}}^{\mathcal{S}}$, $\beta^{\mathcal{S}}$, and $\gamma^{\mathcal{S}}$ based on the spectral radius $\rho_{\infty}^{\mathcal{S}}$ see the original paper [39]. As in the fluid, the generalized- α method formulates the balance of linear momentum at an intermediate instance in time, namely $t_{\text{m}}^{\mathcal{S}} = t_{n+1-\alpha_{\text{f}}^{\mathcal{S}}}$.

Remark 3.2.3 *Although generalized- α schemes for fluid and solid use different indexing for the intermediate point in time, namely $t_{n+\alpha_{\text{f}}^{\mathcal{F}}}$ vs. $t_{n+1-\alpha_{\text{f}}^{\mathcal{S}}}$, choosing equal spectral radii $\rho_{\infty}^{\mathcal{F}} = \rho_{\infty}^{\mathcal{S}}$ indeed results in $\alpha_{\text{f}}^{\mathcal{F}} \neq \alpha_{\text{f}}^{\mathcal{S}}$, but still $t_{\text{m}}^{\mathcal{F}} = t_{\text{m}}^{\mathcal{S}}$.*

Using the splitting into interface and interior degrees of freedom and the shorthand notation

$$\mathfrak{S}_{\alpha\beta} = \frac{\partial \mathbf{r}_{\alpha}^{\mathfrak{S}}}{\partial \mathbf{d}_{\beta}^{\mathfrak{S}}}$$

with $\alpha, \beta \in \{\text{I}, \Gamma\}$ to denote the linearization of the discrete nonlinear residual $\mathbf{r}^{\mathfrak{S}}$ yields the linear system of equations

$$\begin{bmatrix} \mathfrak{S}_{\text{II}} & \mathfrak{S}_{\text{I}\Gamma} \\ \mathfrak{S}_{\Gamma\text{I}} & \mathfrak{S}_{\Gamma\Gamma} \end{bmatrix}_{n+1}^k \begin{bmatrix} \Delta \mathbf{d}_{\text{I}}^{\mathfrak{S}} \\ \Delta \mathbf{d}_{\Gamma}^{\mathfrak{S}} \end{bmatrix}_{n+1}^{k+1} = - \begin{bmatrix} \mathbf{r}_{\text{I}}^{\mathfrak{S}} \\ \mathbf{r}_{\Gamma}^{\mathfrak{S}} \end{bmatrix}_{n+1}^k \quad (3.16)$$

to be solved in every NEWTON step $k \geq 0$. The linear system (3.16) includes an expression for $\partial \mathbf{r}_{\mathbf{d}^{\mathfrak{S}}}^{\mathfrak{S}} / \partial \mathbf{x}^{\mathfrak{S}}$, which is needed in (3.9).

3.2.3 LAGRANGE Multiplier Field

In this thesis, a *dual mortar method* is used to weakly enforce the kinematic coupling condition (2.12c). The mortar method has been introduced by BELGACEM [15] to impose interface constraints weakly. WOHLMUTH [253, 254] proposed an alternative choice of mortar shape functions, namely *dual shape functions*. Their properties and benefits will be discussed below. Meanwhile, the treatment of curved interfaces [80], the extension to higher order [146], and aspects of numerical evaluation of the mortar integrals [68, 189, 190] have been addressed in literature. Many applications can be found in computational analysis of contact mechanics problems [96, 119, 184–186, 191, 192]. For a literature survey on the development and the application of mortar methods in contact mechanics see the thesis by POPP [184]. In recent years, mortar methods have spread to a broad spectrum of applications like mesh tying in solid mechanics [141, 189] and fluid dynamics [64], FSI problems [138, 163], or fluid-structure-contact interaction [162]. Very recently, FARAH *et al.* [69] extended the mortar method to volume-coupled multi-physics problems, e.g. poroelasticity or thermo-structure interaction.

In the context of FSI, the first application of the dual mortar method to problems with non-matching interface discretizations has been described by KLÖPPEL *et al.* [138]. However, alternative approaches to deal with non-matching interface discretizations are available based on interpolation techniques [56, 72], localized LAGRANGE multipliers [205, 206], or special interface elements [135] to name a few. Very recently, DEPARIS *et al.* [52] proposed an interpolation-based coupling method called INTERNODES, that also allows for non-matching grids. It consists of two coupling operators, one for the continuity of the primal and one for the continuity of the dual variable. Either LAGRANGEan basis functions or *rescaled localized radial basis functions* [54] are used to construct the interpolation operators where the latter result in lower interpolation errors in case of non-conforming interfaces and a low polynomial degree [52]. The INTERNODES method has been applied to fluid-structure interaction problems by DEPARIS *et al.* [55] and recently in the thesis by FORTI [83].

Here, only a very brief introduction to some basics of the dual mortar method is given. For detailed derivations as well as theoretical background and analysis see further literature, e.g. [80, 253, 254] and references therein.

In mortar methods, one usually distinguishes between a *master* and a *slave* side of the interface. Discretization of the LAGRANGE multiplier field is always performed on the slave side.

Opposed to classical LAGRANGE multiplier choices, the discretization of the LAGRANGE multiplier field $\underline{\lambda}$ with the dual mortar method is based on so-called *dual shape functions* Φ_j leading to the discretization

$$\underline{\lambda} \approx \sum_{j=1}^{n^{\text{nd,sl}}} \Phi_j \lambda_j \quad (3.17)$$

with discrete nodal LAGRANGE multipliers λ_j and $n^{\text{nd,sl}}$ nodes on the slave side of the interface Γ_{FSI} . The dual shape functions Φ_j are constructed such that they satisfy the *biorthogonality condition*

$$\int_{\Gamma^{\text{sl}}} \Phi_j N_k^{\text{sl}} d\Gamma = \delta_{jk} \int_{\Gamma^{\text{sl}}} N_k^{\text{sl}} d\Gamma \quad (3.18)$$

which is often replaced by its element-wise representation

$$\int_{\Gamma^{\text{sl},e}} \Phi_j N_k^{\text{sl}} d\Gamma = \delta_{jk} \int_{\Gamma^{\text{sl},e}} N_k^{\text{sl}} d\Gamma. \quad (3.19)$$

The construction of dual shape functions Φ_j based on the biorthogonality condition (3.19) with the standard LAGRANGE polynomial shape function N_k has first been presented by SCOTT and ZHANG [217] for projection-like operators and has been transferred to the LAGRANGE multiplier case by WOHLMUTH [253]. Since (3.19) must hold in the physical space rather than in the element parameter space, the dual shape functions depend on the actual distortion of the underlying finite element. Thus, they can be given explicitly only for cases with constant Jacobian mappings, but need to be computed for all other cases based on the actual mesh. For details of the construction of the dual shape functions, see [80, 146, 147, 187, 254]. From a computational point of view, they only have to be computed once in the beginning and can then be reused throughout the entire computation since the coupling is performed in the material configuration [138].

Integration over the interface leads to the mortar coupling matrices \mathcal{D} and \mathcal{M} associated with the slave and the master side, respectively, cf. Section 3.2.4. The biorthogonality condition plays a major role in the evaluation of \mathcal{D} and \mathcal{M} since it leads to a purely diagonal form of \mathcal{D} . Hence, the inversion of \mathcal{D} will be computationally cheap enabling the condensation of the LAGRANGE multiplier field from the global system of linear equations as will be shown in Sections 3.3.1 and 3.3.2.

3.2.4 Fluid-Structure Interface

As discussed in Section 2.3, kinematic and dynamic coupling conditions need to be satisfied at the fluid-structure interface. Since at the interface, degrees of freedom of all involved fields come together, one has to deal with the kinematic coupling of structure, fluid, and ALE degrees of freedom resulting in two separate discrete kinematic coupling conditions, see Figure 3.1.

On the one hand, the evolution of the fluid interface motion, described by \mathbf{d}_Γ^g , has to be related to the fluid velocity \mathbf{u}_Γ^f at the interface. This coupling between fluid and ALE degrees of freedom takes place purely in the fluid domain and does not involve any structural degrees of freedom. On the other hand, a discrete version of the kinematic continuity constraint (2.12c) has to be

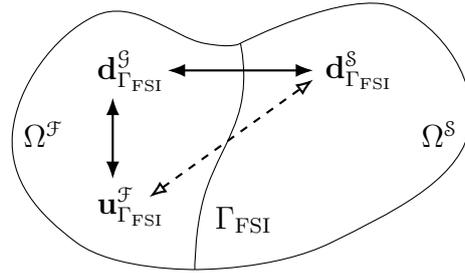


Figure 3.1: Illustration of discrete kinematic interface coupling conditions as given by MAYR *et al.* [163] — The conversion of interface fluid velocity degrees of freedom $\mathbf{u}_{\Gamma_{\text{FSI}}}^{\mathcal{F}}$ into interface ALE displacement degrees of freedom $\mathbf{d}_{\Gamma_{\text{FSI}}}^{\mathcal{G}}$ happens internally in the fluid field and does not include the mortar coupling across the fluid-structure interface. The mortar coupling itself involves structure and ALE displacement degrees of freedom $\mathbf{d}_{\Gamma_{\text{FSI}}}^{\mathcal{S}}$ and $\mathbf{d}_{\Gamma_{\text{FSI}}}^{\mathcal{G}}$, respectively. Combining these two couplings, that are illustrated by solid arrows, yields the FSI coupling of interface fluid velocity degrees of freedom $\mathbf{u}_{\Gamma_{\text{FSI}}}^{\mathcal{F}}$ and interface structure displacement degrees of freedom $\mathbf{d}_{\Gamma_{\text{FSI}}}^{\mathcal{S}}$, that is indicated by the dashed arrow.

provided for the 'meshtying' problem at the interface whereby the ALE deformation $\mathbf{d}_{\Gamma}^{\mathcal{G}}$ needs to be associated with the structural deformation $\mathbf{d}_{\Gamma}^{\mathcal{S}}$ at the interface. Thereby, two sets of degrees of freedom which are separated by the interface are coupled and, thus, the mortar coupling will play an important role. Both discrete kinematic coupling conditions can finally be combined to establish a link between fluid interface velocities $\mathbf{u}_{\Gamma}^{\mathcal{F}}$ and structural interface displacements $\mathbf{d}_{\Gamma}^{\mathcal{S}}$ leading to the discrete representation of the no-slip condition (2.12a). The connections and dependencies of these kinematic coupling conditions are illustrated in Figure 3.1 and will be discussed in the following sections.

In the time-continuous regime, interface coupling conditions as depicted in Figure 3.1 are fully equivalent. However, after time discretization only two out of three can be satisfied exactly in general since time derivatives are computed differently in solid, fluid, and ALE field due to different time integration schemes in each field. In this thesis, it is aimed at satisfying discrete versions of (2.12c) and (2.3), i.e. matching of interface displacements and matching of interface fluid and grid velocities, leaving mismatches in the equivalency of interface velocities in every time step. The fluid interface velocities $\mathbf{u}_{\Gamma}^{\mathcal{F}}$ evolve as primary solution to the NAVIER–STOKES equations, while the solid interface velocities $\dot{\mathbf{d}}_{\Gamma}^{\mathcal{S}}$ are computed based on the solid interface displacements $\mathbf{d}_{\Gamma}^{\mathcal{S}}$, which match the grid interface displacements $\mathbf{d}_{\Gamma}^{\mathcal{G}}$, and do not evolve by themselves. Thus, these possible mismatches in the equivalency of interface velocities cannot accumulate over time.

For the discretization of the interface traction equilibrium (2.13), the mortar method will come into play since the LAGRANGE multiplier field $\underline{\lambda}$ can be interpreted as interface traction field. In addition, a temporal interpolation of interface tractions will be introduced that allows for the freedom of choosing fluid and structure time integration schemes independently and tailored to the needs of the specific field.

Discrete Coupling Condition of Fluid Velocities and ALE Displacements

In the thesis by FÖRSTER [81], exact conservation of the volume of the fluid domain $\Omega^{\mathcal{F}}$ is achieved by deriving a conversion of interface fluid velocities and interface ALE displacements that is consistent with the fluid time integration scheme. The fluid time integration schemes used in this thesis can be assumed to have an averaged acceleration in one time step $t_n \rightarrow t_{n+1} = t_n + \Delta t_n$. The deformation of the fluid domain can be computed by integrating this averaged acceleration twice w.r.t. time. This yields a second order accurate transformation of interface fluid velocities and interface ALE displacements

$$\mathbf{d}_{\Gamma,n+1}^{\mathcal{G}} - \mathbf{d}_{\Gamma,n}^{\mathcal{G}} = \frac{\Delta t_n}{2} (\mathbf{u}_{\Gamma,n+1}^{\mathcal{F}} + \mathbf{u}_{\Gamma,n}^{\mathcal{F}}) \quad (3.20)$$

which can be identified as the *trapezoidal rule*. It correctly preserves the volume of the fluid domain $\Omega^{\mathcal{F}}$ and, thus, satisfies and extends the geometric conservation law towards the interface. Despite this desirable property, the trapezoidal rule may tend to oscillations of the velocity over time. Hence, it is sometimes replaced by the dissipative *backward EULER scheme*

$$\mathbf{d}_{\Gamma,n+1}^{\mathcal{G}} - \mathbf{d}_{\Gamma,n}^{\mathcal{G}} = \Delta t_n \mathbf{u}_{\Gamma,n+1}^{\mathcal{F}}. \quad (3.21)$$

To enable the inclusion of (3.20) or (3.21) into the global monolithic system of equations, both schemes have to be expressed in incremental form. After introducing the KRONECKER delta

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

this is done by recasting them into the form

$$\Delta \mathbf{d}_{\Gamma,n+1}^{\mathcal{G},k+1} = \tau \Delta \mathbf{u}_{\Gamma,n+1}^{\mathcal{F},k+1} + \delta_{i0} \Delta t_n \mathbf{u}_{\Gamma,n}^{\mathcal{F}}, \quad (3.22)$$

where the parameter

$$\tau = \begin{cases} \frac{\Delta t_n}{2} & \text{for trapezoidal rule (3.20)} \\ \Delta t_n & \text{for backward EULER scheme (3.21)} \end{cases} \quad (3.23)$$

determines the applied scheme.

When using the backward EULER scheme, which is first order accurate only, the overall temporal convergence rate of the coupled solver is expected to deteriorate to first order. If the trapezoidal rule is applied, second order convergence can be achieved if the fluid and structure time integration schemes are chosen as second order accurate. This will be shown in Section 3.4.1, where temporal convergence rates of the proposed monolithic FSI solver are studied.

Discrete Coupling Condition for Solid and ALE Displacements

A discrete representation of the kinematic interface coupling conditions (2.12) needs to be established. This is done by means of the dual mortar method. Its first application to interface constraint enforcement in the context of FSI has been done by KLÖPPEL *et al.* [138]. While KLÖPPEL [137] performs the discretization starting from the velocity constraint (2.12a),

the displacement constraint (2.12c) is used as starting point in this thesis. Both procedures result in a discrete coupling of interface displacements and in exactly the same expressions for the mortar coupling matrices \mathcal{D} and \mathcal{M} . Though, the latter approach seems to be more straightforward to prepare assembly of the monolithic system of equations.

Inserting the spatial discretizations (3.11), (3.15), and (3.17) for the ALE displacement, the structural displacement, and the LAGRANGE multiplier field into the weak kinematic coupling condition (3.4c) yields

$$\begin{aligned}
 & \left(\sum_{j=1}^{n^{\text{sl}}} \Phi_j \boldsymbol{\mu}_j, \sum_{k=1}^{n^\gamma} N_k^\gamma \mathbf{d}_k^{\mathcal{S},\gamma} - \sum_{l=1}^{n^\varepsilon} N_l^\varepsilon \mathbf{d}_l^{\mathcal{G},\varepsilon} \right)_\Gamma \\
 &= \sum_{j=1}^{n^{\text{sl}}} \boldsymbol{\mu}_j^\top \cdot \left[\sum_{k=1}^{n^\gamma} \int_\Gamma \Phi_j N_k^\gamma d\Gamma \mathbf{d}_k^{\mathcal{S},\gamma} - \sum_{l=1}^{n^\varepsilon} \int_\Gamma \Phi_j N_l^\varepsilon d\Gamma \mathbf{d}_l^{\mathcal{G},\varepsilon} \right] \\
 &= \sum_{j=1}^{n^{\text{sl}}} \boldsymbol{\mu}_j^\top \cdot \left[\mathbf{C}_{\text{SF}}[j, k] \mathbf{d}_k^{\mathcal{S},\gamma} - \mathbf{C}_{\text{FS}}[j, l] \mathbf{d}_l^{\mathcal{G},\varepsilon} \right] = 0 \quad \forall \boldsymbol{\mu}_j \neq \mathbf{0}
 \end{aligned} \tag{3.24}$$

with $\gamma, \varepsilon \in \{\text{ma}, \text{sl}\}$, $\gamma \neq \varepsilon$ and n^{sl} being the number of slave nodes. Using $\mathbf{J}_{n^{\text{dim}}} \in \mathbb{R}^{n^{\text{dim}} \times n^{\text{dim}}}$ to denote an identity matrix with n^{dim} being the spatial dimension, i.e. $n^{\text{dim}} \in \{2, 3\}$, the nodal coupling matrices

$$\begin{aligned}
 \mathbf{C}_{\text{SF}}[j, k] &= \mathbf{C}_{\text{SF}}^{jk} \mathbf{J}_{n^{\text{dim}}} = \int_\Gamma \Phi_j N_k^\gamma d\Gamma \mathbf{J}_{n^{\text{dim}}}, \\
 \mathbf{C}_{\text{FS}}[j, l] &= \mathbf{C}_{\text{FS}}^{jl} \mathbf{J}_{n^{\text{dim}}} = \int_\Gamma \Phi_j N_l^\varepsilon d\Gamma \mathbf{J}_{n^{\text{dim}}}
 \end{aligned}$$

have been introduced. They can be assembled to the coupling matrices \mathbf{C}_{SF} and \mathbf{C}_{FS} on system level. For example, if the structure field is chosen as the slave field, i.e. $\gamma = \text{sl}$, the biorthogonality condition (3.18) is employed to write

$$\mathbf{C}_{\text{SF}} = \sum_{k=1}^{n^\gamma} \int_\Gamma \Phi_j N_k^{\text{sl}} d\Gamma = \sum_{k=1}^{n^\gamma} \delta_{jk} \int_\Gamma N_k^{\text{sl}} d\Gamma.$$

When choosing the fluid field as the slave field, the coupling matrix \mathbf{C}_{FS} takes a diagonal form in an analogous way. Association of \mathbf{C}_{SF} and \mathbf{C}_{FS} with the mortar matrices \mathcal{D} and \mathcal{M} depends on the actual choice of master and slave side. Possible choices will be discussed in Sections 3.3.1 and 3.3.2. Full details on the numerical evaluation of the mortar integrals are given in [189, 191, 192]. Different numerical integration schemes have been compared by FARAH *et al.* [68]. A consistent linearization is given by POPP *et al.* [186].

Since (3.24) needs to hold for all values of $\boldsymbol{\mu}$, the kinematic coupling residual can be expressed in terms of the global coupling matrices \mathbf{C}_{SF} and \mathbf{C}_{FS} , reading

$$\mathbf{r}^{\text{coupl}} = \mathbf{C}_{\text{SF}} \mathbf{d}_\Gamma^{\mathcal{S}} - \mathbf{C}_{\text{FS}} \mathbf{d}_\Gamma^{\mathcal{G}}.$$

Its linearization yields the kinematic coupling constraint

$$\mathbf{C}_{\text{SF}} \Delta \mathbf{d}_{\Gamma, n+1}^{\mathcal{S}, k+1} - \mathbf{C}_{\text{FS}} \Delta \mathbf{d}_{\Gamma, n+1}^{\mathcal{G}, k+1} = -\delta_{i0} \mathbf{C}_{\text{SF}} \Delta \mathbf{d}_{\Gamma, p}^{\mathcal{S}} \tag{3.26}$$

formulated in incremental form. The violation of the interface continuity requirement due to possible non-constant predictors is measured by $\Delta \mathbf{d}_{\Gamma,p}^s$ and accounted for by the right-hand side term, which is necessary only in the first nonlinear iteration step $k = 0$ denoted by the KRONECKER delta symbol δ_{i0} . Due to the linearity of the kinematic coupling condition, the kinematic interface continuity requirement is guaranteed to be satisfied for all nonlinear iteration steps $k > 0$.

Note that coupling is performed in the material configuration. The coupling matrices \mathbf{C}_{SF} and \mathbf{C}_{FS} depend solely on the undeformed configuration, but not on the actual deformation. Thus, they do not need to be re-evaluated during the simulation, but can be reused [138]. However, in scenarios like computational contact mechanics or in the FSI approach by KLÖPP-PEL [137], that allows for relative motion of fluid and solid meshes between two time steps, an update of the mortar matrices is necessary as soon as the configuration has changed. See the comparison by FARAH *et al.* [68] for some efficient techniques for an accurate evaluation of the mortar integrals, which become essential in such update scenarios.

Conversion of Fluid Velocities and Solid Displacements

With the discrete coupling conditions (3.22) and (3.26), all necessary conditions are at hand to assemble the global monolithic system. However, it is desirable to derive a direct conversion of fluid velocities and structural displacements at the fluid-structure interface by replacing the interface ALE displacements. On the one hand, this emphasizes the fact that the ALE field is not a physical field but rather an auxiliary field to describe the fluid motion. On the other hand, this eases the notation of the global monolithic system when it comes to choosing master and slave side in the context of the dual mortar method.

Combination of (3.22) and (3.26) results in

$$\mathbf{C}_{SF} \Delta \mathbf{d}_{\Gamma,n+1}^{s,k+1} + \delta_{i0} \mathbf{C}_{SF} \Delta \mathbf{d}_{\Gamma,p}^s = \tau \mathbf{C}_{FS} \Delta \mathbf{u}_{\Gamma,n+1}^{\mathcal{F},k+1} + \delta_{i0} \Delta t_n \mathbf{C}_{FS} \mathbf{u}_{\Gamma,n}^{\mathcal{F}}. \quad (3.27)$$

Note that (3.27) does not take the role of an additional coupling condition. It is just a redundant reformulation of (3.22) and (3.26). A graphical representation of (3.27) is given by the dashed arrow in Figure 3.1.

Contributions to the Balances of Linear Momentum

Finally, the last two terms in the weak form, which are related to the interface tractions, are dealt with. Inserting the spatial discretizations (3.11b) and (3.17) of the fluid velocity field and the LAGRANGE multiplier field into the interface traction term of the fluid's weak form (3.4a) results in

$$\begin{aligned} (\underline{\mathbf{v}}^{\mathcal{F}}, \underline{\boldsymbol{\lambda}})_{\Gamma} &\approx \left(\sum_{l=1}^{n^{\varepsilon}} N_l^{\varepsilon} \mathbf{v}_l^{\mathcal{F},\varepsilon}, \sum_{j=1}^{n^{\text{nd,sl}}} \Phi_j \boldsymbol{\lambda}_j \right)_{\Gamma} = \sum_{l=1}^{n^{\varepsilon}} \left(\mathbf{v}_l^{\mathcal{F},\varepsilon} \right)^{\text{T}} \cdot \left[\sum_{j=1}^{n^{\text{nd,sl}}} \int_{\Gamma} N_l^{\varepsilon} \Phi_j \, \text{d}\Gamma \boldsymbol{\lambda}_j \right] \\ &= \sum_{l=1}^{n^{\varepsilon}} \left(\mathbf{v}_l^{\mathcal{F},\varepsilon} \right)^{\text{T}} \cdot \mathbf{C}_{FS}^{\text{T}}[j, l] \boldsymbol{\lambda}_j. \end{aligned} \quad (3.28a)$$

Similarly, a discrete representation of the respective term of the solid's weak form (3.4b) yields

$$\begin{aligned} -(\underline{\mathbf{w}}^s, \underline{\boldsymbol{\lambda}})_\Gamma &\approx - \left(\sum_{k=1}^{n^\gamma} N_k^\gamma \mathbf{w}_k^{s,\gamma}, \sum_{j=1}^{n^{\text{nd,sl}}} \Phi_j \boldsymbol{\lambda}_j \right)_\Gamma = - \sum_{k=1}^{n^\gamma} \left(\mathbf{w}_k^{s,\gamma} \right)^\text{T} \cdot \left[\sum_{j=1}^{n^{\text{nd,sl}}} \int_\Gamma N_k^\gamma \Phi_j \, \text{d}\Gamma \boldsymbol{\lambda}_j \right] \\ &= - \sum_{k=1}^{n^\gamma} \left(\mathbf{w}_k^{s,\gamma} \right)^\text{T} \cdot \mathbf{C}_{\text{SF}}^\text{T}[j, k] \boldsymbol{\lambda}_j, \end{aligned} \quad (3.28b)$$

where the discretizations (3.15) and (3.17) of the solid displacement field and the LAGRANGE multiplier field have been used. In both expressions (3.28), the transposes of the nodal blocks of the coupling matrices \mathbf{C}_{FS} and \mathbf{C}_{SF} can be found. The number of slave side interface nodes is denoted by $n^{\text{nd,sl}}$. Furthermore, $\gamma, \varepsilon \in \{\text{ma}, \text{sl}\}$ and $\gamma \neq \varepsilon$.

As already indicated during discretization of both the fluid and the structure field in Sections 3.2.1 and 3.2.2, each field-specific time integration scheme evaluates the dynamic equilibrium at an intermediate point $t_m^\mathcal{F}, t_m^\mathcal{S} \in]t_n, t_{n+1}]$ with possibly

$$t_m^\mathcal{F} \neq t_m^\mathcal{S}. \quad (3.29)$$

To impose equilibrium of interface traction fields weakly, the interface residual expressions of the fluid and the structure field are extracted from the discrete weak residual expressions of the respective field. They need to be evaluated exactly at these mid-point instances in time and, thus, read

$$\mathbf{r}_{\boldsymbol{\lambda},m}^\mathcal{F} = \mathbf{C}_{\text{FS}}^\text{T} \boldsymbol{\lambda}_m^\mathcal{F}, \quad (3.30a)$$

$$\mathbf{r}_{\boldsymbol{\lambda},m}^\mathcal{S} = -\mathbf{C}_{\text{SF}}^\text{T} \boldsymbol{\lambda}_m^\mathcal{S}, \quad (3.30b)$$

where (3.28) is used to express them in terms of the coupling matrices \mathbf{C}_{FS} and \mathbf{C}_{SF} and the discrete LAGRANGE multiplier field $\boldsymbol{\lambda}$. Since the coupling matrices are purely based on geometry and do not depend on time, only the interface traction needs to be incorporated at $t_m^\mathcal{F}$ and $t_m^\mathcal{S}$ reading $\boldsymbol{\lambda}_m^\mathcal{F}$ and $\boldsymbol{\lambda}_m^\mathcal{S}$, respectively. Having in mind the goal of choosing the time integration schemes for fluid and solid independently and recalling (3.29) generally yields

$$\boldsymbol{\lambda}_m^\mathcal{F} \neq \boldsymbol{\lambda}_m^\mathcal{S}. \quad (3.31)$$

Thus, the single field time integration schemes need to be taken into account when exchanging interface tractions between the fields. As usual for fully implicit, single-step, single-stage time integration schemes used in this work, linear interpolations are applied to evaluate $\mathbf{r}_{\boldsymbol{\lambda},m}^\mathcal{F}$ and $\mathbf{r}_{\boldsymbol{\lambda},m}^\mathcal{S}$ at the intermediate time instances $t_m^\mathcal{F}$ and $t_m^\mathcal{S}$, respectively. Then, the coupling terms (3.30) can be expressed in terms of known nodal quantities $\boldsymbol{\lambda}_n$ and unknown nodal quantities $\boldsymbol{\lambda}_{n+1}^k$ reading

$$\mathbf{r}_{\boldsymbol{\lambda},m}^{\mathcal{F},k} = \mathbf{C}_{\text{FS}}^\text{T} (b\boldsymbol{\lambda}_n + (1-b)\boldsymbol{\lambda}_{n+1}^k), \quad (3.32a)$$

$$\mathbf{r}_{\boldsymbol{\lambda},m}^{\mathcal{S},k} = -\mathbf{C}_{\text{SF}}^\text{T} (a\boldsymbol{\lambda}_n + (1-a)\boldsymbol{\lambda}_{n+1}^k) \quad (3.32b)$$

with interpolation weights a and b of the solid and fluid field, respectively. The actual values of a and b depend on the specific field time integrators. They are always chosen equal to the

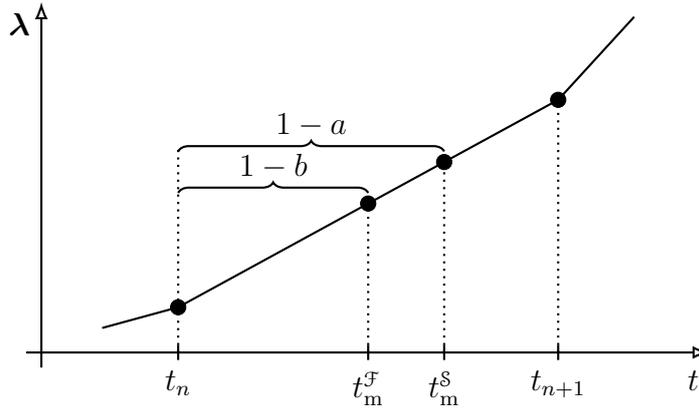


Figure 3.2: Temporal interpolation of the interface traction field inspired by [132] — The interface traction λ is interpolated to the respective generalized midpoints t_m^s and t_m^f of the structural and fluid time integrator, respectively. The interpolation weights a and b are chosen to match the time integration weights associated with quantities at time t_n .

weighting of the previous solution in (3.32). For example, when using generalized- α time integration [39] for the structure field and generalized- α time integration [129] for the fluid field, time interpolation factors have to be chosen as $a = \alpha_f^s$ and $b = 1 - \alpha_f^f$. In the case of one-step- θ , they are chosen as $a = \theta^s$ and $b = \theta^f$. An illustration of the temporal interpolation of interface traction is shown in Figure 3.2.

In (3.32), a known contribution λ_n from the last time step t_n always occurs on the right-hand side in both fluid and structure field interface equations, whereas λ_{n+1}^k is an additional unknown and has to be solved for. Since the coupling terms in the weak forms (3.4a) and (3.4b) are linear in the LAGRANGE multiplier field $\underline{\lambda}$, cf. also (3.28), linearizations of the residual terms (3.32) are just the coupling matrices themselves wherein the temporal interpolation factors occur, too:

$$\frac{\partial \mathbf{r}_{\lambda, m}^{\mathcal{F}, k}}{\partial \lambda_{n+1}^k} = (1 - b) \mathbf{c}_{\text{FS}}^T, \quad \frac{\partial \mathbf{r}_{\lambda, m}^{\mathcal{S}, k}}{\partial \lambda_{n+1}^k} = -(1 - a) \mathbf{c}_{\text{SF}}^T. \quad (3.33)$$

Although the evaluation of interface traction at the generalized midpoints t_m^s and t_m^f has already been suggested by KLÖPPEL [137] in the context of monolithic solvers, it has been implemented for the first time in the present contribution. A similar interpolation strategy has previously been proposed by JOOSTEN *et al.* [132] for partitioned solution schemes, cf. Remark 3.2.4.

Remark 3.2.4 For partitioned FSI schemes, the case of $t_m^s \neq t_m^f$ has been studied by JOOSTEN *et al.* [132]. Therein, a stability analysis of a model problem with generalized- α time integration in fluid and solid is performed. They conclude that stability and accuracy of the coupling scheme can be maintained for all choices of spectral radii ρ_∞^s and ρ_∞^f if and only if a temporal interpolation of the interface traction field as shown in Figure 3.2 is applied. With that temporal interpolation of interface traction, the spectral radii can be chosen from the entire range $0 \leq \rho_\infty^s \leq 1$ and $0 \leq \rho_\infty^f \leq 1$ while resulting in a stable and accurate solution.

Although choices of the spectral radius close or equal to zero are interesting from a theoretical point of view when it comes to the analysis of high-frequency dissipation [39, 129], they are usually limited to the reasonable range $\rho_\infty^s \geq 0.5$ and $\rho_\infty^f \geq 0.5$ for practical simulations. These bounds are also imposed by the research code `Baci` [243] used in this thesis. Otherwise, parameters α_f and α_m are in the theoretically admissible range, but such that the inertia contributions to the dynamic equilibrium are taken from outside the actual time interval of interest.

In the analysis by JOOSTEN et al. [132], the instabilities without interpolation of the interface traction occur only in the case of $\rho_\infty^s < \rho_\infty^f$ with $\rho_\infty^s < 0.5$. Thus, their theoretical meaning is perceived, but their impact on practical applications is considered as rather small since such parameter choices are usually avoided.

Energy Considerations at the Fluid-Structure Interface

To assess a possible production of artificial energy due to time integration, a similar line of arguments as in the thesis by MOK [170], which dealt with partitioned solution schemes, is followed. Such an analysis for monolithic FSI solvers has already been performed by MAYR *et al.* [163]. The amount of energy production per time step at the fluid-structure interface can be computed by balancing the work of the interface traction field along the motion of the solid and the fluid side of the interface reading

$$\Delta E_\Gamma^{n \rightarrow n+1} = E_\Gamma^{s, n \rightarrow n+1} + E_\Gamma^{f, n \rightarrow n+1}. \quad (3.34)$$

The work contributions related to the solid and fluid interface motion are denoted by $E_\Gamma^{s, n \rightarrow n+1}$ and $E_\Gamma^{f, n \rightarrow n+1}$, respectively, and should cancel each other out to obtain an energy-stable scheme. They are defined as

$$E_\Gamma^{s, n \rightarrow n+1} = \int_\Gamma \lambda_m^s \cdot (\mathbf{d}_{\Gamma, n+1}^s - \mathbf{d}_{\Gamma, n}^s) \, d\Gamma, \quad (3.35a)$$

$$E_\Gamma^{f, n \rightarrow n+1} = \int_\Gamma \lambda_m^f \cdot (\mathbf{d}_{\Gamma, n+1}^f - \mathbf{d}_{\Gamma, n}^f) \, d\Gamma, \quad (3.35b)$$

where the interface traction needs to be evaluated at the field's time integration midpoint t_m^s and t_m^f , respectively. Inserting (3.35) into (3.34) and using the temporal interpolation (3.32) yields

$$\begin{aligned} \Delta E_\Gamma^{n \rightarrow n+1} &= \int_\Gamma (a\lambda_n + (1-a)\lambda_{n+1}) \cdot (\mathbf{d}_{\Gamma, n+1}^s - \mathbf{d}_{\Gamma, n}^s) \, d\Gamma \\ &\quad - \int_\Gamma (b\lambda_n + (1-b)\lambda_{n+1}) \cdot (\mathbf{d}_{\Gamma, n+1}^f - \mathbf{d}_{\Gamma, n}^f) \, d\Gamma. \end{aligned}$$

Exploiting the discrete coupling condition (3.26), namely the exact equality of solid and fluid interface displacements in the time-discrete setting, the amount of energy production per time step is finally given as

$$\Delta E_\Gamma^{n \rightarrow n+1} = \int_\Gamma ((a-b)\lambda_n + (b-a)\lambda_{n+1}) \cdot (\mathbf{d}_{\Gamma, n+1}^s - \mathbf{d}_{\Gamma, n}^s) \, d\Gamma \quad (3.36)$$

$$= (a-b) \int_\Gamma (\lambda_n - \lambda_{n+1}) \cdot (\mathbf{d}_{\Gamma, n+1}^s - \mathbf{d}_{\Gamma, n}^s) \, d\Gamma. \quad (3.37)$$

contains the primary unknowns of each field as well as the LAGRANGE multiplier field. Finally, the residual vector is given by

$$\mathbf{r}_{n+1}^{\text{FSI},k} = \begin{bmatrix} \mathbf{r}_I^S \\ \mathbf{r}_I^S \\ \mathbf{r}_I^F \\ \mathbf{r}_I^F \\ \mathbf{r}_I^G \\ \mathbf{0} \end{bmatrix}_{n+1}^k + \begin{bmatrix} \mathbf{0} \\ -a\mathbf{c}_{\text{SF}}^T \boldsymbol{\lambda}_n \\ \mathbf{0} \\ b\mathbf{c}_{\text{FS}}^T \boldsymbol{\lambda}_n \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \delta_{k0} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \Delta t_n \mathbf{c}_{\text{FS}} \mathbf{u}_{\Gamma,n}^F - \mathbf{c}_{\text{SF}} \Delta \mathbf{d}_{\Gamma,p}^S \end{bmatrix}. \quad (3.40c)$$

Due to the 0-block on the main diagonal, the global monolithic system of linear equations (3.40) is of saddle-point type. In order to circumvent the saddle-point like system to be able to use the efficient FSI-specific preconditioners detailed in Chapter 5, the unknown LAGRANGE multipliers $\boldsymbol{\lambda}_{n+1}$ will be condensed, yielding a problem with structural displacement, fluid velocity and pressure as well as ALE grid displacement degrees of freedom as the only unknowns. By exploitation of the kinematic coupling conditions derived in Section 3.2.4, one can condense the interface degrees of freedom of the slave side from the global system of equations. The slave side's interface balance of linear momentum will be used to perform the static condensation of the LAGRANGE multipliers. The application of the dual mortar methods is crucial to enable static condensation. If standard shape functions are used to discretize the LAGRANGE multiplier field, the condensation is numerically very costly or even unfeasible. Then, the saddle-point type system needs to be solved with appropriate saddle-point solvers. However, this is beyond the scope of this thesis. An introduction to and survey of saddle-point solver is given by BENZI *et al.* [20], for example.

System (3.40) does not include a specific choice of master and slave side, yet. In addition, the coupling of ALE interface displacements \mathbf{d}_I^G to the motion of the fluid-structure interface Γ_{FSI} is omitted at this point, although the coupling of ALE interface displacements \mathbf{d}_I^G to the fluid interface velocities \mathbf{u}_I^F has already been established in (3.22). To finally close system (3.40), the required relation will be added after master and slave side have been chosen. This procedure allows to express the interface motion in terms of master side's interface degrees of freedom only, which will be a good starting point for the condensation of the LAGRANGE multiplier field. For both possible choices of master and slave side, the missing coupling conditions as well as the process of condensation will be detailed in Sections 3.3.1 and 3.3.2, respectively.

When static condensation has been completed, the interface motion is purely described and handled in terms of unknowns of the master field. Thus, two algorithmic variants are distinguished, namely *fluid-handled interface motion* and *structure-handled interface motion*. They will be discussed in the subsequent sections. Section 3.3.1 deals with the variant, where the fluid field has been chosen as the master field, while Section 3.3.2 describes the case of the structure field being the master field. For both choices, the process of static condensation will be detailed that leads to the final system of linear equations to be implemented.

3.3.1 Fluid-handled Interface Motion

First, the variant where the fluid field is chosen as the master field and the structure field as the slave field is considered. Hence, the interface motion is expressed in terms of fluid velocity

From the balance of linear momentum of the structural interface degrees of freedom together with (3.42), the unknown LAGRANGE multipliers are expressed by

$$\begin{aligned} \boldsymbol{\lambda}_{n+1} = & -\frac{a}{1-a}\boldsymbol{\lambda}_n + \frac{1}{1-a}\mathbf{D}^{-\text{T}} \left(\mathbf{r}_{\Gamma,n+1}^{\text{S}} + \mathbf{S}_{\Gamma\text{I}}\Delta\mathbf{d}_{\text{I},n+1}^{\text{S},k+1} + \tau\mathbf{S}_{\Gamma\Gamma}\mathcal{P}\Delta\mathbf{u}_{\Gamma,n+1}^{\text{F},k+1} \right) \\ & + \delta_{k0}\frac{1}{1-a}\mathbf{D}^{-\text{T}} \left(\Delta t_n\mathbf{S}_{\Gamma\Gamma}\mathcal{P}\mathbf{u}_{\Gamma,n}^{\text{F}} - \mathbf{S}_{\Gamma\Gamma}\Delta\mathbf{d}_{\Gamma,\text{p}}^{\text{S}} \right). \end{aligned} \quad (3.45)$$

Equation (3.45) is used to recover the LAGRANGE multiplier solution at the end of each time step as a postprocessing step.

Using (3.42), (3.44), and (3.45), condensation of the system of equations (3.41) can be performed. The condensed linear system with fluid-handled interface motion consists of the Jacobian matrix

$$\mathcal{J}_{n+1}^k = \begin{bmatrix} \mathbf{S}_{\text{II}} & & \tau\mathbf{S}_{\text{IF}}\mathcal{P} & & \\ & \mathcal{F}_{\text{II}} & \mathcal{F}_{\text{IF}} + \tau\mathcal{F}_{\text{IF}}^{\text{G}} & \mathcal{F}_{\text{II}}^{\text{G}} & \\ \frac{1-b}{1-a}\mathcal{P}^{\text{T}}\mathbf{S}_{\Gamma\text{I}} & \mathcal{F}_{\Gamma\text{I}} & \mathcal{F}_{\Gamma\Gamma} + \tau\mathcal{F}_{\Gamma\Gamma}^{\text{G}} + \frac{1-b}{1-a}\tau\mathcal{P}^{\text{T}}\mathbf{S}_{\Gamma\Gamma}\mathcal{P} & \mathcal{F}_{\Gamma\text{I}}^{\text{G}} & \\ & & \tau\mathcal{A}_{\text{IF}} & \mathcal{A}_{\text{II}} & \\ & & & & \end{bmatrix}_{n+1}^k, \quad (3.46a)$$

the solution increment vector

$$\Delta\mathbf{x}_{n+1}^{k+1\text{T}} = \begin{bmatrix} \Delta\mathbf{d}_{\text{I}}^{\text{S}\text{T}} & \Delta\mathbf{u}_{\text{I}}^{\text{F}\text{T}} & \Delta\mathbf{u}_{\Gamma}^{\text{F}\text{T}} & \Delta\mathbf{d}_{\text{I}}^{\text{G}\text{T}} \end{bmatrix}_{n+1}^{k+1}, \quad (3.46b)$$

and the residual vector

$$\begin{aligned} \mathbf{r}_{n+1}^k = & \begin{bmatrix} \mathbf{r}_{\text{I}}^{\text{S}} \\ \mathbf{r}_{\text{I}}^{\text{F}} \\ \mathbf{r}_{\Gamma}^{\text{F}} + \frac{1-b}{1-a}\mathcal{P}^{\text{T}}\mathbf{r}_{\Gamma}^{\text{S}} \\ \mathbf{r}_{\Gamma}^{\text{G}} \end{bmatrix}_{n+1}^k + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \left(b - \frac{a(1-b)}{1-a} \right) \mathcal{M}^{\text{T}}\boldsymbol{\lambda}_n \\ \mathbf{0} \end{bmatrix} \\ & + \delta_{k0} \begin{bmatrix} \Delta t_n\mathbf{S}_{\text{IF}}\mathcal{P}\mathbf{u}_{\Gamma,n}^{\text{F}} - \mathbf{S}_{\text{IF}}\Delta\mathbf{d}_{\Gamma,\text{p}}^{\text{S}} \\ \Delta t_n\mathcal{F}_{\text{IF}}^{\text{G}}\mathbf{u}_{\Gamma,n}^{\text{F}} \\ \Delta t_n\mathcal{F}_{\Gamma\Gamma}^{\text{G}}\mathbf{u}_{\Gamma,n}^{\text{F}} + \frac{1-b}{1-a}\Delta t_n\mathcal{P}^{\text{T}}\mathbf{S}_{\Gamma\Gamma}\mathcal{P}\mathbf{u}_{\Gamma,n}^{\text{F}} - \frac{1-b}{1-a}\mathcal{P}^{\text{T}}\mathbf{S}_{\Gamma\Gamma}\Delta\mathbf{d}_{\Gamma,\text{p}}^{\text{S}} \\ \Delta t_n\mathcal{A}_{\text{IF}}\mathbf{u}_{\Gamma,n}^{\text{F}} \end{bmatrix}. \end{aligned} \quad (3.46c)$$

Remark 3.3.2 According to PUSO and LAURSEN [190], essential boundary conditions on the mortar interface require special treatment in order to circumvent stability problems. From the three remedies proposed by PUSO [190], we adopt the following strategy: Apply essential boundary conditions on the master side of the interface only. Due to the mortar coupling, they will be effectively enforced weakly on the slave side of the mortar interface. In case of fluid-handled interface motion, this means that at the interface only fluid degrees of freedom are allowed to be subject to DIRICHLET boundary conditions while the boundary condition will be imposed weakly on the structure degrees of freedom via the mortar coupling.

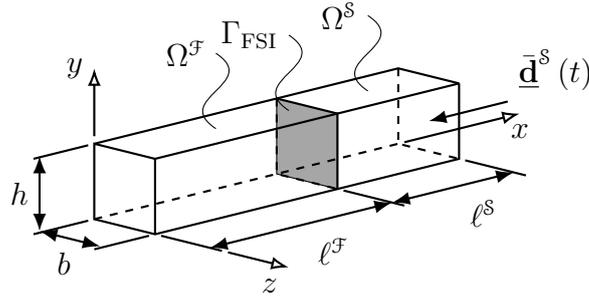


Figure 3.3: Geometry of a pseudo one-dimensional FSI problem with analytical solution — The structural block Ω^S moves in x -direction due to a time-dependent DIRICHLET boundary condition $\bar{\mathbf{d}}^S(t)$ at $x = \ell^F + \ell^S$. Thus, fluid is pushed out or sucked in across the traction-free NEUMANN boundary at $x = 0$. All movement in y - and z -direction is suppressed, leaving a pseudo one-dimensional problem. The dimensions are given as: $h = 1, b = 1, \ell^F = 3, \ell^S = 2$.

3.4 Discussion of the Formulation using a Pseudo One-Dimensional FSI Example

For demonstration and discussion of essential properties of the present formulation, a simplified example is studied in detail. It mimics one-dimensional behavior and, thus, allows for an analytical solution. A sketch of the problem is given in Figure 3.3. The reduction to one dimension is mimicked by constraining any lateral motion in y - and z -direction with DIRICHLET boundary conditions. Actual constitutive parameters are of minimal importance here due to the simplicity of the example and, thus, are only given when necessary for the discussion.

Temporal convergence behavior of the proposed FSI scheme is studied in Section 3.4.1, while Section 3.4.2 sheds some light on the physical interpretation of the LAGRANGE multiplier field $\underline{\lambda}$ that is used to enforce the FSI coupling conditions.

A less detailed version of this example has previously been published by MAYR *et al.* [163].

3.4.1 Temporal Convergence Study

To study temporal convergence rates, first, an analytical solution is derived. Then, the relative L_2 -error of the velocity field at the end time $T = 1$ is computed as

$$e_{L_2}^{\mathcal{F}, \mathbf{u}} = \frac{\|\underline{\mathbf{u}}_{\text{ex}}^{\mathcal{F}} - \underline{\mathbf{u}}_{\text{h}}^{\mathcal{F}}\|_{L_2}}{\|\underline{\mathbf{u}}_{\text{ex}}^{\mathcal{F}}\|_{L_2}}$$

by comparison of the exact analytical solution $\underline{\mathbf{u}}_{\text{ex}}^{\mathcal{F}}$ and its numerical approximation $\underline{\mathbf{u}}_{\text{h}}^{\mathcal{F}}$ via the L_2 -norm

$$\|(\bullet)\|_{L_2} = \sqrt{\int_{\Omega} (\bullet)^2}. \quad (3.54)$$

Analogously, the L_2 -error $e_{L_2}^{\mathcal{F},p}$ of the pressure field can be computed by comparison of the exact pressure solution $p_{\text{ex}}^{\mathcal{F}}$ with its discrete counterpart $p_{\text{h}}^{\mathcal{F}}$. Finally, temporal convergence of the overall FSI formulation is assessed in terms of temporal convergence rates. A comparison to theoretically expected convergence rates is performed, that demonstrates optimal behavior of the proposed FSI solver.

Analytical Solution

Assuming rigidity of the solid body and prescribing its boundary motion $\bar{\mathbf{d}}^{\text{S}}(t)$ at $x = \ell^{\mathcal{F}} + \ell^{\text{S}}$, the displacement, velocity, and acceleration field of the solid domain Ω^{S} are given as

$$\underline{\mathbf{d}}^{\text{S}}(\underline{\mathbf{x}}, t) = \bar{\mathbf{d}}^{\text{S}}(t), \quad \dot{\underline{\mathbf{d}}}^{\text{S}}(\underline{\mathbf{x}}, t) = \dot{\bar{\mathbf{d}}}^{\text{S}}(t), \quad \ddot{\underline{\mathbf{d}}}^{\text{S}}(\underline{\mathbf{x}}, t) = \ddot{\bar{\mathbf{d}}}^{\text{S}}(t) \quad \forall \underline{\mathbf{x}} \in \Omega^{\text{S}} \wedge t \in (0, T). \quad (3.55)$$

The kinematic coupling conditions at the interface transfer the interface velocity from the solid to the fluid, reading

$$\underline{\mathbf{u}}_{\Gamma_{\text{FSI}}}^{\mathcal{F}}(t) = \dot{\bar{\mathbf{d}}}^{\text{S}}(t), \quad \dot{\underline{\mathbf{u}}}_{\Gamma_{\text{FSI}}}^{\mathcal{F}}(t) = \ddot{\bar{\mathbf{d}}}^{\text{S}}(t) \quad \text{on } \Gamma_{\text{FSI}} \times (0, T), \quad (3.56)$$

which finally, after exploiting incompressibility of the fluid flow field, yields the fluid velocity and acceleration fields

$$\underline{\mathbf{u}}^{\mathcal{F}}(\underline{\mathbf{x}}, t) = \underline{\mathbf{u}}_{\Gamma_{\text{FSI}}}^{\mathcal{F}}(t) = \dot{\bar{\mathbf{d}}}^{\text{S}}(t) \quad \forall \underline{\mathbf{x}} \in \Omega^{\mathcal{F}} \wedge t \in (0, T), \quad (3.57a)$$

$$\dot{\underline{\mathbf{u}}}^{\mathcal{F}}(\underline{\mathbf{x}}, t) = \dot{\underline{\mathbf{u}}}_{\Gamma_{\text{FSI}}}^{\mathcal{F}}(t) = \ddot{\bar{\mathbf{d}}}^{\text{S}}(t) \quad \forall \underline{\mathbf{x}} \in \Omega^{\mathcal{F}} \wedge t \in (0, T). \quad (3.57b)$$

Using the incompressibility condition (2.4b) and the restriction to one dimension as well as neglecting body forces $\underline{\mathbf{b}}^{\mathcal{F}}$, the balance of linear momentum (2.4a) of the NAVIER–STOKES equations reduces to

$$\rho^{\mathcal{F}} \frac{\partial u_x^{\mathcal{F}}}{\partial t} = - \frac{\partial p^{\mathcal{F}}}{\partial x} \quad \forall \underline{\mathbf{x}} \in \Omega^{\mathcal{F}} \wedge t \in (0, T).$$

After integration along the channel direction x , the fluid pressure solution is given as

$$p^{\mathcal{F}}(\underline{\mathbf{x}}, t) = -\rho^{\mathcal{F}} \frac{\partial u_x^{\mathcal{F}}}{\partial t} x + p_{\infty}^{\mathcal{F}}|_{x=0} \quad \forall \underline{\mathbf{x}} \in \Omega^{\mathcal{F}} \wedge t \in (0, T) \quad (3.58)$$

where the prescribed pressure at the free outlet is denoted by $p_{\infty}^{\mathcal{F}}|_{x=0}$.

Temporal Convergence Rates

For the temporal convergence study, the structure field is chosen as master field. Hence, structure interface degrees of freedom are subject to DIRICHLET boundary conditions, while fluid interface degrees of freedom are not allowed to carry DIRICHLET boundary conditions according to Remark 3.3.2. Besides the assumption of rigidity of the solid domain, all other constitutive parameters are of no importance.

For spatial discretization, linear hexahedral finite elements are used for the structure field as well as linear, equal-order interpolated, stabilized, and hexahedral finite elements are used

for the fluid field. Since displacement, velocity, and pressure solutions vary at most linearly in space, their spatial behavior can be captured exactly by the finite element approximation. Thus, a mesh refinement study has not to be performed for this example.

When choosing the imposed time-dependent DIRICHLET boundary condition $\bar{\mathbf{d}}^s(t)$ such that the analytical solution is also contained in the discrete temporal solution space, e.g. $\bar{\mathbf{d}}^s(t) = -t^2$, the analytical solution is fully recovered by the numerical scheme up to machine precision.

For this reason, a DIRICHLET boundary condition $\bar{\mathbf{d}}^s(t) = -t^5$ is prescribed on the structure in order to study temporal convergence behavior. The solution can still be fully recovered spatially, but the involved time integration schemes are not able to capture the temporal evolution exactly. Thus, temporal refinement is expected to reduce the error. As indicated above, the error is measured as L_2 -errors $e_{L_2}^{\mathcal{F},\mathbf{u}}$ and $e_{L_2}^{\mathcal{F},p}$ in the fluid velocity and pressure field, respectively.

The structure field employs the generalized- α time integration scheme [39] with a spectral radius $\rho_\infty^s = 1.0$, i.e. without artificial numerical dissipation. In the fluid field, either the generalized- α scheme [129] with various spectral radii $\rho_\infty^{\mathcal{F}}$ or the one-step- θ scheme [67] with various choices of $\theta^{\mathcal{F}}$ is employed. In addition, the conversion between fluid interface velocities $\mathbf{u}_\Gamma^{\mathcal{F}}$ and ALE interface displacements \mathbf{d}_Γ^s is varied between the trapezoidal rule and the backward EULER scheme as indicated in (3.20) and (3.21), respectively.

If second order accurate time integration schemes in both fluid and structure field and a second order accurate conversion between interface ALE displacements and fluid velocities is applied, the overall FSI algorithm is expected to be second order accurate in time. However, if only one of them is chosen as first order accurate, the overall order of temporal accuracy is expected to reduce to first order.

Temporal convergence plots for velocity and pressure in the fluid field $\Omega^{\mathcal{F}}$ are shown in Figure 3.4. In this figure, the parametrization of the fluid time integration schemes are detailed, where *Gen- α* and *OST* are shorthand notations for the generalized- α and one-step- θ time integration scheme, respectively. The corresponding user-chosen parameter, either $\rho_\infty^{\mathcal{F}}$ or $\theta^{\mathcal{F}}$, is specified as well as the temporal order of accuracy of the conversion between interface ALE displacements and fluid velocities.

Only in those cases, where the conversion between interface ALE displacements and fluid velocities is performed with the backward EULER scheme (3.21), the temporal convergence rate deteriorates to first order. In all cases with second order time integration schemes and second order conversion between interface ALE displacements and fluid velocities, the temporal convergence rate is of second order as expected. Summing up, the theoretically expected convergence rates are fully recovered by the proposed monolithic FSI scheme.

3.4.2 LAGRANGE Multiplier Field and Choice of Master and Slave Side

As introduced in Section 2.3, the LAGRANGE multiplier field is assumed to equal the interface traction field that acts onto the structure side of the interface. To clarify its physical meaning as well as to illustrate its relation to the enforcement of essential boundary conditions and interface coupling conditions, the pseudo one-dimensional example is solved with both fluid-handled and structure-handled interface motion algorithms.

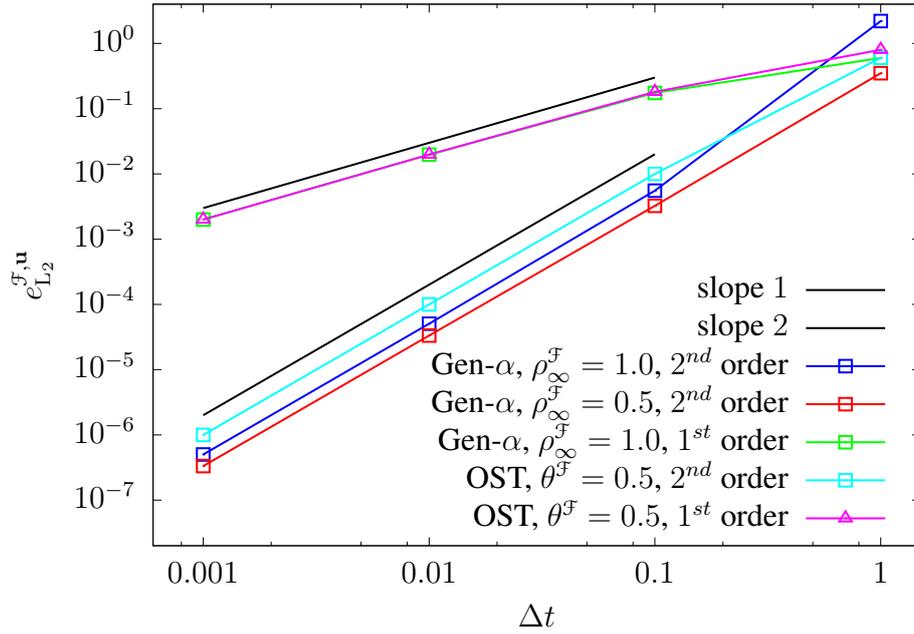
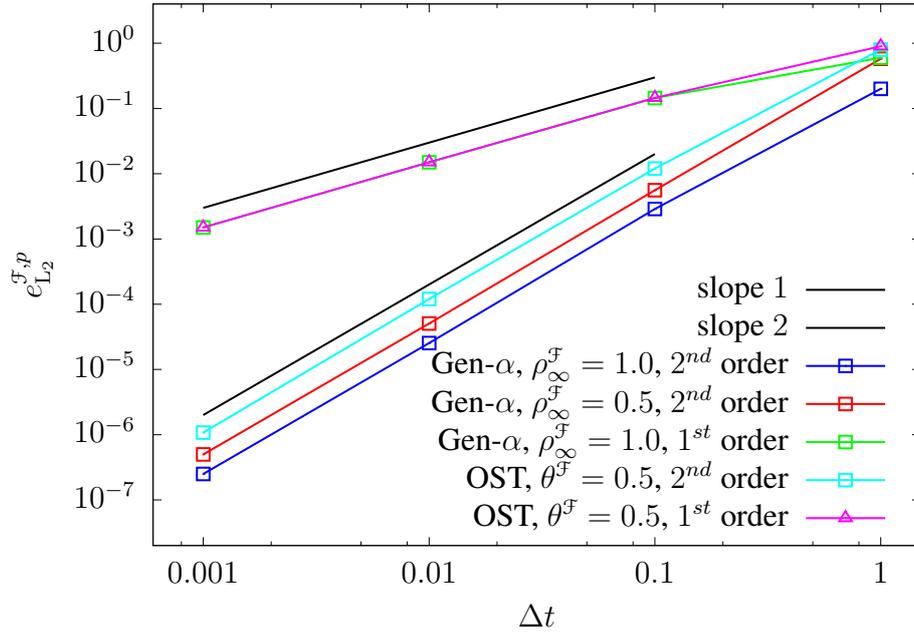

 (a) L_2 -error in velocity field

 (b) L_2 -error in pressure field

Figure 3.4: Temporal convergence study for pseudo one-dimensional FSI example — Configurations with varying fluid time integration schemes and conversions of ALE interface displacements to fluid interface velocities are compared. Temporal convergence rates are measured using L_2 -errors of velocity field and pressure field in the fluid domain $\Omega^{\mathcal{F}}$. The computed convergence rates perfectly match theoretical expectations.

Table 3.1: Material parameters for discussion of the LAGRANGE multiplier field of a pseudo one-dimensional FSI example

Fluid			Solid		
dynamic viscosity	$\mu_{\text{dyn}}^{\mathcal{F}}$	1.0	YOUNG's modulus	$E^{\mathcal{S}}$	10^3
density	$\rho^{\mathcal{F}}$	1.0	POISSON's ratio	$\nu^{\mathcal{S}}$	$\{0, 0.3\}$
			density	$\rho^{\mathcal{S}}$	1.0

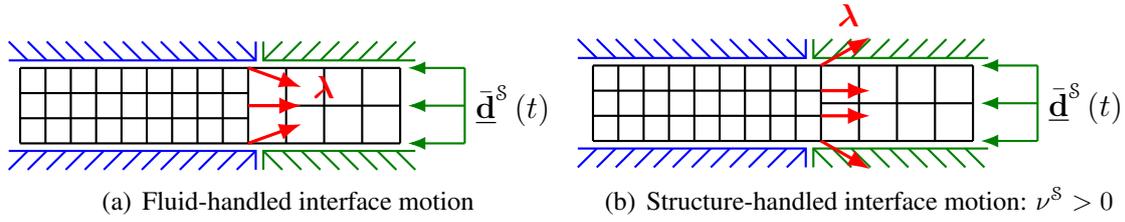


Figure 3.5: DIRICHLET boundary conditions for a pseudo one-dimensional example — This projection in a two-dimensional setting illustrates the set of DIRICHLET boundary conditions for fluid-handled and structure-handled interface motion algorithms, that are imposed on the master side of the interface only [190]. DIRICHLET boundary conditions on fluid and structure domain are indicated in blue and green color, respectively. Arrows indicate the expected LAGRANGE multiplier field at the interface whose lateral components preclude lateral motion of the slave side's interface degrees of freedom.

The assumption of a rigid solid body is dropped since the influence of POISSON's ratio $\nu^{\mathcal{S}}$ on the LAGRANGE multiplier field facilitates an illustrative and fruitful discussion. So, the material parameters listed in Table 3.1 are used, where two values for the solid's POISSON's ratio are tested. Due to the prescribed displacement $\bar{\mathbf{d}}^{\mathcal{S}}(t) = -t^3$ of the solid surface at $x = \ell^{\mathcal{F}} + \ell^{\mathcal{S}}$ in negative x -direction, the solid is compressed. Thus, it tends to expand in lateral direction if $\nu^{\mathcal{S}} > 0$.

As advised by PUSO and LAURSEN [190] and stated in Remarks 3.3.2 and 3.3.3, the DIRICHLET boundary conditions at the fluid-structure interface are directly imposed on the master side only, see Figure 3.5. Their transfer to and weak enforcement on the slave side are handled by the mortar coupling.

The following cases are studied:

- Fluid-handled interface motion with solid with POISSON's ratio $\nu^{\mathcal{S}} = 0$
- Fluid-handled interface motion with solid with POISSON's ratio $\nu^{\mathcal{S}} = 0.3$
- Structure-handled interface motion with solid with POISSON's ratio $\nu^{\mathcal{S}} = 0$
- Structure-handled interface motion with solid with POISSON's ratio $\nu^{\mathcal{S}} = 0.3$

In case of fluid-handled interface motion, indicated in Figure 3.5(a), the structure field is chosen as slave field and, thus, the LAGRANGE multiplier field is discretized on the solid side

of the interface. Since no DIRICHLET boundary conditions are allowed on the fluid interface nodes, the interface traction field $\mathbf{h}_{\Gamma_{\text{FSI}}}^{\mathcal{F}}$ on the fluid side needs to exhibit a lateral component, that points towards the center line of the channel to prevent the fluid to leave the channel in lateral direction at the fluid-structure interface. Taking into account NEWTON's third law [175], i.e. *actio = reactio*, yields that the lateral component of $\boldsymbol{\lambda}$ at the channel walls is expected to point away from the center line of the channel. This is in accordance with the assumption, that the LAGRANGE multiplier field $\boldsymbol{\lambda}$ is interpreted as the interface traction, that acts onto the structure, as well as with the interface equilibrium (2.13). Due to the absence of shear in the velocity field, the LAGRANGE multiplier field's x -component λ_x is supposed to equal the fluid pressure $p_{\Gamma}^{\mathcal{F}}$ at the interface.

In contrast, in case of structure-handled interface motion as illustrated in Figure 3.5(b), the LAGRANGE multiplier field is discretized on the fluid side of the interface. If POISSON's ratio $\nu^{\mathcal{S}} > 0$, the structure tends to expand in lateral direction which then needs to be precluded by the lateral component of the LAGRANGE multiplier field. With $\nu^{\mathcal{S}} = 0$, this tendency doesn't exist and, thus, the interface traction field is expected to be aligned with the x -direction. Again, the fluid pressure at the interface has to match the x -component λ_x of the LAGRANGE multiplier field.

Figure 3.6 shows visualizations of the discrete interface LAGRANGE multiplier field $\boldsymbol{\lambda}$ for all cases. For all cases, the fluid pressure $p_{\Gamma}^{\mathcal{F}}$ at the fluid-structure interface equals the x -component λ_x of the LAGRANGE multiplier field. However, the level of the fluid interface pressure depends on the actual choice of $\nu^{\mathcal{S}}$. The choice $\nu^{\mathcal{S}} = 0$ allows for more compressibility of the solid domain. Hence, the fluid volume that needs to be pushed by the solid block is larger, which results in a larger fluid interface pressure than in case of $\nu^{\mathcal{S}} > 0$. The DIRICHLET boundary conditions at the slave side of the interface are fulfilled as well due to the mortar coupling. The LAGRANGE multiplier field exhibits lateral components in those cases where it is expected. Finally, it can be concluded that the numerical solutions perfectly match the expected results described above.

3.5 Summary

In this chapter, the discretization of the fluid field, the solid field, and the fluid-structure interface have been shown and the assembly of the monolithic system of equations has been performed. Thereby, focus was put on the temporally consistent exchange of interface traction fields between the solid and the fluid field, which could be realized by introducing a temporal interpolation of these traction fields. Static condensation of the LAGRANGE multiplier field as well as the slave side's interface degrees of freedom led to two different algorithmic variants, namely the cases of fluid-handled interface motion and structure-handled interface motion. Using a one-dimensional example, optimal temporal convergence rates could be shown. Furthermore, the interplay of the LAGRANGE multiplier field with the imposition of DIRICHLET boundary conditions at the fluid-structure interface was illustrated and discussed.

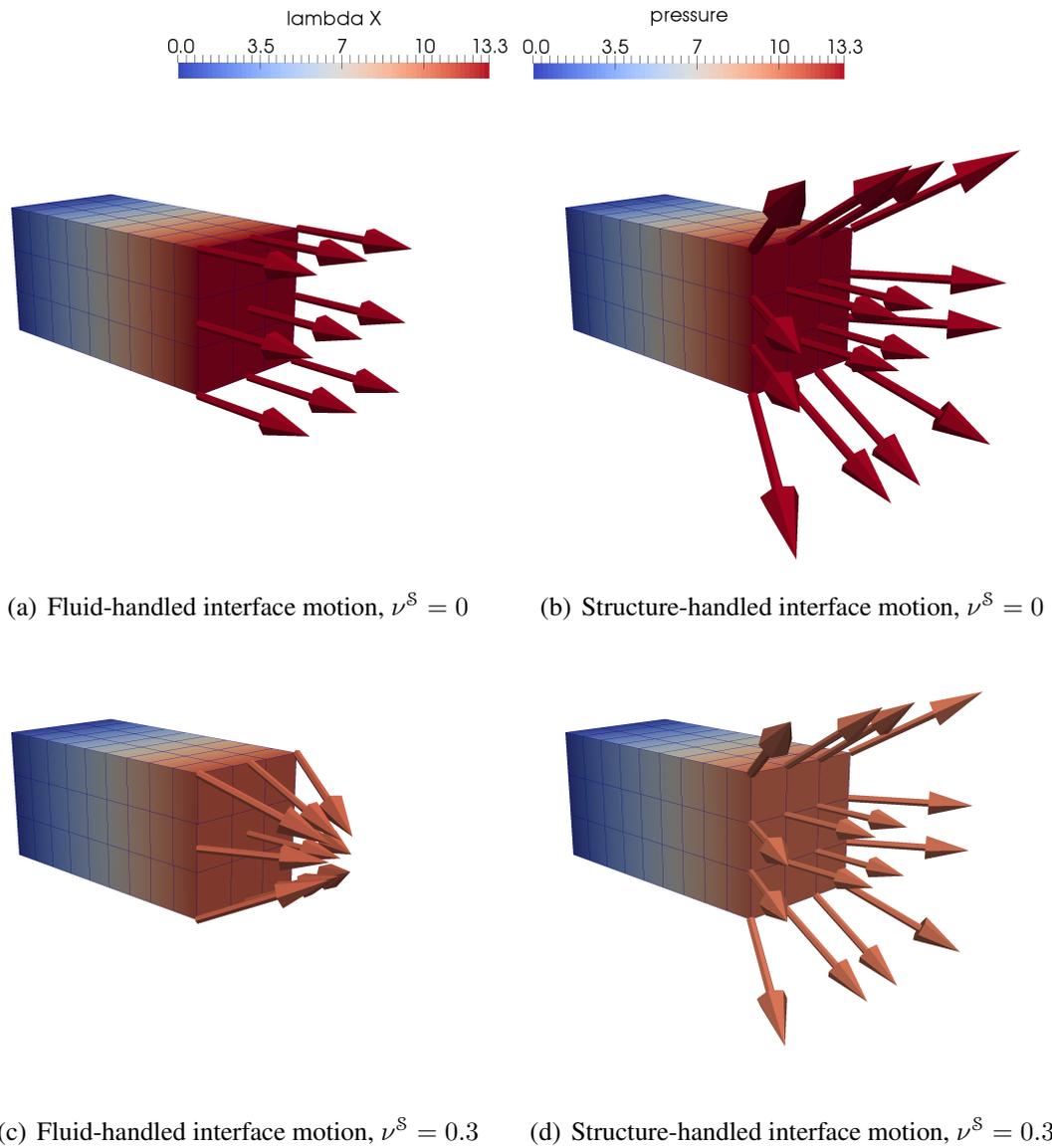


Figure 3.6: Comparison of LAGRANGE multiplier fields for fluid-handled and structure-handled interface motion algorithms — The fluid domain is colored according to its pressure field, while the LAGRANGE multiplier field, that represents the interface traction, is colored according to its x -component. The solid domain is not shown for clarity of presentation. Interface pressure $p_{\Gamma}^{\mathcal{F}}$ and the traction component λ_x match as expected. If the fluid field is chosen as the slave field, the LAGRANGE multiplier field always exhibits lateral components, that prevent the fluid from leaving the channel, cf. Figures 3.6(b) and 3.6(d). If the structure field is chosen as the slave field, a lateral component is only evident for $\nu^s > 0$, cf. Figure 3.6(c). An x -aligned LAGRANGE multiplier field is obtained only for fluid-handled interface motion with $\nu^s = 0$, cf. Figure 3.6(a).

4 An Adaptive Time Stepping Procedure for Monolithic FSI Solvers

All FSI problems in this thesis are transient problems and, thus, involve temporal discretization and time integration. However, the actual choice of the time step size Δt is not an easy one, especially if the transient characteristic of the problem varies over time. A too large choice for the time step size leads to a reduced accuracy or even stability problems. Choosing the time step size very small results in a more accurate solution. However, even with a small value for the time step size Δt , quantification of the level of accuracy would require error estimation or a series of simulations with different time step sizes to allow a comparison of the different solutions. Furthermore, a constantly small time step size increases the number of time steps to be computed and, thus, is very inefficient in many scenarios. Especially when the transient characteristics of the problem change over time, a uniform time step size Δt for every time step is likely too far from an optimal choice. Thus, one aims at providing a temporal discretization with varying time step sizes Δt_n with the subscript n indicating the variability of the time step size. Such a variable time step size Δt_n accounts for the current physical state of the problem such that the deviation of the numerical solution from the exact, but unknown solution is minimized or at least bounded by a user-given tolerance [58]. In the design of such algorithms, the following important requirements have to be kept in mind [58, 130]:

- The deviation of the discrete solution from the exact solution should be below a user-given tolerance for any time $t \in (0, T)$.
- The additional computational cost associated with adapting the time step size should be comparatively small.
- The algorithms should be based on theoretical considerations.
- No (or only very rough) *a priori* knowledge about the exact solution of the problem should be required.
- To be computationally efficient, the time step size should be as large as possible, but as small as necessary.

Usually, deviations of the numerical solution from the exact solution are denoted as *errors*. Following the definition by HUERTA *et al.* [121], either *error estimation*, where a measure of the actual error is approximated in a given norm, or *error indication*, that relies on heuristics, are applied to assess the error. Speaking in these terms, error estimation is applied exclusively in this thesis. Further classification into *a priori* and *a posteriori* error estimation can be made. The former one can be performed before solving the problem, while the latter one requires

knowledge of the solution. All error estimation processes used throughout this thesis are of a *posteriori* type.

Alternative approaches based on control theory have been developed by GUSTAFSSON *et al.* [107], GUSTAFSSON [104–106], and SÖDERLIND [221–223]. Such approaches are said to increase stability and to produce a smoother evolution of time step sizes. A detailed analysis of the analogy of these approaches to the more classical methods based on *a posteriori* error estimation is carried out by DEUFLHARD and BORNEMANN [58], for example. Since algorithms based on *a posteriori* error estimation produced satisfying results for FSI problems, approaches based on control theory are not considered in this thesis.

Recently, approaches for *goal-oriented error estimation* became very popular, where one aims at controlling the error in a user-chosen *quantity of interest*. Since these techniques usually require the solution of an adjoint problem, they become computationally and storage-wise very expensive in transient problems and, thus, are not considered in this thesis. However, some approaches addressing these issues are available in literature. For instance, CYR *et al.* [42] use data compression techniques to reduce the huge storage demands, while CAREY *et al.* [32] apply a block-wise adaptivity approach based on coarse scale adjoint solutions. Promising work based on modal analysis has been done by VERDUGO *et al.* [238, 239] for time-dependent solid mechanics problems. Further approaches can be found in [151, 161].

To the author’s knowledge, the present work poses the first development of an *a posteriori* error estimation based adaptive time stepping scheme in the context of monolithic solvers for FSI with incompressible fluid flow and finite deformation.

This chapter is organized as follows: At first, fundamentals of adaptive time stepping schemes are introduced in Section 4.1. At second, an adaptive time stepping scheme for monolithic solvers for FSI problems is proposed in Section 4.2 for the first time. In addition, several important practical aspects are highlighted in this section as well. Three numerical examples are presented in Section 4.3, that are used to demonstrate and discuss features and properties of the proposed adaptive time stepping scheme. Finally, an intermediate summary is given in Section 4.4.

4.1 Fundamentals

Here, only those fundamental relations are given, that are necessary to lay the foundation for the development of adaptive time stepping procedures based on *a posteriori* error estimation. For further insight into the numerical treatment of *ordinary differential equations (ODEs)* and time integration, see the textbooks by ASCHER and PETZOLD [3], BUTCHER [31], DEUFLHARD and BORNEMANN [58], HAIRER *et al.* [110], or HAIRER and WANNER [111] to name few. Moreover, many books on numerical methods include sections about ODEs and time integration, for example FAIRES and BURDEN [67], QUARTERONI *et al.* [193], QUARTERONI *et al.* [194], and SCHWARZ and KÖCKLER [216].

Section 4.1.1 reflects some basics on time integration, before the important terms of *error* and *order of convergence*, that are associated with every time integration scheme, are replicated in Section 4.1.2. A well-established strategy to adapt the time step size is given in 4.1.3, which relies on practical error estimation detailed in Section 4.1.4.

4.1.1 Preliminaries

The *ordinary differential equation (ODE) of first order*

$$\dot{\underline{\mathbf{x}}}(t) = \underline{\mathbf{f}}(\underline{\mathbf{x}}(t), t) \quad (4.1)$$

is given. A *temporal discretization* is performed, introducing $N + 1$ discrete time instances $\{t_0, t_1, \dots, t_N\}$ where the solution is sought. This temporal grid is not necessarily equally spaced, i.e. the time step size

$$\Delta t_n = t_{n+1} - t_n$$

may differ from time step to time step where n is used to index the time steps. The exact solution at time t_{n+1} is denoted by $\underline{\mathbf{x}}(t_{n+1})$, while its time-discrete approximation is indicated by $\underline{\mathbf{x}}_{n+1}$.

Following the presentation by DEUFLHARD and BORNEMANN [58], the computation of the unknown solution $\underline{\mathbf{x}}(t_{n+1})$ based on known values $\underline{\mathbf{x}}(t_m)$, $m \leq n$, can be summarized in the discrete recurrence relation

$$\underline{\mathbf{x}}(t_{n+1}) = \underline{\Phi}_{n+1,n} \underline{\mathbf{x}}(t_n) \quad (4.2)$$

where $\underline{\Phi}_{n+1,n}$ denotes the *exact evolution* from time t_n to time t_{n+1} . Recursive application relates the current solution to the initial value $\underline{\mathbf{x}}(t_0)$ by

$$\underline{\mathbf{x}}(t_{n+1}) = \underline{\Phi}_{n+1,n} \underline{\Phi}_{n,n-1} \dots \underline{\Phi}_{1,0} \underline{\mathbf{x}}(t_0) = \underline{\Phi}_{n+1,0} \underline{\mathbf{x}}(t_0).$$

With $\underline{\Psi}_{n+1,n}$ denoting the *discrete or approximate evolution* from time t_n to time t_{n+1} , a time integration step is given as

$$\underline{\mathbf{x}}_{n+1} = \underline{\Psi}_{n+1,n} \underline{\mathbf{x}}_n \quad (4.3)$$

where the actual expression for $\underline{\Psi}_{n+1,n}$ depends on the time integration scheme at hand.

4.1.2 Errors and the Order of Convergence

The deviation of the time-discrete solution $\underline{\mathbf{x}}_{n+1}$ from the exact solution $\underline{\mathbf{x}}(t_{n+1})$ is measured by the *global error* [58, 110]

$$\underline{\mathbf{g}}_{n+1} = \underline{\mathbf{x}}(t_{n+1}) - \underline{\mathbf{x}}_{n+1}. \quad (4.4)$$

Inserting the exact evolution (4.2) and the discrete evolution (4.3) into (4.4) yields

$$\begin{aligned} \underline{\mathbf{g}}_{n+1} &= \underline{\Phi}_{n+1,n} \underline{\mathbf{x}}(t_n) - \underline{\Psi}_{n+1,n} \underline{\mathbf{x}}_n \\ &= \underbrace{\underline{\Phi}_{n+1,n} \underline{\mathbf{x}}(t_n) - \underline{\Psi}_{n+1,n} \underline{\mathbf{x}}(t_n)}_{\underline{\mathbf{l}}_{n+1}} + \underbrace{\underline{\Psi}_{n+1,n} \underline{\mathbf{x}}(t_n) - \underline{\Psi}_{n+1,n} \underline{\mathbf{x}}_n}_{\underline{\mathbf{g}}_{n+1}} \\ &= \underline{\mathbf{l}}_{n+1} + \underline{\mathbf{g}}_{n+1}. \end{aligned} \quad (4.5)$$

This decomposes the global error into the *local discretization error*

$$\underline{\mathbf{l}}_{n+1} = \underline{\Phi}_{n+1,n} \underline{\mathbf{x}}(t_n) - \underline{\Psi}_{n+1,n} \underline{\mathbf{x}}(t_n) \quad (4.6a)$$

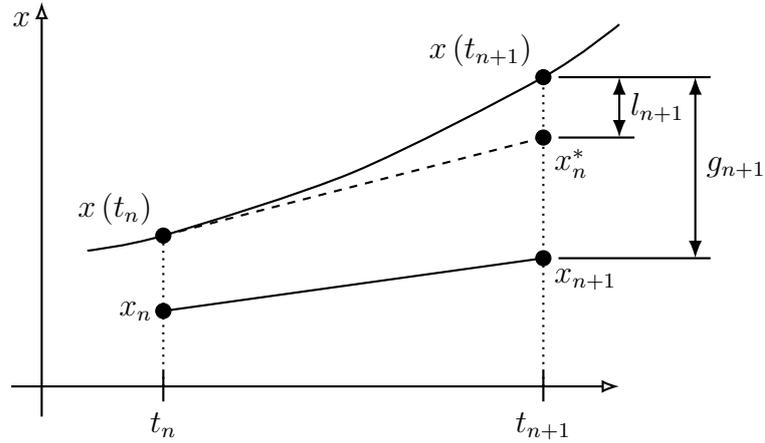


Figure 4.1: Illustration of local and global error for the forward EULER method adopted from QUARTERONI *et al.* [193] — The global error g_{n+1} measures the deviation of the discrete solution x_{n+1} from the exact solution $x(t_{n+1})$. The local error l_{n+1} quantifies the deviation from the exact solution $x(t_{n+1})$ from the approximate solution x_n^* after one integration step when starting from the exact solution $x(t_n)$.

$$\approx \underline{\mathbf{x}}(t_{n+1}) - \underline{\mathbf{x}}_{n+1}, \quad (4.6b)$$

that is generated when integrating from t_n to t_{n+1} while assuming $\underline{\mathbf{x}}_n = \underline{\mathbf{x}}(t_n)$, and the *propagation error*

$$\bar{\underline{\mathbf{g}}}_{n+1} = \underline{\Psi}_{n+1,n} \underline{\mathbf{x}}(t_n) - \underline{\Psi}_{n+1,n} \underline{\mathbf{x}}_n, \quad (4.7)$$

that represents accumulated errors from previous time steps [58]. An illustration of local and global error in case of the forward EULER method is shown in Figure 4.1.

Inserting the TAYLOR series approximation

$$\underline{\mathbf{x}}(t_{n+1}) = \sum_{m=0}^{\infty} \frac{1}{m!} \left. \frac{\partial^m \underline{\mathbf{x}}(t)}{\partial t^m} \right|_{t=t_n} \Delta t_n^m \quad (4.8)$$

of $\underline{\mathbf{x}}(t_{n+1})$ around t_n into (4.6b) and using the following shorthand notation of a time derivative

$$\underline{\mathbf{x}}^{(m)}(t_n) = \left. \frac{\partial^m \underline{\mathbf{x}}(t)}{\partial t^m} \right|_{t=t_n} \quad \text{with } m = 0, 1, 2, \dots \quad (4.9)$$

yields

$$\underline{\mathbf{l}}_{n+1} = C \underline{\mathbf{x}}^{(p+1)}(t_n) \Delta t_n^{p+1} + \mathcal{O}(\Delta t_n^{p+2}) \quad (4.10)$$

for the local error [110, 219], where p denotes the *order of consistency* of the time integration scheme at hand [216] that equals the *order of convergence* of the time integration scheme [58, 216]. The constant C is often referred to as *leading error coefficient*. Its value depends on the actual time integration scheme at hand.

4.1.3 Adapting the Time Step Size

Primarily, adaptive time stepping algorithms aim at controlling the global error $\underline{\mathbf{g}}_{n+1}$, such that the relation

$$\left\| \underline{\mathbf{g}}_{n+1} \right\|_{L_2} \leq \varepsilon_g^{\Delta t} \quad (4.11)$$

with a user-given tolerance $\varepsilon_g^{\Delta t}$ for the global error holds in every time step where the L_2 -norm has already been defined in (3.54). A brief discussion of the norm calculation in practical computer codes is given in Section 4.2.3.

Unfortunately, two major issues come along with the goal of controlling the global error: First, the global error is not accessible directly since the exact solution is unknown. Second, as stated in (4.5), the global error $\underline{\mathbf{g}}_{n+1}$ contains the propagation error $\bar{\underline{\mathbf{g}}}_{n+1}$ which cannot be controlled without re-running the entire simulation from the beginning [58]. Both issues make a control of the global error infeasible for practical applications. Hence, the alternative goal of keeping the local error below a user-given tolerance $\varepsilon_l^{\Delta t}$ is pursued [58], reading

$$\left\| \underline{\mathbf{l}}_{n+1} \right\|_{L_2} \leq \varepsilon_l^{\Delta t}. \quad (4.12)$$

SHAMPINE [219] justifies this choice by the fact, that the error does not accumulate if the error per step is kept constant. Furthermore, “the behavior of the bound on the global error with respect to the local tolerance $[\varepsilon_l^{\Delta t}]$ is particularly regular [...]” [219, p. 198]. If the global error is approximated as sum of the local errors, then according to BUTCHER [31] optimality in a sense is obtained when the local error is maintained to be constant in every time step, i.e.

$$\left\| \underline{\mathbf{l}}_{n+1} \right\|_{L_2} \doteq \varepsilon_l^{\Delta t} = \text{const.} \quad \forall n = 0, 1, \dots, N - 1. \quad (4.13)$$

Based on these observations, a *practical* adaptive time stepping scheme has to satisfy (4.12) in every time step. Therefore, it computes an *optimized scaling factor* κ^* used to increase or decrease the time step size such that the local error, that has been obtained with an optimally scaled time step size, approximately equals the local error tolerance, reading

$$\varepsilon_l^{\Delta t} \approx \left\| \underline{\mathbf{l}}_{n+1} (\kappa^* \Delta t_n) \right\|_{L_2}. \quad (4.14)$$

This idea is now transformed into an algorithm. Inserting the TAYLOR expansion (4.10) of the local error into (4.14), i.e.

$$\varepsilon_l^{\Delta t} \approx C \left\| \underline{\mathbf{x}}^{(p+1)}(t_n) \right\|_{L_2} (\kappa^* \Delta t_n)^{p+1} + \mathcal{O}(\Delta t_n^{p+2}), \quad (4.15)$$

and deducing the norm of (4.10) for the current step $t_n \rightarrow t_{n+1}$ gives

$$\left\| \underline{\mathbf{l}}_{n+1} \right\|_{L_2} = C \left\| \underline{\mathbf{x}}^{(p+1)}(t_n) \right\|_{L_2} \Delta t_n^{p+1} + \mathcal{O}(\Delta t_n^{p+2}). \quad (4.16)$$

After neglecting higher order terms $\mathcal{O}(\Delta t_n^{p+2})$ and combining (4.15) and (4.16), the intermediate expression

$$\frac{\varepsilon_l^{\Delta t}}{\left\| \underline{\mathbf{l}}_{n+1} \right\|_{L_2}} \approx \frac{C \left\| \underline{\mathbf{x}}^{(p+1)}(t_n) \right\|_{L_2} (\kappa^* \Delta t_n)^{p+1}}{C \left\| \underline{\mathbf{x}}^{(p+1)}(t_n) \right\|_{L_2} \Delta t_n^{p+1}}$$

can be solved for the optimized scaling factor

$$\kappa^* \approx \sqrt[p+1]{\frac{\varepsilon_l^{\Delta t}}{\|\mathbf{1}_{n+1}\|_{L_2}}}, \quad (4.17)$$

that is used to increase or decrease the time step size. Equation (4.17) establishes a relation between the local error, that represents the current state of accuracy of the simulation, and the desired accuracy $\varepsilon_l^{\Delta t}$. If the accuracy demand (4.12) holds, the scaling factor $\kappa^* \geq 1$, which will result in an increased time step size $\Delta t_{n+1} \geq \Delta t_n$, that will be used to proceed to the next time step. However, in case of a violation of (4.12), the time step needs to be repeated to satisfy (4.12), which is then more likely, since in this case a scaling factor $\kappa^* < 1$ will decrease the time step size [58, 110].

Since the exact error $\|\mathbf{1}_{n+1}\|_{L_2}$ is not accessible, an estimate $\|\mathbf{1}_{n+1}^\diamond\|_{L_2} \approx \|\mathbf{1}_{n+1}\|_{L_2}$ is computed and used instead [58], where the superscript $(\bullet)^\diamond$ indicates the estimation process. In practice, the optimal scaling factor κ^* is computed as

$$\kappa^* \approx \sqrt[p+1]{\frac{\varepsilon_l^{\Delta t}}{\|\mathbf{1}_{n+1}^\diamond\|_{L_2}}}, \quad (4.18)$$

where the local error is estimated by any of the methods detailed in Section 4.1.4. Based on the current time step size Δt_n , the set

$$[\kappa] = \{\kappa_{\min}, \kappa_{\max}, \kappa_s\}$$

of algorithmic parameters as well as some user-given bounds on the time step size, an optimal time step size can be computed as

$$\Delta t^* = \min \{ \Delta t_{\max}, \max \{ \min \{ \kappa_{\max}, \max \{ \kappa_{\min}, \kappa_s \kappa^* \} \} \Delta t_n, \Delta t_{\min} \} \}. \quad (4.19)$$

In order to avoid too many repetitions of time steps, a *safety factor* $\kappa_s < 1$ is used, that is supposed to keep the local error below and away from the tolerance barrier [110]. The effective scaling factor $\kappa_s \kappa^*$ is limited by user-given factors κ_{\min} and κ_{\max} that denote the minimal and maximal ratio of time step size decrease or increase, respectively, such that $\kappa_{\min} \leq \kappa_s \kappa^* \leq \kappa_{\max}$. Finally, the time step size is limited by the lower and upper bounds Δt_{\min} and Δt_{\max} . Section 4.2.3 will provide some hints for the choice of these algorithmic parameters.

The algorithm is summarized in Figure 4.2. In each time step, one starts with performing the integration from t_n , where all values are known, to t_{n+1} using the fully implicit scheme as the marching time integration scheme. Afterwards, the error associated with the result of this marching step is assessed by means of error estimation, yielding $\|\mathbf{1}_{n+1}^\diamond\|_{L_2}$. The estimated error is then used to compute a new time step size Δt^* using (4.19) in combination with (4.18). If the estimated error $\|\mathbf{1}_{n+1}^\diamond\|_{L_2} > \varepsilon_l^{\Delta t}$, the accuracy demand (4.12) has been violated and, thus, the current time step needs to be repeated with a *reduced* time step size $\Delta t_n = \Delta t^*$ where (4.18) guarantees $\kappa^* < 1$ in (4.19). If $\|\mathbf{1}_{n+1}^\diamond\|_{L_2} \leq \varepsilon_l^{\Delta t}$, one can directly proceed to the next time step with a new, probably *increased*, time step size $\Delta t_{n+1} = \Delta t^*$.

In case of drastic changes of the transient behavior, especially if these changes occur suddenly like in the case of buckling events, it might be necessary to repeat a single time step $t_n \rightarrow$

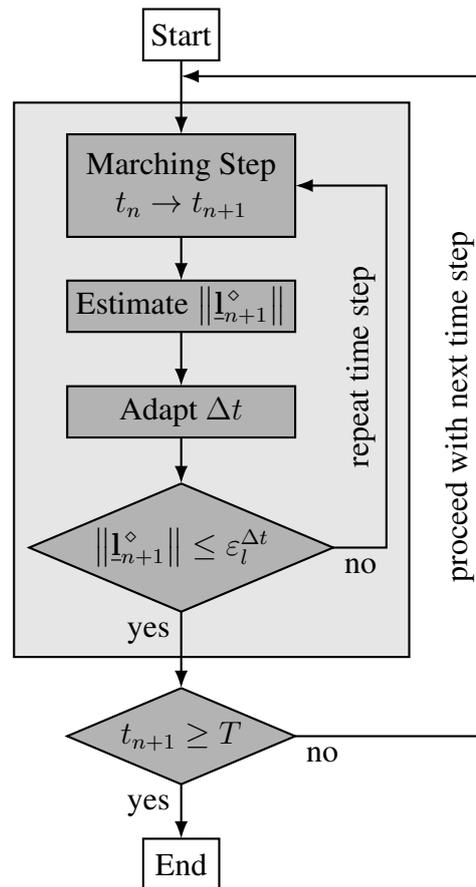


Figure 4.2: Adaptive time stepping algorithm based on *a posteriori* error estimation — In each time step, the *marching step* is done using the fully implicit time integrator. Based on an *a posteriori* error estimate, one can adapt the time step size. After checking the accuracy demand (4.12), one either needs to repeat the current time step with a reduced time step size or can proceed with the next time step. This procedure is done in every time step until the simulation time reaches the final time T .

t_{n+1} several times while reducing the time step size in each repetition. For practical reasons, the number of repetitions of a single time step may be limited. See Section 4.2.3 for further discussion.

To increase robustness of application codes, time step size reductions and time step repetitions may be necessary if the nonlinear solver fails to converge. Further details on such problems as well as a possible strategy to deal with them will be outlined in Section 4.2.3.

4.1.4 Estimation of the Local Discretization Error

According to DEUFLHARD and HOHMANN [59], an *error estimator* ϵ^\diamond for an unknown error ϵ is characterized by the fact that $\kappa_1\epsilon \leq \epsilon^\diamond \leq \kappa_2\epsilon$ with constants κ_1, κ_2 satisfying $\kappa_1 \leq 1 \leq \kappa_2$. However, the actual design of the error estimator poses one of the most difficult tasks in the design of an adaptive algorithm. It is common practice to compare two approximations to each other. These approximations are usually of different orders of accuracy. Some guidelines for the design of local error estimators are given by ROMERO and LACOMA [203, 204].

Several methods to estimate the local discretization error are available in literature. Some of them are described briefly in the sequel. Although approaches like RICHARDSON *extrapolation* and *embedded methods* are very popular, error estimation in this thesis is mostly performed by comparison of two different schemes of order p and \hat{p} . For the solid field, error estimation based on TAYLOR series expansion is available. For detailed derivations and discussion of other error estimators see the textbooks by BUTCHER [31], DEUFLHARD and BORNEMANN [58], or HAIRER *et al.* [110] to name only a few.

RICHARDSON Extrapolation

Based on knowledge about the relation of the error as a function of the time step size Δt_n , the classic approach by RICHARDSON, originally proposed in [199] and fully detailed in [200], allows for an approximation of the local discretization error at time $t_{n+1} = t_n + \Delta t_n$ by comparison of the solution $\underline{\mathbf{x}}_{\Delta t_n}$ after one step with Δt_n to the solution $\underline{\mathbf{x}}_{\Delta t_n/2}$ after two steps with $\Delta t_n/2$. Figure 4.3 illustrates both integration steps. An estimate of the local discretization error of the solution associated with the smaller step size $\Delta t_n/2$ is then given as

$$\mathbf{l}_{n+1}^\diamond \approx \frac{1}{2^p - 1} (\underline{\mathbf{x}}_{\Delta t_n} - \underline{\mathbf{x}}_{\Delta t_n/2})$$

with p being the order of the time integration scheme at hand. Then, RICHARDSON *extrapolation* can be used to compute an improved approximation to the solution at time $t_n + \Delta t_n$ which is of order $p + 1$. However, this possibility is not exploited in the present work.

Although RICHARDSON extrapolation provides a very reliable estimate for the error, it lacks efficiency if implicit time integration schemes are used. If the solution $\underline{\mathbf{x}}_{\Delta t_n}$ with the lower accuracy is propagated, the additional cost for error estimation is twice as large as for the marching scheme since the error estimation requires the computation of two additional time steps. However, when propagating the more accurate solution $\underline{\mathbf{x}}_{\Delta t_n/2}$, the additional cost is only 50%, which is still far from negligible [31]. Summing up, RICHARDSON extrapolation is an interesting concept. But since only fully implicit time integration schemes are applied in this thesis,

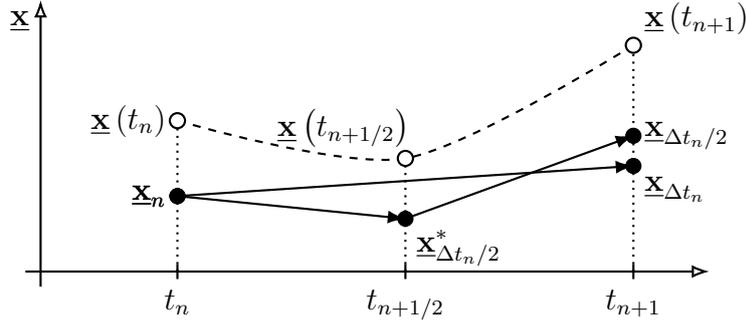


Figure 4.3: Error estimation by RICHARDSON extrapolation — In RICHARDSON extrapolation, two solutions are compared to each other, namely $\underline{x}_{\Delta t_n}$ and $\underline{x}_{\Delta t_n/2}$. By marching two steps with step size $\Delta t_n/2$ and passing the intermediate solution $\underline{x}_{\Delta t_n/2}^*$ one arrives at the more accurate solution $\underline{x}_{\Delta t_n/2}$. To arrive at an error estimate, it is then compared to $\underline{x}_{\Delta t_n}$ which is obtained by a single step with step size Δt_n .

RICHARDSON extrapolation would be unaffordable in terms of computational cost and, thus, is not considered any further.

Note that the basic concepts of RICHARDSON [199, 200] are also applicable for estimating the spatial approximation error in case of adaptive grid refinement.

Methods with Built-in Error Estimators

Embedded methods are very popular if the employed time integration scheme is a *multi-stage scheme*, i.e. it uses several intermediate function evaluations, called *stages*, within a single time step, e.g. RUNGE-KUTTA methods. Embedded methods use one set of stage vectors, but two sets of coefficients to compute two different approximations $\hat{\underline{x}}_{n+1}$ and $\tilde{\underline{x}}_{n+1}$ of the solution at time t_{n+1} . The coefficients are selected such that the two solutions are usually of different orders p and \hat{p} , e.g. $\hat{p} = p + 1$. An estimate of the local discretization error of the solution with lower order is then computed as

$$\mathbf{I}_{n+1}^\diamond \approx \hat{\underline{x}}_{n+1} - \tilde{\underline{x}}_{n+1}.$$

The first method of this type was proposed by MERSON [167]. According to [110], early methods of this type were proposed also by CESCHINO [34] and ZONNEVELD [265] while further developments have been performed by SARAFYAN [212] and ENGLAND [66]. Very popular embedded methods in the framework of RUNGE-KUTTA schemes are those by FEHLBERG [73–76]. Following [216], CESCHINO and KUNTZMANN [35] developed estimators for local discretization errors of arbitrary *4th* order RUNGE-KUTTA methods. Other well-known methods are those by VERNER [241], who uses two different quadrature formulas for both methods, and by DORMAND and PRINCE [63].

In this thesis, only single-stage time integration schemes are used. For further details on multi-stage schemes and error estimation for such schemes, the reader is referred to [31, 58, 110].

Comparison of Two Schemes of Order p and \hat{p}

In this thesis, error estimation is mostly performed by comparison of two schemes of order p and \hat{p} . Thereby, one scheme is the *marching implicit* scheme and the other is an *auxiliary explicit* scheme. Compared to RICHARDSON extrapolation, such a method reduces computational effort since only a single implicit scheme is employed. Due to the explicit character of the auxiliary scheme, its computational cost is almost negligible since it requires only a couple of vector updates.

To derive an error estimator, one starts with the TAYLOR series expansion (4.10) for both schemes, reading

$$\underline{\mathbf{l}}_{n+1} = C_{\underline{\mathbf{x}}^{(p+1)}}(t_n) \Delta t_n^{p+1} + \mathcal{O}(\Delta t_n^{p+2}), \quad (4.20a)$$

$$\hat{\underline{\mathbf{l}}}_{n+1} = \hat{C}_{\underline{\mathbf{x}}^{(\hat{p}+1)}}(t_n) \Delta t_n^{\hat{p}+1} + \mathcal{O}(\Delta t_n^{\hat{p}+2}), \quad (4.20b)$$

where quantities of the auxiliary scheme are identified by the notation $(\hat{\bullet})$. Using definition (4.6b) of the local error for both schemes yields

$$\underline{\mathbf{l}}_{n+1} \approx \underline{\mathbf{x}}(t_{n+1}) - \underline{\mathbf{x}}_{n+1}, \quad (4.21a)$$

$$\hat{\underline{\mathbf{l}}}_{n+1} \approx \underline{\mathbf{x}}(t_{n+1}) - \hat{\underline{\mathbf{x}}}_{n+1}. \quad (4.21b)$$

Subtracting them from each other results in

$$\underline{\mathbf{l}}_{n+1} - \hat{\underline{\mathbf{l}}}_{n+1} \approx \hat{\underline{\mathbf{x}}}_{n+1} - \underline{\mathbf{x}}_{n+1}. \quad (4.22)$$

One can now study two cases, namely $p < \hat{p}$ and $p = \hat{p}$. The case $p > \hat{p}$ is closely related to the first one and, thus, dealt with at the same time.

At first, the case $p < \hat{p}$ is examined. Inserting (4.20b) into (4.22) yields

$$\underline{\mathbf{l}}_{n+1} \approx \hat{\underline{\mathbf{x}}}_{n+1} - \underline{\mathbf{x}}_{n+1} + \hat{C}_{\underline{\mathbf{x}}^{(\hat{p}+1)}}(t_n) \Delta t_n^{\hat{p}+1} + \mathcal{O}(\Delta t_n^{\hat{p}+2}), \quad (4.23)$$

where $\hat{C}_{\underline{\mathbf{x}}^{(\hat{p}+1)}}(t_n) \Delta t_n^{\hat{p}+1} + \mathcal{O}(\Delta t_n^{\hat{p}+2})$ is at least $\mathcal{O}(\Delta t_n^{p+2})$ since $p < \hat{p}$. Exploiting the order of $\underline{\mathbf{l}}_{n+1}$ as given in (4.20a) and, thus, neglecting higher order terms $\mathcal{O}(\Delta t_n^{p+2})$, expression (4.23) reduces to

$$\underline{\mathbf{l}}_{n+1} \approx \underline{\mathbf{l}}_{n+1}^{\diamond} = \hat{\underline{\mathbf{x}}}_{n+1} - \underline{\mathbf{x}}_{n+1}, \quad (4.24)$$

which asymptotically is an estimate of the local discretization error of the lower order scheme of order p [110]. Such an error estimator is sometimes referred to as *upward estimator* [25].

If $p > \hat{p}$, the concept of error estimation is abandoned and $\underline{\mathbf{l}}_{n+1}^{\diamond}$ is merely used for step size selection. This case is often referred to as *local extrapolation* [110] or *downward estimation* [25]. DEUFLHARD and BORNEMANN [58] justify this approach using an analogy to control theory. Thereby, the exponent of the root in (4.18) is required to satisfy certain conditions, however their details are not provided here for the sake of brevity. Note that these conditions are fulfilled for all methods used throughout this thesis.

Next, the second case with $p = \hat{p}$ is considered which is discussed in the thesis by BORNEMANN [25] in full detail. Manipulating (4.20) by exploiting the fact $p = \hat{p}$ and neglecting higher order terms $\mathcal{O}(\Delta t_n^{p+1})$ yields

$$\frac{1}{C} \underline{\mathbf{l}}_{n+1} = \underline{\mathbf{x}}^{(p+1)}(t_n) \Delta t_n^{p+1}, \quad (4.25a)$$

$$\frac{1}{\hat{C}} \hat{\mathbf{l}}_{n+1} = \underline{\mathbf{x}}^{(p+1)}(t_n) \Delta t_n^{p+1}. \quad (4.25b)$$

Both expressions in (4.25) are equal and, thus,

$$\frac{1}{C} \mathbf{l}_{n+1} = \frac{1}{\hat{C}} \hat{\mathbf{l}}_{n+1}$$

which can be recast to

$$\hat{\mathbf{l}}_{n+1} = \frac{C}{\hat{C}} \mathbf{l}_{n+1}. \quad (4.26)$$

Inserting (4.26) into (4.22) finally gives the following estimate of the local discretization error if $p = \hat{p}$:

$$\mathbf{l}_{n+1}^\diamond \approx \frac{C}{C - \hat{C}} (\underline{\mathbf{x}}_{n+1} - \hat{\underline{\mathbf{x}}}_{n+1}). \quad (4.27)$$

Remark 4.1.1 *The estimate (4.27) for marching and auxiliary scheme being of the same order $p = \hat{p}$ differs from the estimate (4.24) for the case $p < \hat{p}$ only by a constant factor $C/(C - \hat{C})$, that depends on the involved time integration schemes. The difference between these two error estimates is just the scaling with a constant factor. In practice, this factor can be dropped which reduces (4.27) to (4.24). By doing so, the concept of error estimation is abandoned and $\mathbf{l}_{n+1}^\diamond$ is just used for the purpose of step size selection. A similar argument has been used to justify the application of (4.24) to the case where $p > \hat{p}$.*

Error Estimation based on TAYLOR Series Expansions

ZIENKIEWICZ and XIE [264] developed an error estimator for solid mechanics problems for NEWMARK-type time integration schemes [173, 174] like the generalized- α method of CHUNG and HULBERT [39]. It is based on the comparison of a TAYLOR series expansion around the known solution at time t_n and the NEWMARK update formulas. The estimate is given as

$$\mathbf{l}_{n+1}^\diamond \approx \left(\beta - \frac{1}{6} \right) \Delta t_n^2 (\ddot{\underline{\mathbf{x}}}_{n+1}^{\text{NM}} - \ddot{\underline{\mathbf{x}}}_n) \quad (4.28)$$

with the NEWMARK parameter β , the acceleration $\ddot{\underline{\mathbf{x}}}_{n+1}^{\text{NM}}$ at time t_{n+1} based on the NEWMARK update, and the known acceleration $\ddot{\underline{\mathbf{x}}}_n$ at time t_n . Obviously, estimate (4.28) is only valid for $\beta \neq 1/6$. Although an alternative estimate can be derived for this special case, it is of no interest in the present work, since this parameter choice spoils unconditional stability of the time integrator [264]. For further details see the original publication by ZIENKIEWICZ and XIE [264].

4.2 Application to Monolithic Fluid-Structure Interaction Solvers

The proposed adaptive time stepping procedure for monolithic fluid-structure interaction solvers is based on *a posteriori* error estimation of the temporal discretization errors in both the fluid

and the structure field. Since dynamics and accuracy demands may differ between the fluid and the structure field, the temporal discretization errors are estimated separately for both fields. To account for the central role of the fluid-structure interface, additional attention is paid to its temporal discretization error. Although error estimation is performed field-wise and, thus, different suggestions for the new time step size can be calculated, a uniform time step size for all fields needs to be found, since the proposed monolithic solution scheme does not allow for subcycling as it is possible in case of partitioned approaches [17, 182]. Due to the field-wise error estimation, errors and tolerances for both fields need to be distinguished, which is done by the superscripts $(\bullet)^{\mathcal{F}}$ and $(\bullet)^{\mathcal{S}}$ for fluid and solid quantities, respectively. However, if the distinction is clear from the context or a statement applies equally to both fields, the superscript will be omitted for the sake of an uncluttered notation.

A typical outline of an adaptive time stepping algorithm has already been shown in Figure 4.2 for single-field applications. In case of the FSI problem, just two minor modifications are necessary: First, the estimation of the local error is replaced by estimations of local errors in both the fluid and the structure field as well as at the fluid-structure interface. Second, during adaption of the time step size, a single value for the time step size needs to be selected, that is then used in all fields. Section 4.2.1 will elaborate the error estimation procedure in case of a monolithic FSI solver, while the selection of the time step size will be detailed in Section 4.2.2.

A preliminary implementation of such an adaptive time stepping scheme for monolithic FSI solvers has been done by WILHELM [251].

4.2.1 Estimation of the Local Discretization Error

Unfortunately, a single number that quantifies the temporal discretization error of the coupled problem is not accessible. Hence, the temporal discretization error is estimated separately in both fields, while additional focus is put on the fluid-structure interface. In this thesis, the temporal discretization error in the structure field is estimated either by comparison to another, explicit scheme like *explicit EULER* [110], *ADAMS-BASHFORTH-2* [110], or the *central difference method* [67] or by the popular method of ZIENKIEWICZ and XIE [264]. In the fluid field, the comparison to either explicit EULER or ADAMS-BASHFORTH-2 is available. It is stressed, that the proposed framework is not limited to these approaches, but can easily be used in combination with other, possibly more sophisticated error estimators. Further error estimation approaches for solid and fluid dynamics are given in [36, 38, 126, 154, 155, 180, 204, 214, 233, 248, 260] and [61, 101, 102, 130, 131, 133, 211, 237], respectively, to name a few without claiming completeness of that list.

When accounting for the central role of the interface, the algorithmic decision of master and slave side comes into play. On the master side, whose interface degrees of freedom are used to express the interface motion, an interface error is deduced by extracting the interface degrees of freedom from the master's side vector of local error. In addition, the error associated with the interior degrees of freedom is considered in both master and slave field. The notation e_{β}^{α} is introduced to denote the norm of the estimated local error, where the superscript $\alpha \in \{\mathcal{F}, \mathcal{S}\}$ indicates fluid or structure field and the subscript $\beta \in \{\Gamma \cup \text{I}, \Gamma, \text{I}\}$ provides information on the involved degrees of freedom. The time step index has been dropped to ease notation. Table 4.1 provides an overview of the different sets of estimated error norms e_{β}^{α} in case of fluid-handled and structure-handled interface motion. Estimating the error of the inner or interface subset of

Table 4.1: Error norms e_β^α , $\alpha \in \{\mathcal{F}, \mathcal{S}\}$ and $\beta \in \{\Gamma \cup \text{I}, \Gamma, \text{I}\}$, for fluid-handled and structure-handled interface motion — On the master side, a separate interface error e_Γ^{ma} is estimated, while on the slave side only interior degrees of freedom contribute to the error norm e_Γ^{sl} . The set of estimated error norms for both choices of master and slave side are indicated by the symbol \times .

Interface motion handled by	master field	slave field	$e_{\Gamma \cup \text{I}}^\mathcal{F}$	$e_\Gamma^\mathcal{F}$	$e_\text{I}^\mathcal{F}$	$e_{\Gamma \cup \text{I}}^\mathcal{S}$	$e_\Gamma^\mathcal{S}$	$e_\text{I}^\mathcal{S}$
Fluid field	$\Omega^\mathcal{F}$	$\Omega^\mathcal{S}$	\times	\times	\times			\times
Structure field	$\Omega^\mathcal{S}$	$\Omega^\mathcal{F}$			\times	\times	\times	\times

degrees of freedom does not lead to notable additional computational costs. One just takes the vector of estimated temporal discretization error with all degrees of freedom and extracts only the inner or the interface portion. The computational cost of this extraction is negligible.

As given in Table 4.1, the set of estimated error norms in case of fluid-handled interface motion is denoted by

$$[e]^\mathcal{F} = \{e_{\Gamma \cup \text{I}}^\mathcal{F}, e_\Gamma^\mathcal{F}, e_\text{I}^\mathcal{F}, e_\text{I}^\mathcal{S}\}, \quad (4.29)$$

where the superscript \mathcal{F} at $[e]^\mathcal{F}$ indicates the fact that the fluid field has been chosen as master field. Accordingly, the set of estimated error norms $[e]^\mathcal{S}$ for the case of structure-handled interface motion is given as

$$[e]^\mathcal{S} = \{e_{\Gamma \cup \text{I}}^\mathcal{S}, e_\Gamma^\mathcal{S}, e_\text{I}^\mathcal{S}, e_\text{I}^\mathcal{F}\}. \quad (4.30)$$

Details on the computation of the error norms e_β^α , $\alpha \in \{\mathcal{F}, \mathcal{S}\}$ and $\beta \in \{\Gamma \cup \text{I}, \Gamma, \text{I}\}$, that are necessary to provide a scalar value for each estimated error, will be given in Section 4.2.3. Note, that degrees of freedom, that are subject to DIRICHLET boundary conditions, are excluded from the error estimation, since their values are prescribed and, thus, exact.

4.2.2 Adapting the Time Step Size

Every error norm in the set $[e]^{\text{ma}}$, $\text{ma} \in \{\mathcal{F}, \mathcal{S}\}$, needs to be taken into account to compute the new time step size for the coupled problem. In order to transform the set of error norms to a single value for the optimal time step size Δt^* , the following steps are performed: First, a set of optimal scaling factors $[\kappa^*]$ is computed by applying (4.18) to each estimated error norm in the set of errors $[e]^{\text{ma}}$. To account for the accuracy demands in the single fields, the respective tolerance $\varepsilon_l^{\mathcal{S}, \Delta t}$ or $\varepsilon_l^{\mathcal{F}, \Delta t}$ is used. Secondly, a set $[\Delta t^*]$ of time step size suggestions is calculated based on the set $[\kappa^*]$ of optimal scaling factors as well as the algorithmic parameters $[\kappa]$. Finally, the optimal time step size Δt^* of the coupled problem is determined as

$$\Delta t^* = \min [\Delta t^*], \quad (4.31)$$

Equation (4.31) guarantees that the time step size is governed by the subset of degrees of freedom, that is most critical in terms of achieving the desired tolerance $\varepsilon_l^{\mathcal{S}, \Delta t}$ or $\varepsilon_l^{\mathcal{F}, \Delta t}$. Doing so, the accuracy demand can be satisfied globally in the entire computational domain.

4.2.3 Practical Considerations

In this section, some practical aspects are discussed and guidelines for the choice of all algorithmic parameters for the adaptive time stepping procedure are given as well as measures of computational efficiency of the proposed scheme are developed. Whenever possible, it is tried to provide theory-indicated justifications for these guidelines. However, this is not possible in every case and, thus, sometimes only rules of thumb or knowledge that has been obtained through extensive numerical studies can be established.

Computation of Norms

To deduce a scalar quantity representing the estimated error, a norm of the error needs to be computed, cf. (4.11) or (4.12) for example. Following the derivations from Section 4.1.3, where space-continuous but time-discrete functions are used, the appropriate norm is the L_2 -norm as previously defined in (3.54). However, in computer codes the required integral needs to be evaluated numerically using a suitable quadrature rule, e.g. GAUSS quadrature. The influence of the mesh size is respected in this quadrature process since it involves an integration over all elements with possibly different sizes. Such an numerical evaluation of the integral in (3.54) gives accurate results, but is also very costly and time-consuming. Hence, one aims at a cheaper norm calculation while paying the price of a reduced accuracy which is acceptable especially in the case of error estimation.

A first simplification can be thought of as approximating the exact L_2 -norm by an area- or volume-weighted 2-norm of the vector of nodal values in two or three dimensions, respectively, i.e. by computing an EUCLIDIAN vector norm where each nodal entry in the vector has been weighted with the area or volume associated to the node. The weighting allows for comparability of norms computed on different meshes. Although this is computationally more efficient than the previous approaches, it still involves numerical effort to compute the weights and to apply the weights to the vector.

In many practical computer codes, the norm computation is just done by EUCLIDIAN vector norms $\|(\bullet)\|_2$, which is also advised by DEUFLHARD and BORNEMANN [58] in the context of time step size adaptivity due to their smoothness properties. To account for the size of the spatial discretization and to allow comparability in the case of mesh refinement, a length scaling is introduced such that

$$\|(\bullet)\|_2 = \sqrt{\frac{\sum_{m=1}^M (\bullet)_m^2}{M}} \quad (4.32)$$

with M being the length of the respective vector (\bullet) . In this thesis, all error norms for time step size adaptivity are computed using (4.32).

Averaging of Increasing Time Step Sizes

As given in (4.19), the increase of the time step size is limited by the user-given factor κ_{\max} . However, this still may lead to *overshooting*, i.e. to situations where the time step size is increased so much that the subsequent time step requires a time step size reduction which finally yields a non-smooth and wiggly evolution of the time step size as demonstrated in the example

in Section 4.3.1. However, a smooth evolution of the time step size is said to be beneficial for the conservation of stability and accuracy properties of time integration schemes, which usually are only guaranteed in the case of a constant time step size. Another key player is the time step size dependency of the fluid stabilization [21, 100], for example. Changes in the time step size possibly change the stabilization parameter and, thus, the amount of stabilization brought into the system. However, changing the amount of stabilization too rapidly leads to vast variations in the stabilization, which should be avoided to maintain good stability properties of the fluid solution. As a remedy, an averaging procedure is applied in case of an increasing time step size. The new time step size Δt_{n+1} is computed as a linear combination of the newly determined optimal time step size Δt^* and M previous time step sizes via

$$\Delta t_{n+1} = \gamma_{n+1} \Delta t^* + \sum_{m=0}^{M-1} \gamma_{n-m} \Delta t_{n-m} = \gamma_{n+1} \Delta t^* + \gamma_n \Delta t_n + \gamma_{n-1} \Delta t_{n-1} + \dots \quad (4.33)$$

with user-chosen weights γ_i satisfying $\sum_{m=0}^M \gamma_{n-m+1} = 1$ and $\gamma_{n-m+1} = 0 \forall m > M$. This can effectively reduce the occurrence of overshooting events. Furthermore, it fosters a smooth evolution of the time step size, which seems to be beneficial when having in mind the time step size dependency of the fluid stabilization or stability and accuracy properties of the time integration schemes, that are only guaranteed if the ratio $\Delta t_{n+1}/\Delta t_n$ is not too large [111].

Since a decrease of the time step size is always triggered by a violation of the accuracy demand, i.e. $\|\mathbf{1}_{n+1}^\diamond\|_{L_2} > \varepsilon_t^{\Delta t}$, it is crucial to allow for the desired decrease immediately in order to satisfy the accuracy demand. Henceforth, the averaging procedure is not applied in case of a decrease of the time step size. The time step size Δt_n of the time step, that needs to be repeated, is set to the decreased value Δt^* immediately without any further modifications.

Choice of Algorithmic Parameters

The upper time step size limit Δt_{\max} can often be chosen based on the desired temporal resolution, meaning that the time step size may not exceed a certain value in order to be able to capture the transient behavior of the problem at hand. Usually, the computational engineer has a notion of the transient behavior of the expected solution and, thus, can determine Δt_{\max} quite easily.

The lower time step size limit Δt_{\min} should be chosen small enough such that the algorithm is able to satisfy the accuracy demand. However, this value may not be chosen arbitrarily small. For example, numerical results for a STOKES problem reported in BOCHEV *et al.* [22] indicate that the stabilizing effect of the fluid stabilization deteriorates if the time step size is too small in comparison to the spatial grid resolution. A detailed analysis of this issue has been performed by BOCHEV *et al.* [21], which can be seen as providing a rule of thumb to choose the lower time step size limit Δt_{\min} given a stabilization technique and a certain grid resolution. Since the fluid stabilization is not in the focus of this thesis, the interested reader is referred to literature for further information.

Every simulation is started using the lower bound Δt_{\min} of the time step size as initial value, i.e. $\Delta t_0 = \Delta t_{\min}$. This ensures the satisfaction of the accuracy demand (4.12) right from the beginning of the simulation. Since the initial time step size is chosen as the lower bound, a rapid increase is expected during the first couple of time steps. If the time step size does not increase at

the beginning of the simulation, the lower bound Δt_{\min} is not low enough to achieve the desired error level $\varepsilon_l^{\Delta t}$. Either the tolerance $\varepsilon_l^{\Delta t}$ needs to be reviewed or Δt_{\min} has to be decreased to allow for satisfaction of the accuracy demand (4.12) throughout the entire simulation.

The parameters κ_{\min} and κ_{\max} , that limit the ratio of two subsequent time step sizes, are chosen in the ranges of $0.1 - 0.5$ and $1.5 - 5$ [110], respectively. Depending on the time integration scheme, these limitations have to be chosen even tighter, e.g. for the *backward differentiation formula of 2nd order* the increase needs to be limited to $\kappa_{\max} = 1 + \sqrt{2}$ [110]. Typical values for the safety factor κ_s are in the range of $0.8 - 0.95$ [110]. Note that the safety factor $\kappa_s < 1$ leads to reductions of Δt_n even if the estimated error matches the tolerance quite well as desired in (4.14). Although seeming to be counterintuitive, choosing $\kappa_s < 1$ speeds up the computation a lot by avoiding many time step repetitions, that would become necessary if the error was only slightly larger than the tolerance.

As indicated in the outline of the adaptive time stepping algorithm in Section 4.1.3, multiple repetitions of the same time step $t_n \rightarrow t_{n+1}$ may be necessary. However, the number of allowed repetitions is limited. The reason for this is twofold: On the one hand, an infinite number of repetitions needs to be avoided which would lead to stalling of the simulation. On the other hand, an increased number of repetitions may indicate, that the time step size has been far too large just before the transient event. In such cases, the algorithm may have stepped over the beginning of the transient event with a very large time step size and, thus, the event has not been resolved properly. In case that the algorithm reaches the maximum number of repetitions, the simulation is aborted with an error. It is advised to rerun the simulation with a decreased tolerance $\varepsilon_l^{\Delta t}$ and a decreased upper limit Δt_{\max} of the time step size. Numerical studies have shown that a value of at most five repetitions of a single time step is sufficient even in very demanding cases, cf. examples in Section 4.3.

Choice of Error Tolerances

The tolerance for the local discretization error is usually related to a characteristic quantity $(\bullet)_{char}$ of the solution, i.e. a characteristic velocity magnitude $u_{char}^{\mathcal{F}}$ of the fluid flow field or a characteristic displacement magnitude $d_{char}^{\mathcal{S}}$ of the solid body, by a relative tolerance $\varepsilon_{rel}^{\Delta t}$. The tolerance for the local error is then given as

$$\varepsilon_l^{\Delta t} = \varepsilon_{rel}^{\Delta t} (\bullet)_{char}$$

with $(\bullet)_{char}$ being the characteristic quantity of the respective field. Typically, values of $\varepsilon_{rel}^{\Delta t}$ are in the range of $10^{-3} - 10^{-4}$ [242].

SHAMPINE [219] provides insight on the relation of local errors to global errors and summarizes some rules of thumb, that are often applied to choose error tolerances in practical computations. His assumptions of a *'moderately stable problem'* and a *'small tolerance'* are supposed to hold in the present work. Choosing the tolerance for the local error somewhat smaller than the desired tolerance for the global error seems to be quite satisfactory. By studying bounds as well as estimates of the global error, SHAMPINE [219] concludes that the global error will be comparable to the tolerance of the local error. A similar result is given by DEUFLHARD and BORNEMANN [58]. To further increase confidence in the numerical solution, the problem can also be solved with a sequence of local error tolerances and an evaluation of the results by consistency [219]. Thereby, differences in the solutions obtained with various error tolerances are

evaluated until they become negligible. Such a process somehow has the notion of a temporal convergence study.

If the error is kept constant per time step as previously presented, it accumulates over the duration of the simulation. In the case of a small tolerance, many time steps are required. This might lead to a larger error at the end time T as if only a few large time steps with a less tight tolerance had been done. This issue can be addressed by a slightly different approach. By introducing the notion of an *error density per unit time*, the tolerance can be adapted in each time step depending on the current time step size Δt_n , such that the accumulated error in the end does not exceed a prescribed value. Such an approach is closely related to methods known as *error per unit step* in literature, cf. [219] for example.

Handling of Convergence Issues of the Nonlinear Solver

In the presence of an exceptionally strong nonlinearity, the nonlinear solver might fail to converge. In such an event, the solution at time t_{n+1} is not available and, thus, cannot be assessed by means of error estimation. This is more likely to be the case if the tolerance $\varepsilon_l^{\Delta t}$ is chosen very loosely. To avoid abortion of the simulation in such cases, the convergence of the nonlinear solver can be taken into account for time step size adaptivity as additional safety and robustness measure.

In case of failure of the nonlinear solver, repeating the current time step with a reduced time step size usually allows for continuation of the simulation. Lacking an error estimate, the time step size is adapted heuristically. In this thesis, the reduced time step size is chosen as $\kappa_{\text{nl}} \Delta t_n$ with the user-given factor $\kappa_{\text{nl}} \in]0, 1[$. Often, the reduction factor κ_{nl} is chosen in the range of 0.5 – 0.8. As in adaptivity based on error estimation, multiple repetitions of the same time step with progressive reductions of Δt_n have to be performed, while the number of repetitions is limited in the same manner as in error estimation-based adaptivity. If the nonlinear solver converged successfully with the decreased time step size, further increase or decrease of Δt_n is again based on error estimation. It seems to make sense to limit the increase of Δt_n for a couple of time steps right after such a solver-based reduction of Δt_n in case that the strong nonlinearity is more pronounced. This might not be the most efficient procedure, but is still more favorable than aborting the simulation and not obtaining any result at all.

It is stressed that this strategy does not replace time step size adaptivity based on error estimation. It is rather an *additional* safety measure that acts on top of the underlying adaptivity approach. It is a pure safety measure to avoid breakdown of the simulation in case of unexpected events and is not advised to be used as a stand-alone algorithm.

Computational Efficiency and Savings

As postulated in (4.11), the primary reason to apply error estimation-based adaptive time stepping is to control the accuracy of the numerical solution. Basically, one can obtain an accurate solution with non-adaptive schemes as well. However, the number of time steps to be computed might be much larger than with an adaptive time stepping scheme. Since the adaptive scheme imposes small time step sizes only when necessary for accuracy reasons, but increases the time step size if possible, it is able to advance in time much faster than a non-adaptive scheme of the

same accuracy. Now, an efficiency measure shall be developed, that allows for a quantification of savings of computational cost when aiming at a given level of accuracy.

A straightforward approach would be based on comparing wall clock time of both the adaptive and the non-adaptive simulation. However, finding a constant time step size, that produces a solution with the desired accuracy, is a non-trivial task. It probably requires many runs of the same simulation with different values for Δt , which all are costly. This approach seems to be infeasible for practical applications.

Assuming that each time step is associated with roughly the same cost, an alternative and much cheaper efficiency measure can be based on the number of time steps, that can be saved by the adaptive scheme. First, the problem is solved with the adaptive time stepping scheme and a given tolerance for the local error. Then, the smallest time step size $\min\{\Delta t_n\}$ that occurred in that simulation is used to compute a fictitious number

$$N^{\text{const}} = \frac{T}{\min\{\Delta t_n\}} \quad (4.34)$$

of time steps that would be necessary in a non-adaptive scheme with the same level of accuracy. Note that the denominator in (4.34) should only include time step sizes that have been achieved by time step size reduction caused by the algorithm to exclude the small values from the starting phase of the simulation where initially $\Delta t_0 = \Delta t_{\text{min}}$. Now, relative savings in the number of computed time steps δN^{save} are given as

$$\delta N^{\text{save}} = 100\% - \frac{N^{\text{ada}}}{N^{\text{const}}} \quad (4.35)$$

with N^{ada} and N^{const} being the number of time steps performed by the adaptive and non-adaptive scheme, respectively. To account for the extra effort of possible time step size repetitions, repeated time steps need to be included into N^{ada} .

4.3 Numerical Examples

Three numerical examples are used to demonstrate and discuss properties of the adaptive time-stepping procedure for monolithic FSI solvers. In the first two examples, the time step size is adapted based on error estimation in the structure or the fluid field only. The last example uses both fields for time step size adaptivity.

Common to all examples in this thesis is the use of equal-order interpolated linear finite elements for the spatial discretization of the fluid field. *Residual-based* stabilization is applied, namely *Streamline Upwind* PETROV–GALERKIN (*SUPG*) [28], *Pressure-Stabilized* PETROV–GALERKIN (*PSPG*) [123], and a *grad-div* term [46]. The stabilization parameter follows the definition by GRAVEMEIER *et al.* [100]. Details on the specific types of elements as well as the employed time integration schemes are given separately for each example.

4.3.1 Buckling of a Solid Cylindrical Shell Submerged into Fluid

The first example is used to study time step size adaptivity based on error estimation in the structure field only. In order to include the error estimate of the interface degrees of freedom, the structure-handled interface motion algorithm is applied, cf. Section 3.3.2.

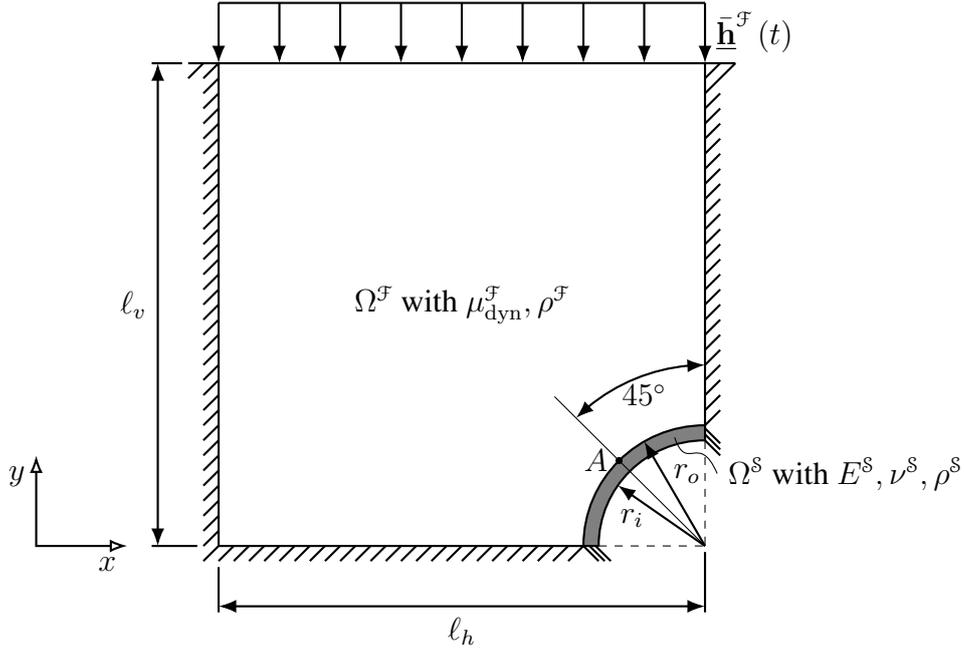


Figure 4.4: Rectangular fluid domain $\Omega^{\mathcal{F}}$ ($\ell_v = 2.0$, $\ell_h = 2.0$) with submerged thin-walled solid cylinder segment $\Omega^{\mathcal{S}}$ ($r_o = 0.5$, $r_i = 0.49$) — The solid is clamped at both ends. No-slip boundary conditions are assumed at the fluid walls and at the interface. The fluid top boundary is loaded with a time-dependent Neumann load $\bar{\mathbf{h}}^{\mathcal{F}}(t)$.

A very thin-walled cylindrical shell segment is submerged into a fluid domain as depicted in Figure 4.4. At the fluid walls and at the fluid-structure interface, a no-slip condition is assumed. The structure is clamped at both ends. Material parameters are listed in Table 4.2. On top of the fluid domain, a time-dependent external traction $\bar{\mathbf{h}}^{\mathcal{F}}(t)$ is applied, that points in negative y -direction and varies in time according to

$$\bar{\mathbf{h}}^{\mathcal{F}}(t) = \begin{cases} \hat{h}^{\mathcal{F}} \cdot \frac{1}{2} (1 - \cos \pi t) \begin{bmatrix} 0 \\ -1 \end{bmatrix} & \text{if } 0 < t \leq 1 \\ \mathbf{0} & \text{if } t > 1 \end{cases}$$

with the peak value $\hat{h}^{\mathcal{F}} = 0.5$. Solid and fluid are initially at rest. Due to the time-dependent external load on the fluid top surface, the structure will buckle if the fluid traction onto the

Table 4.2: Material parameters for solid cylindrical shell under fluid loading

Fluid			Solid		
dynamic viscosity	$\mu_{\text{dyn}}^{\mathcal{F}}$	0.01	YOUNG'S modulus	$E^{\mathcal{S}}$	10^4
density	$\rho^{\mathcal{F}}$	0.01	POISSON'S ratio	$\nu^{\mathcal{S}}$	0.0
			density	$\rho^{\mathcal{S}}$	1.0

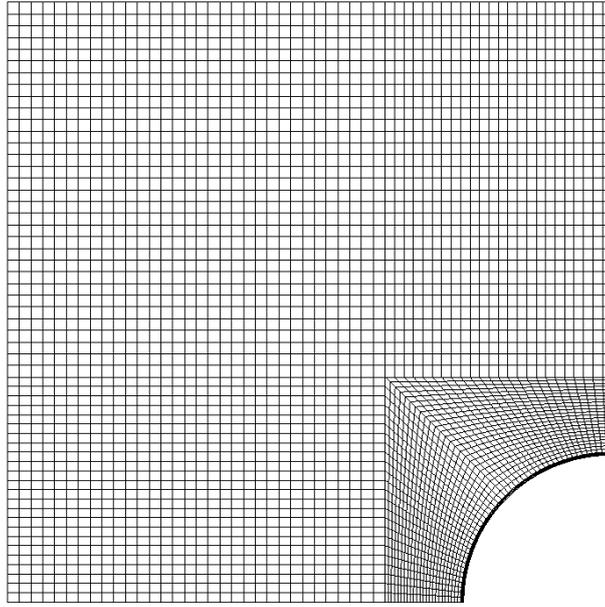


Figure 4.5: Mesh for solid cylindrical shell under fluid loading — The thin-walled cylindrical shell is discretized with 128×4 bilinear quadrilaterals, the fluid domain with 3712 linear equal-order interpolated elements.

structure exceeds its critical value. This will lead to a highly transient motion of the structure, which requires extremely fast and drastic changes in the time step size in order to enable a numerical solution and to resolve the transient behavior accurately.

The structural domain is discretized with 128×4 quadrilateral finite elements, that use a *plain strain* assumption to allow for two-dimensional modelling. To deal with possible locking phenomena, *Enhanced Assumed Strains (EAS)* are utilized. The fluid domain is discretized with 3712 equal-order interpolated P1P1 finite elements, where residual-based stabilizations as detailed above are employed. The finite element mesh is depicted in Figure 4.5. To enable the ALE mesh to deal with localized, but large deformation close to the solid domain, an elasticity model is used for ALE mesh motion, cf. Appendix A.2.3. The ALE domain is divided into two subdomains, both using a ST.-VENANT-KIRCHHOFF material but with different material properties. Softer material properties (YOUNG's modulus $E_{\text{soft}}^{\text{g}} = 1$, POISSON's ratio $\nu_{\text{soft}}^{\text{g}} = 0$) are assigned to the L-shaped subdomain with the cartesian grid away from the interface, cf. Figure 4.5, whereas the smaller portion next to the interface uses a stiffer material with YOUNG's modulus $E_{\text{stiff}}^{\text{g}} = 1000$ and POISSON's ratio $\nu_{\text{stiff}}^{\text{g}} = 0.49$.

Temporal discretization is done with generalized- α time integration in the structure field and fluid field as given in [39] and [129], respectively. The spectral radii are chosen as $\rho_{\infty}^{\text{s}} = 0.9$ and $\rho_{\infty}^{\text{f}} = 0.5$. The adaptive time stepping procedure for monolithic fluid-structure interaction solvers is applied. Different settings for the adaptive time stepping algorithms are studied and described in the following.

Figure 4.6 exemplarily depicts the solution obtained using time step size adaptivity with the ZIENKIEWICZ-XIE error estimator [264] and a tolerance $\varepsilon_l^{\text{s}, \Delta t} = 10^{-3}$ at selected time steps.

The y -components d_y^s and \dot{d}_y^s of the structural displacement and velocity of its center point A are plotted over time in Figure 4.7. The buckling event, that starts at time $t = 0.75466$, cf. Figure 4.6(b), can be seen clearly by large amplitudes and fast changes in displacement and velocity.

Monitoring of the Local Discretization Error with Different Error Estimators

To test error estimation, simulations with constant time step sizes are done while error estimation is active. Figure 4.8 compares the estimated error in the structure field for the available error estimators, namely the ZIENKIEWICZ-XIE [264] approach as well as comparisons to explicit methods like ADAMS-BASHFORTH-2, explicit EULER and the central difference scheme. Two different values for the time step size are used, namely $\Delta t = 10^{-2}$ and $\Delta t = 10^{-3}$. The smaller time step size produces smaller errors, which is consistent with theory. The evolution of the estimated error corresponds very well to the dynamics of the system. In the beginning, when the system is still at rest and the load is increased slowly and smoothly, the error stays at quite a low level. Around time $t = 0.75466$, the buckling of the structure starts, cf. Figure 4.6(b). This leads to transient behavior, which in turn results in an increased error if the time step size is kept constant. After the load has been removed, the shell returns to its original shape and the system calms down quickly, which results in low error values, again.

Although minor quantitative differences between the applied error estimation methods can be seen, the global qualitative behavior is the same. All compared estimation methods behave similarly w.r.t. the increase of the error during the transient buckling event and w.r.t. an decrease of the error when the system approaches its final steady state, see Figure 4.8. No distinct recommendation for one of the error estimators can be given.

Time Step Size Averaging in case of a Time Step Size Increase

The phenomenon of overshooting as described in Section 4.2.3 is studied. Therefore, a reference simulation without averaging of increasing time step sizes as given in (4.33), i.e. the weight $\gamma_{n+1} = 1$, is performed. Its evolution of the time step size Δt is referred to as *avg0* in Figure 4.9. Right in the beginning, a steep increase of Δt can be seen. This is due to the fact that a very low error results in large increases of the time step size. Due to the choice $\gamma_{n+1} = 1$, the increasing value is always taken as the new time step size to its full extent. Not being slowed down by portions of the previous time step sizes seems to be beneficial in this portion of the simulation. After the buckling event, the time step size increases, but not monotonically. Due to the effect of overshooting, reductions of the time step size have to be performed quite often. Fortunately, this does not result in an massively increased number of time step repetitions in this specific example since a safety factor $\kappa_s = 0.9$ has been chosen. As outlined in Section 4.2.3, this behavior is not desirable. The oscillating error in this portion of the simulation as observed in Figure 4.8 can be seen as a strong indicator that such variations of the time step size may be expected.

To remedy this situation, different averaging strategies *avg0* – *avg4* are applied. The chosen weights for each strategy are listed in Table 4.3. Results are reported in Figure 4.9. The larger the weights of the previous time step sizes, the more the fast increase of Δt at the beginning of the simulation is slowed down. However, the rapid decrease at the beginning of the buckling

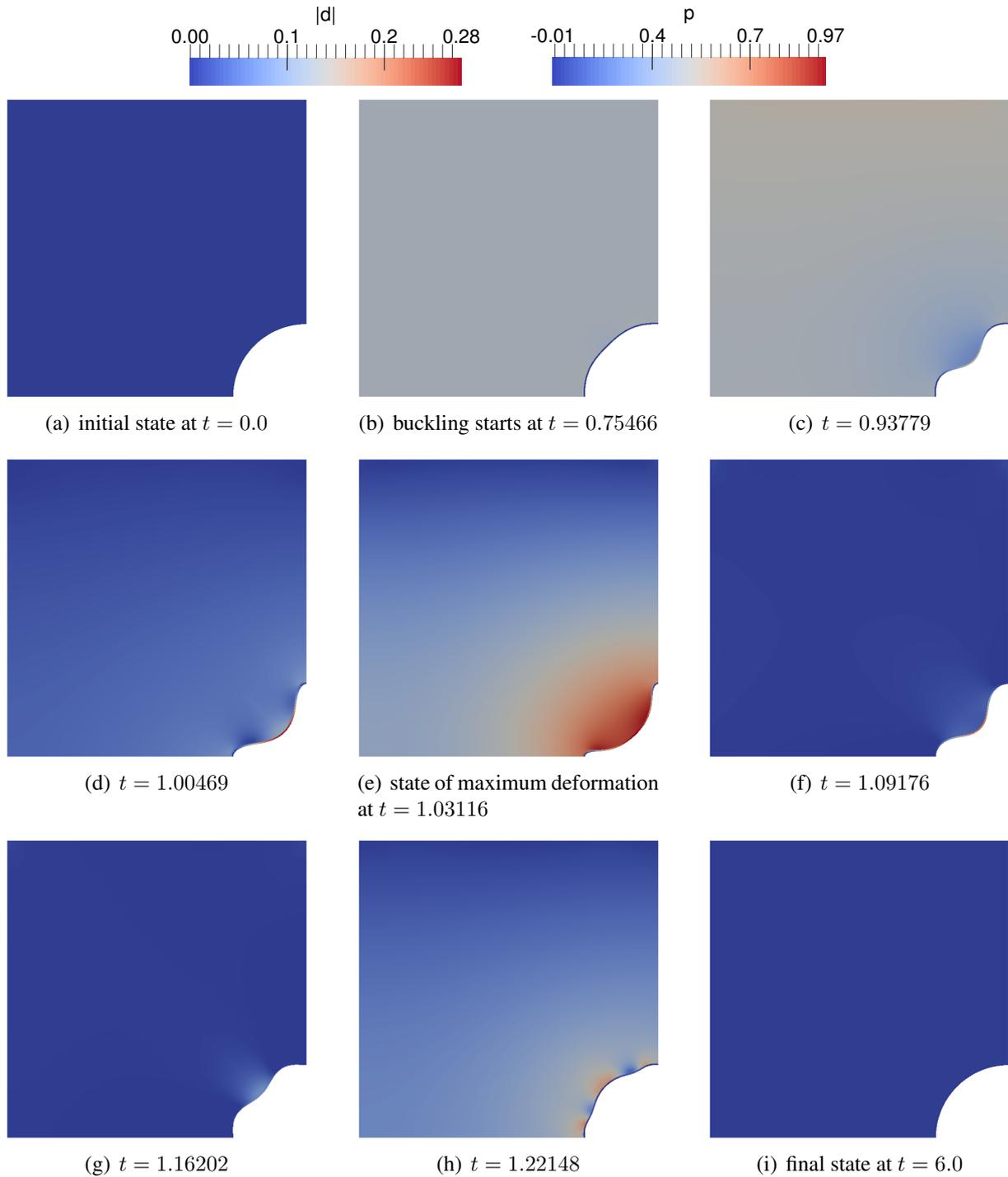


Figure 4.6: Snapshots of the solution of the pressurized cylinder example at selected time steps — The fluid field is colored according to its pressure field values p while the structure field’s color is encoded with the displacement field magnitude denoted by $|d|$. Color scales are calibrated at maximum deformation at $t = 1.03116$. The simulation has been performed with time step size adaptivity using the ZIENKIEWICZ-XIE error estimator [264] in the structure field with a tolerance $\varepsilon_l^{S,\Delta t} = 10^{-3}$.

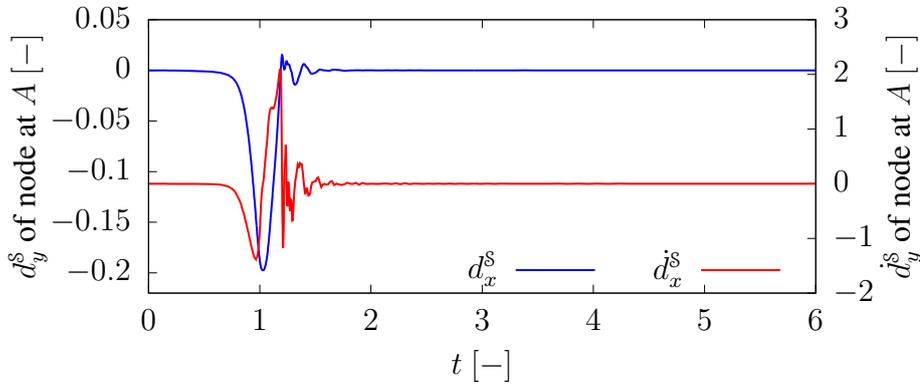


Figure 4.7: Vertical displacement and velocity at the location A of a solid cylindrical shell under fluid loading — Both the displacement d_y^s and the velocity \dot{d}_y^s start from zero values in the initial configuration. During the buckling event, large amplitudes and fast changes can be seen. Both quantities find back to their zero values in the final state which is fully at rest.

Table 4.3: Averaging strategies for solid cylindrical shell under fluid loading — Averaging strategies *avg0* – *avg4* are parametrized with the respective weights γ_i . The last column reports the number of time steps needed with each strategy.

Strategy	γ_{n+1}	γ_n	γ_{n-1}	# time steps
<i>avg0</i>	1.0	0.0	0.0	414
<i>avg1</i>	0.7	0.3	0.0	426
<i>avg2</i>	0.5	0.5	0.0	438
<i>avg3</i>	0.3	0.7	0.0	486
<i>avg4</i>	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	487

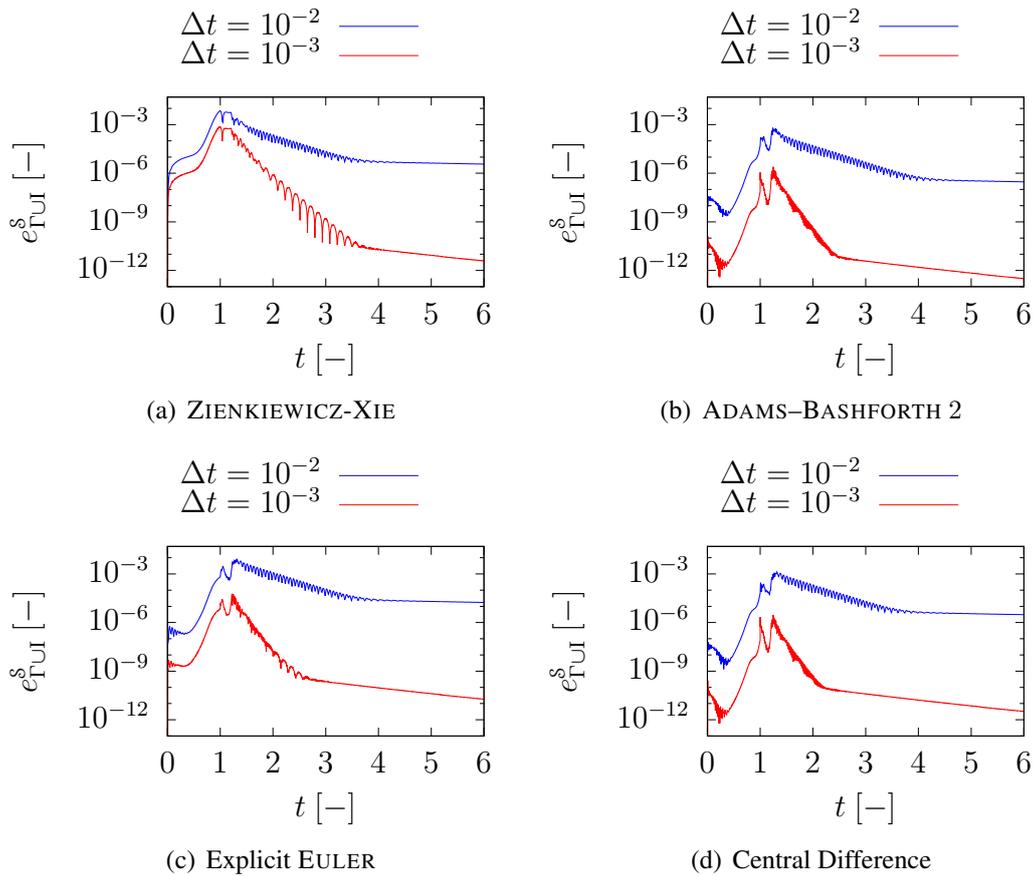


Figure 4.8: Monitoring of the estimated local error in the structure for a solid cylindrical shell under fluid loading — Keeping the time step size constant throughout the simulation, but recording the estimated local error demonstrates qualitatively similar behavior for all applied error estimation schemes. Quantitative differences are minor. All schemes estimate a larger error during the buckling event and smaller errors when the system is close to a steady state. A smaller time step size results in smaller errors. Overall, expected behavior is reproduced.

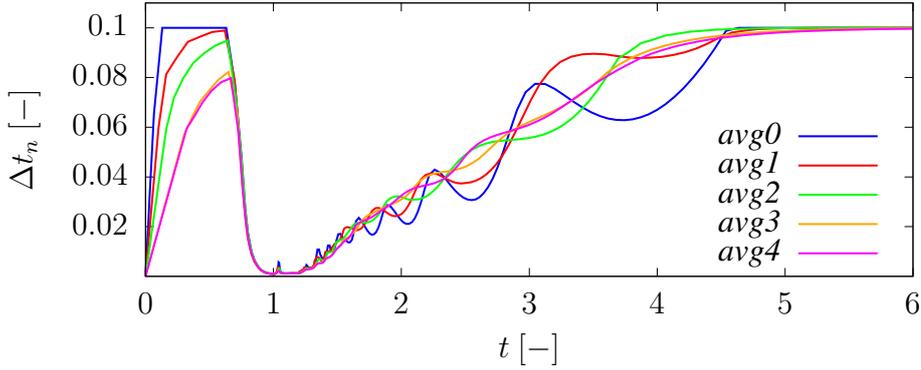


Figure 4.9: Averaging of increasing time step sizes for a solid cylindrical shell under fluid loading — Averaging increasing time step sizes as proposed in (4.33) slows down the increase, but also reduces the number of overshooting events. Strategy *avg3* produces a reasonable evolution of Δt_n compared to the case *avg0* without averaging. Weights γ_i of all strategies are detailed in Table 4.3.

event is captured by all simulations equally well since averaging is only applied for increasing, but not for decreasing the time step size. After the buckling event, averaging helps to reduce the number of overshooting incidences at the cost of a mildly increased number of time steps, that need to be computed, cf. last column of Table 4.3. For this example, strategy *avg3* with the choice of $\gamma_{n+1} = 0.3$ and $\gamma_n = 0.7$ seems to be a good trade-off between robustness, overshooting and increased computational cost.

Evolution of Time Step Size

Finally, the use of different error estimators for adapting the time step size is studied. The same error estimators as for the monitoring of the error are used, namely the ZIENKIEWICZ-XIE [264] approach as well as comparisons to explicit methods like ADAMS-BASHFORTH-2, explicit EULER and the central difference scheme. Two different tolerances $\varepsilon_l^{s,\Delta t} \in \{10^{-3}, 10^{-4}\}$ are used. The remaining algorithmic parameters required to evaluate (4.19) are globally chosen as follows: $\kappa_{\min} = 0.1$, $\kappa_{\max} = 2.0$, $\kappa_s = 0.9$, $\Delta t_{\min} = 10^{-6}$, $\Delta t_{\max} = 10^{-4}$. The number of time step repetitions is limited to five. In case of an increasing time step size, $\gamma_{n+1} = 0.3$ and $\gamma_n = 0.7$ are used as weights for the averaging procedure (4.33) based on previous observations.

Evolutions of the time step size for all these cases are reported in Figure 4.10. All error estimators produce sequences of time step sizes, that behave similarly. The increase in the beginning, a massive reduction of the time step size during the buckling event as well as the final increase towards the end of the simulation is common to all procedures. However, the actual value of Δt varies as expected since different approaches result in different error estimates. Furthermore, the cases with the tighter tolerance $\varepsilon_l^{s,\Delta t} = 10^{-4}$ result mostly in smaller values of Δt_n than for $\varepsilon_l^{s,\Delta t} = 10^{-3}$.

Remark 4.3.1 Error estimation with the central difference scheme and a tolerance $\varepsilon_l^{s,\Delta t} = 10^{-3}$ leads to a series of time step sizes Δt_n , that steps so far over the beginning of the buckling

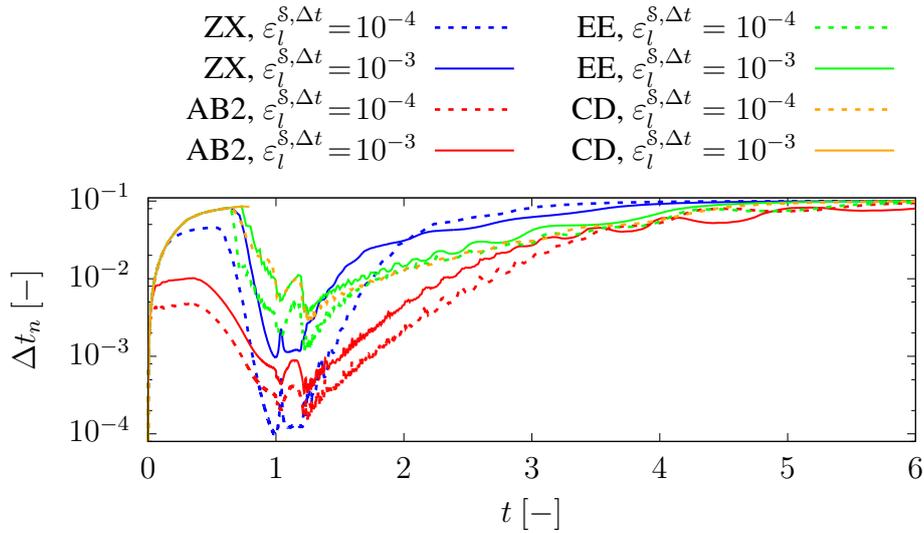


Figure 4.10: Evolution of time step size Δt for solid cylindrical shell under fluid loading — Four different error estimators are used, namely ZIENKIEWICZ-XIE (ZX), ADAMS-BASHFORTH-2 (AB2), explicit EULER (EE), and the central difference scheme (CD). All of them capture the buckling event very well by reducing the time step size. In general, cases with a tighter tolerance result in smaller time step sizes as expected. Using CD with $\varepsilon_l^{s,\Delta t} = 10^{-3}$ fails due to invalid mesh distortion.

event, that it cannot be captured properly. This results in inadmissible mesh distortion, which leads to failure of the simulation. Even the most robust mesh motion schemes could not prevent these issues. To avoid inadmissible mesh distortion, either a reduced maximum time step size Δt_{\max} or a tighter tolerance $\varepsilon_l^{s,\Delta t}$ is necessary. Since this choice would corrupt comparability, this case is excluded from the remainder of the discussion.

In the adaptive algorithm, the number of time step repetitions is limited to five. However, such repetitions occur rarely and only at the beginning of the buckling event, when fast and massive reductions of Δt_n are necessary. The case with explicit EULER as auxiliary scheme for error estimation is an exception, where repetitions are necessary also throughout the buckling event. The series of time steps based on error estimation with ADAMS-BASHFORTH-2 as auxiliary scheme does not allow the time step to grow that much before the buckling event. In addition, it starts with time step size reductions quite early and, thus, does not need to perform any time step repetitions at all. For all error estimators, a single repetition is sufficient for most cases, while even the utmost reductions of Δt_n require two repetitions at most.

Computational savings according to (4.35) are reported in Table 4.4. Compared to the case of a non-adaptive time stepping scheme, the adaptive time stepping scheme reduces the number of time steps to be computed by about 80% – 95% in this example if the same level of accuracy is desired.

Table 4.4: Computational savings for a solid cylindrical shell under fluid loading — Based on N^{ada} , the number of time steps of the adaptive simulation, computational savings δN^{save} are estimated using (4.35). In addition to controlling accuracy, the adaptive scheme reduces the number of time steps to be computed by roughly 80% – 95% which is a huge gain w.r.t. efficiency.

Error estimator	$\varepsilon_l^{S,\Delta t}$	N^{ada}	N^{const}	δN^{save}
ZIENKIEWICZ-XIE	10^{-3}	497	6171	92.0%
ZIENKIEWICZ-XIE	10^{-4}	3304	61903	94.7%
ADAMS-BASHFORTH-2	10^{-3}	1654	17683	90.6%
ADAMS-BASHFORTH-2	10^{-4}	3385	37958	91.1%
Explicit EULER	10^{-3}	315	1765	82.1%
Explicit EULER	10^{-4}	511	5000	89.8%
Central differences	10^{-3}	—	—	—
Central differences	10^{-4}	354	2143	83.5%

Table 4.5: Material parameters for elastic wall in channel flow

Fluid			Solid		
dynamic viscosity	$\mu_{\text{dyn}}^{\mathcal{F}}$	0.01	YOUNG's modulus	$E^{\mathcal{S}}$	500.0
density	$\rho^{\mathcal{F}}$	1.0	POISSON's ratio	$\nu^{\mathcal{S}}$	0.0
			density	$\rho^{\mathcal{S}}$	1.0

4.3.2 An Elastic Wall in a Channel Flow

To analyze the behavior of the fluid error estimator as well as its influence on the evolution of the time step size, an elastic wall is put into a channel flow. Figure 4.11 depicts the domain of interest by means of different views including all geometric features and boundary conditions. The solid is modelled as NEO-HOOKEan material, while the fluid is assumed to be a NEWTONian fluid. Material parameters are listed in Table 4.5. At $x = 0$, the x -component $\bar{u}_x^{\mathcal{F}}(y, z, t)$ of the inflow velocity follows the parabolic profile

$$\bar{u}_x^{\mathcal{F}}(y, z, t) = \hat{u}^{\mathcal{F}}(t) \left(1 - \frac{4y^2}{(h^{\mathcal{F}})^2}\right) \left(1 - \frac{4z^2}{(b^{\mathcal{F}})^2}\right)$$

with the time-dependent peak value

$$\hat{u}^{\mathcal{F}}(t) = \begin{cases} \hat{u}^{\mathcal{F}} \cdot 0.5 \left(1 - \cos\left(\frac{\pi}{6}t\right)\right) & \text{if } 0 < t \leq 6 \\ \hat{u}^{\mathcal{F}} & \text{if } 6 < t \leq 20 \\ \hat{u}^{\mathcal{F}} \left(1 + \cos\left(\frac{\pi}{10}(t - 20)\right)\right) & \text{if } 20 < t \leq 60 \\ \hat{u}^{\mathcal{F}} & \text{if } 60 < t \leq 80 \\ \hat{u}^{\mathcal{F}} \left(1 + \cos\left(\frac{\pi}{10}(t - 80)\right)\right) & \text{if } 80 < t \leq 120 \end{cases}$$

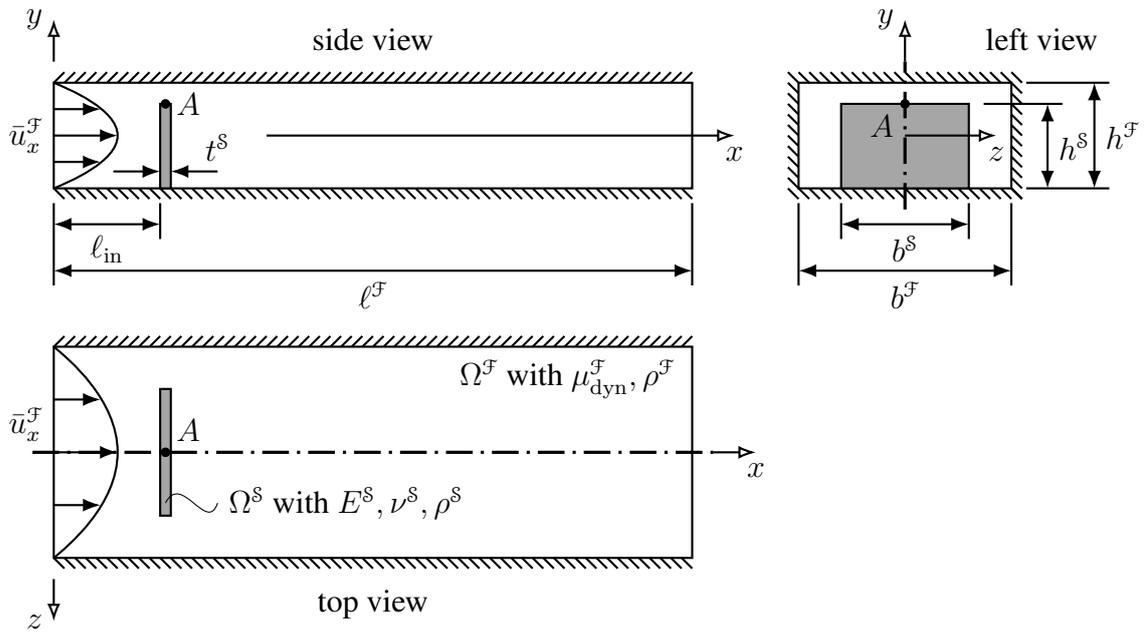


Figure 4.11: Geometry and boundary conditions of an elastic wall in a channel flow — An elastic wall Ω^S ($b^S = 0.6, h^S = 0.4, t^S = 0.05$) is put into a fluid channel Ω^F ($\ell^F = 3.0, b^F = 1.0, h^F = 0.5$) at $\ell_{in} = 0.5$ and clamped at its bottom. A time-dependent parabolic inflow profile is prescribed, while the outlet is traction-free. The channel walls are subject to no-slip boundary conditions. The top, center point of the elastic wall is denoted by A .

with $\hat{u}^{\mathcal{F}} = 0.1$, cf. Figure 4.12(a). The components in y - and z -direction are set to zero. Top and bottom walls as well as lateral walls are subject to a no-slip boundary condition. On the outflow area, a zero-traction boundary condition is applied. Due to the symmetry of the problem, only the portion $z \leq 0$ of the channel is modelled. Appropriate symmetry conditions are applied to the xy -plane.

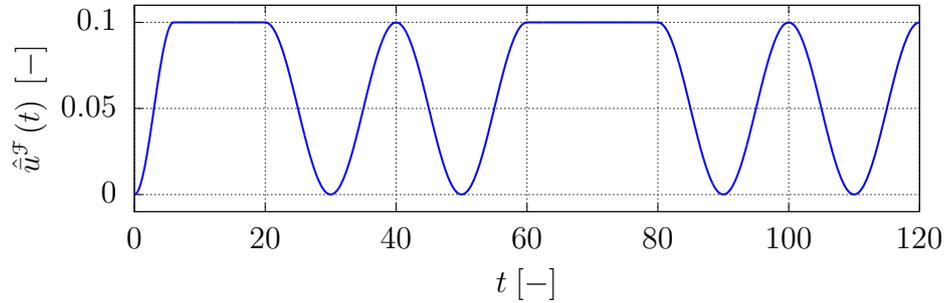
The structural domain is discretized with 2125 nodes grouped to $4 \times 24 \times 16$ Hex8 F-Bar finite elements [47], the fluid domain with 26017 nodes forming 22704 equal-order interpolated Hex8 finite elements using residual-based stabilization as already detailed above.

Temporal discretization is done with generalized- α time integration in structure and fluid field, see [39] and [129], respectively. The spectral radii are chosen as $\rho_{\infty}^s = 0.8$ and $\rho_{\infty}^{\mathcal{F}} = 0.5$. Different tolerances $\varepsilon_l^{\mathcal{F},\Delta t} \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ are used for adapting the time step size based on the estimation of the temporal discretization error in the fluid field, which is performed by comparison of the implicit solution to an explicit one obtained with an ADAMS–BASHFORTH-2 scheme. The remaining algorithmic parameters required to evaluate (4.19) are globally chosen as follows: $\kappa_{\min} = 1/3$, $\kappa_{\max} = 3.0$, $\kappa_s = 0.9$, $\Delta t_{\min} = 10^{-5}$, $\Delta t_{\max} = 1.5$. The number of time step repetitions is limited to five. No averaging of increasing time step sizes is applied.

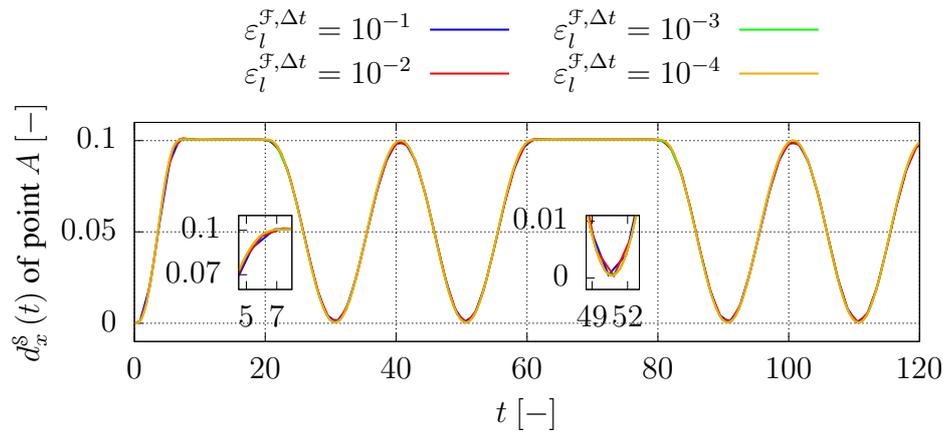
The x -displacement of the center point A at top of the elastic wall, cf. Figure 4.11, is depicted in Figure 4.12(b). Due to the low inflow velocity, the flow does not detach from the solid and, thus, the solid deformation is able to follow the prescribed inflow velocity shown in Figure 4.12(a) very closely. Figure 4.13 reports a snapshot at the state of maximum deflection at $t = 16.7$, which is within the first stationary period of the inflow velocity, cf. Figure 4.12(a).

Figure 4.14 reports the evolution of the time step size Δt_n over time for all chosen tolerances. In case of the loosest tolerance $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-1}$, the time step size starts with the initial value $\Delta t_0 = \Delta t_{\min}$ and increases monotonically until it reaches the maximum bound Δt_{\max} . Since the demanded accuracy is quite low, the simulation does not require any reduction of Δt_n throughout the entire simulation. For $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-2}$, a minor reduction is necessary only during the initial phase of the simulation. During the transient phases, the time step size needs to be reduced slightly once in a while, taking the smallest value $\Delta t_n = 1.47161 < \Delta t_{\max} = 1.5$, which is a reduction by less than 2%. Remarkable changes of the time step size occur for $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-3}$ and $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-4}$. During the stationary phases, the time step size approaches the maximum bound $\Delta t_{\max} = 1.5$, but during the transient inflow the time step size is reduced massively. As expected, lower values for Δt_n are obtained for the lower tolerance $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-4}$. Common to the cases $\varepsilon_l^{\mathcal{F},\Delta t} = \{10^{-2}, 10^{-3}, 10^{-4}\}$ is the fact, that the minimal time step size resulting from time step size reductions occurs in the initial phase of the simulation. In that phase, the increasing inflow velocity needs to be propagated through the entire domain, that is initially at rest. Later, when the flow field has already been established and just needs to change its velocity, the time step size reduction to slightly larger values is sufficient to achieve the desired level of accuracy.

The maximally allowed number of repetitions of time steps due to violations of the accuracy demand (4.12) is limited to five, again. No repetitions are necessary for the cases $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-1}$ and $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-2}$, where no or only very little reductions of Δt_n are necessary. For the smaller tolerances $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-3}$ and $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-4}$, the number of actual time step repetitions is reported in Figure 4.15. Occurrence of time step repetitions coincides with massive reductions of the time



(a) Prescribed inflow velocity



(b) Tip displacement of elastic wall

Figure 4.12: Temporal evolution of the prescribed inflow velocity and the resulting tip displacement of point A of an elastic wall in a channel flow — *Top*: Starting with a value of zero, the inflow velocity is smoothly increased to its maximum value $\hat{u}^{\mathcal{F}} = 0.1$. Portions with constant and varying inflow velocity alternate in order to trigger the adaptive time stepping algorithm. *Bottom*: The deformation of the elastic wall follows the prescribed inflow velocity very closely. Results differ only slightly for different tolerances, cf. zooms around $t = 6$ and $t = 50$. The location of the monitored point A is specified in Figure 4.11.

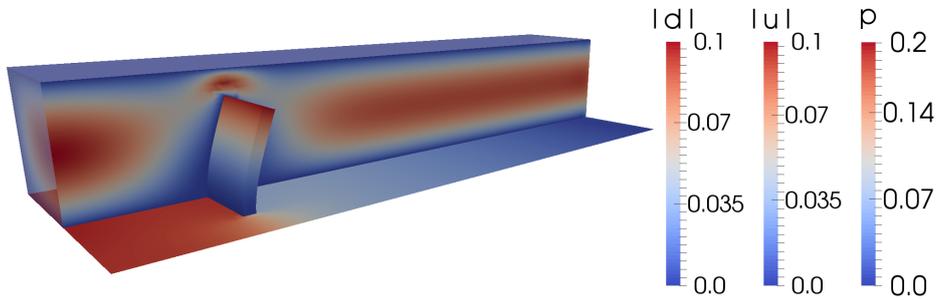


Figure 4.13: State of maximum deformation of an elastic wall in a channel flow — The solution obtained with $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-4}$ is depicted during a phase of constant inflow velocity at time $t = 16.7$. The elastic wall is colored according to its displacement magnitude denoted by $|d|$. The vertical fluid plane shows a contour plot of the velocity magnitude denoted by $|u|$, while the bottom plane illustrates the pressure field denoted by p .

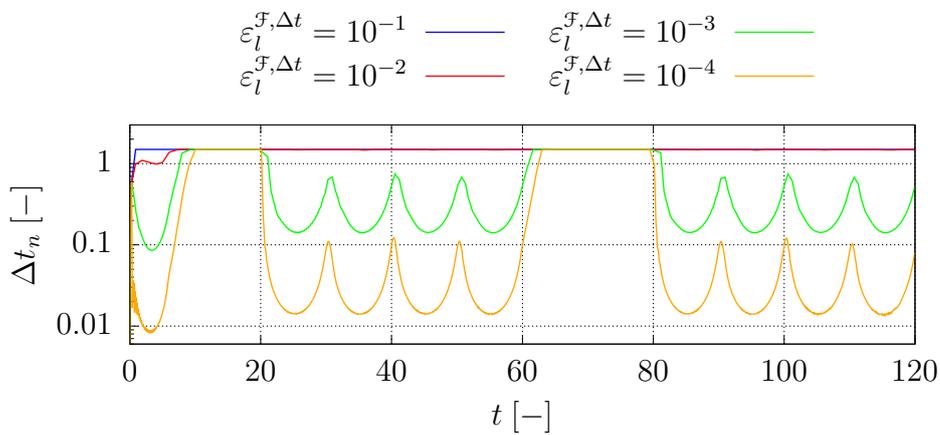


Figure 4.14: Evolution of time step size Δt for an elastic wall in a channel flow — Four different tolerances $\varepsilon_l^{\mathcal{F},\Delta t}$ are used. The choice $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-1}$ produces an immediate increase of Δt_n to Δt_{\max} and uses that value throughout the entire simulation. Lower tolerances use Δt_{\max} only during the stationary phases, while they require a reduction of Δt_n in transient phases. In general, cases with a tighter tolerance result in smaller time step sizes as expected.

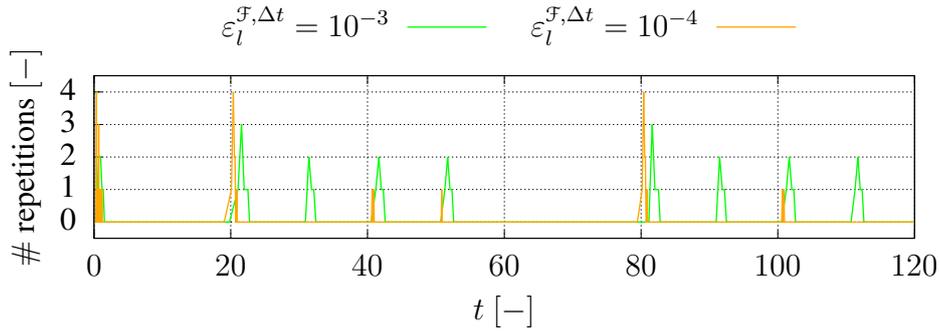


Figure 4.15: Necessary number of time step repetitions for an elastic wall in a channel flow — Only the cases $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-3}$ and $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-4}$ require repetitions of time steps when the time step size needs to be decreased significantly. Since $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-4}$ results in larger decreases, more repetitions are necessary for that case.

Table 4.6: Computational savings for an elastic wall in a channel flow — Based on N^{ada} , the number of time steps of the adaptive simulation, computational savings δN^{save} are estimated using (4.35). Computational savings are more pronounced in case of smaller tolerances.

$\varepsilon_l^{\mathcal{S},\Delta t}$	N^{ada}	N^{const}	δN^{save}
10^{-1}	91	80	-13.8%
10^{-2}	93	122	23.8%
10^{-3}	456	1400	67.4%
10^{-4}	4225	14116	70.1%

step size Δt_n , cf. Figure 4.14. More repetitions are necessary for $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-4}$, since the time step size needs to be reduced by two orders of magnitude instead of only one for $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-3}$.

Computational savings δN^{save} in terms of a reduced number of time steps to be computed are reported in Table 4.6. Obviously, the computational savings δN^{save} as defined in (4.35) depend on the tolerance $\varepsilon_l^{\mathcal{F},\Delta t}$. If a very accurate solution is required, i.e. a low tolerance $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-4}$ is chosen, the adaptive time stepping enables huge computational savings up to $\delta N^{\text{save}} = 70.1\%$ by increasing the time step size during the stationary phases. As already seen in the evolution of the time step size, cf. Figure 4.14, the quite loose tolerance $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-2}$ does not lead to significant reductions of the time step size Δt_n . Only a tiny reduction is required during the starting phase of the simulation. Hence, computational savings are rather small for this case. Only for the loosest choice $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-1}$, the adaptive time stepping seems to be inefficient, since it needs more time steps. This is only due to the initial phase of the simulation, which is necessary to increase the time step from its initial value $\Delta t_0 = \Delta t_{\min}$ up to a suitable value. For the loosest tolerance, this suitable value is limited by Δt_{\max} . Since the time step size is never decreased throughout the simulation, the additional steps from the initial increase of the time step size are seen as additional cost by (4.35), resulting in additional 13.8% time steps to be computed.

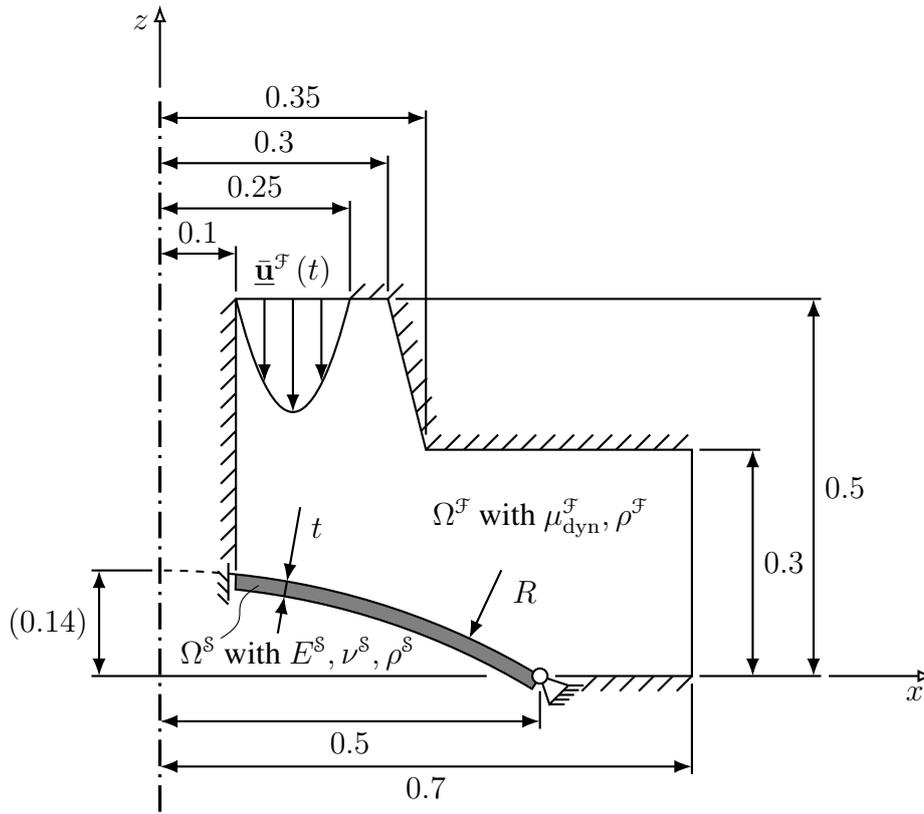
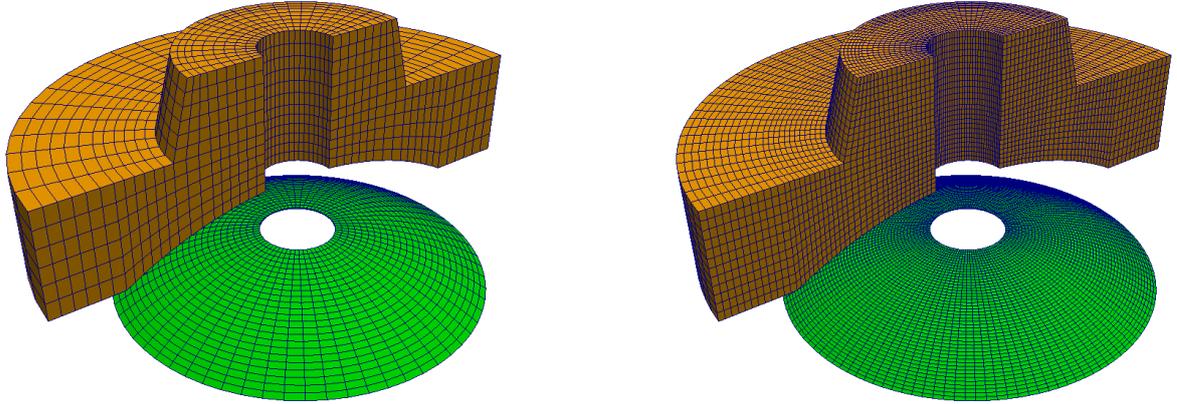


Figure 4.16: Geometry and boundary conditions of gasket-like example — The fluid domain Ω^f is enclosed by the inflow boundary with the prescribed inflow velocity profile $\underline{u}^f(t)$, the outflow boundary with a zero-traction boundary condition, walls with no-slip conditions and the fluid-structure interface. The structural domain Ω^s is a segment of a thin-walled sphere. It can slide along z -direction at its inner radius and rotate around its circumferential direction at its outer radius. The solid shell is characterized by its radius $R = 0.96$ and its thickness $t = 0.0032$ resulting in a slenderness ratio $R/t = 300$. The problem exhibits rotational symmetry w.r.t. the z -axis.

4.3.3 Snap-Through of a Gasket

Finally, error estimation in both the fluid and the structure field are used to adapt the time step size in the simulation of the snap-through of a gasket. The geometry mimics a machine part with rotational symmetry. It is inspired by an example by RAMM and WALL [197]. A detailed sketch including all geometric dimensions can be found in Figure 4.16. Three-dimensional views with two different meshes, that are used to demonstrate mesh independency of the adaptive time stepping scheme, are shown in Figure 4.17. At the top, fluid inflow is prescribed, while fluid outflow is in radial direction. Moreover, the fluid domain is confined by a rigid casing and a thin-walled solid cap mimicking a rubber-like gasket. At the fluid walls and at the FSI interface, a no-slip condition is assumed. The thin-walled spherical solid shell can slide along the z -direction at its inner radius, while it can rotate around the circumferential direction at its outer radius. The solid is modelled as NEO-HOOKEAN material, while the fluid is assumed to be



(a) Coarse mesh with 81504 unknowns in total

(b) Fine mesh with 566496 unknowns in total

Figure 4.17: Views including the mesh of a gasket-like example — Both the fluid and the structure field are meshed with structured grids using hexahedral elements only. Different mesh resolutions are realized. Details on the number of unknowns are given in Table 4.8.

Table 4.7: Material parameters for gasket-like example

Fluid			Solid		
dynamic viscosity	$\mu_{\text{dyn}}^{\mathcal{F}}$	0.1	YOUNG's modulus	$E^{\mathcal{S}}$	10^5
density	$\rho^{\mathcal{F}}$	100.0	POISSON's ratio	$\nu^{\mathcal{S}}$	0.3
			density	$\rho^{\mathcal{S}}$	100.0

a NEWTONian fluid. The used material parameters are listed in Table 4.7. A time-dependent parabolic inflow profile $\bar{\mathbf{u}}^{\mathcal{F}}(x, y, z, t)$ is prescribed. Its x - and y -components are set to zero, while its z -component follows the spatial profile

$$\bar{u}_z^{\mathcal{F}}(x, y) = -\frac{4}{(0.1 - 0.25)^2} \left(\sqrt{x^2 + y^2} - 0.1 \right) \left(\sqrt{x^2 + y^2} - 0.25 \right) \hat{u}^{\mathcal{F}}(t)$$

with the time-dependent peak value

$$\hat{u}^{\mathcal{F}}(t) = \begin{cases} 0.1 \cdot \hat{u}^{\mathcal{F}} \cdot \frac{1}{2} \left(1 - \cos \frac{\pi t}{2} \right) & \text{if } 0 < t \leq 2 \\ 0.1 \cdot \hat{u}^{\mathcal{F}} & \text{if } 2 < t \leq 5 \\ \hat{u}^{\mathcal{F}} \left(0.1 + 0.9 \cdot \frac{1}{2} \left(1 - \cos \pi(t - 2) \right) \right) & \text{if } 5 < t \leq 6 \\ \hat{u}^{\mathcal{F}} & \text{if } t > 6 \end{cases}$$

with $\hat{u}^{\mathcal{F}} = 1.0$. The temporal evolution of $\hat{u}^{\mathcal{F}}(t)$ is illustrated in Figure 4.18. A zero-traction boundary condition is prescribed at the fluid outflow where an additional NEUMANN *inflow boundary condition* [99] is applied to safely account for possible backflow. Starting from an

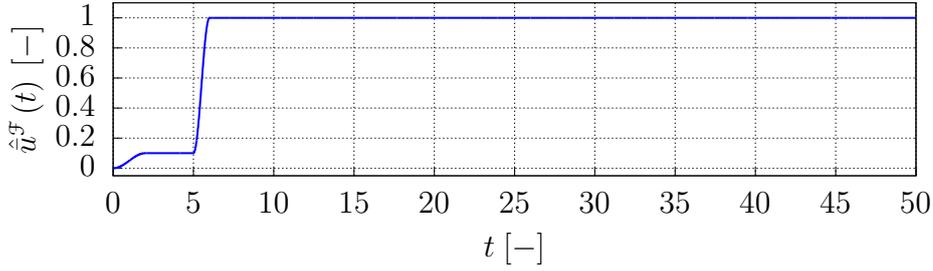


Figure 4.18: Temporal evolution of the inflow velocity $\hat{u}^{\mathcal{F}}$ for a gasket-like example — Starting from a zero velocity field, the inflow velocity is smoothly increased to a low value of $\hat{u}^{\mathcal{F}} = 0.1$. At time $t = 5$, the inflow velocity is rapidly increased to its maximum value $\hat{u}^{\mathcal{F}} = 1.0$ and kept at that value until the end of the simulation at $T = 50.0$.

Table 4.8: Mesh overview and computational resources for the gasket-like example — Two meshes are used, a *coarse* and a *fine* one. Number of unknowns per field as well as total number of unknowns are listed. Computational resources are detailed.

Mesh ID	$n^{\mathcal{S},\text{dof}}$	$n^{\mathcal{F},\text{dof}}$	$n^{\mathcal{S},\text{dof}}$	$n_{\text{total}}^{\text{dof}}$	n^{core}	$n^{\text{dof}}/\text{core}$
<i>coarse</i>	9600	41088	30816	81504	8	10188
<i>fine</i>	37632	302208	226656	566496	64	8851.5

initially resting configuration, the inflow velocity is smoothly increased to $\hat{u}^{\mathcal{F}}(t=2) = 0.1$ and then kept constant until $t = 5$, such that a stationary state can be reached. Under these flow conditions, the load exerted onto the structure is quite small, such that the structure is able to maintain its shape. At $t = 5$, the inflow velocity is smoothly, but rapidly increased to $\hat{u}^{\mathcal{F}}(t=6) = 1.0$ and then kept constant again. This increase of the fluid inflow velocity results in an increase of the load onto the structure above a critical value, such that the thin-walled solid will start buckling. The solid will snap through and finally find a stable equilibrium configuration, such that a stationary flow field can be established.

To demonstrate mesh independency of the adaptive time stepping scheme, two structured and purely hexahedral meshes are used, namely a *coarse* and a *fine* one. Both are depicted in Figure 4.17. Details on the number of unknowns per field and in total are given in Table 4.8. Simulations have been done on the *Xeon* partition of the in-house cluster, cf. Appendix B, where 8 and 64 cores have been used for the *coarse* and *fine* mesh, respectively.

Both the fluid and the structure field use generalized- α time integration schemes with spectral radii chosen as $\rho_{\infty}^{\mathcal{F}} = 0.5$ and $\rho_{\infty}^{\mathcal{S}} = 0.8$, respectively. Both fields contribute to the error estimation and the time step size adaption. In the fluid field, comparison to an auxiliary ADAMS–BASHFORTH-2 scheme with a tolerance $\varepsilon_l^{\mathcal{F},\Delta t}$ is used, while the structure field employs the ZIENKIEWICZ-XIE error estimator [264] with a tolerance $\varepsilon_l^{\mathcal{S},\Delta t}$. The solid’s local error tolerance is varied as $\varepsilon_l^{\mathcal{S},\Delta t} \in \{10^{-3}, 5 \cdot 10^{-4}, 10^{-4}\}$, while the fluid’s one is chosen as $\varepsilon_l^{\mathcal{F},\Delta t} = 10^{-3}$. The remaining algorithmic parameters required to evaluate (4.19) are globally chosen as follows: $\kappa_{\min} = 0.1, \kappa_{\max} = 2.0, \kappa_s = 0.9, \Delta t_{\min} = 10^{-4}, \Delta t_{\max} = 0.2$. The number of time step

repetitions is limited to five. In case of an increasing time step size, $\gamma_{n+1} = 0.3$ and $\gamma_n = 0.7$ are used as weights for the averaging procedure (4.33) based on experience from previous examples.

Figure 4.19 shows a series of snapshots to illustrate the transient solution. Displacement and velocity of a node on the inner radius of the solid are depicted in Figure 4.20 for the *fine* mesh and tolerances $\varepsilon_l^{s,\Delta t} = 5 \cdot 10^{-3}$ and $\varepsilon_l^{f,\Delta t} = 10^{-3}$. Until $t = 5$, the spherical shell is able to maintain an equilibrium state close to its initial configuration, i.e. displacement and velocity are close to zero. After increasing the inflow velocity at $t = 5$, the fluid load onto the structure exceeds a critical value, such that the shell cannot maintain its shape and starts buckling. Hence, large displacements and velocities occur while the shell is transformed to another equilibrium configuration. Due to the viscous damping of the fluid and since the inflow velocity is kept constant for $t \geq 6$, oscillations are damped and a stationary flow field can be established. In this state, the solid maintains a *stable* equilibrium configuration, i.e. the displacement is constant and the velocity vanishes.

Figure 4.21 reports the evolution of time step size for both meshes and all tolerances. Both the coarse and the fine mesh show a similar behavior regarding the evolution of the time step size. In the initial phase, i.e. prior to $t = 5$, the time step size approaches its maximum bound $\Delta t_{\max} = 0.2$. Only the most accurate case with $\varepsilon_l^{s,\Delta t} = 10^{-4}$ requires a slight reduction of Δt_n during this initial phase, all coarser tolerances allow for a monotonic increase. As the buckling starts, all choices for $\varepsilon_l^{s,\Delta t}$ require a massive reduction of the time step size. As the final steady state is approached, the time step size is increased until it reaches Δt_{\max} for all $\varepsilon_l^{s,\Delta t}$. As expected, smaller tolerances result in smaller values of the time step size Δt_n . Time step repetitions are only performed at the beginning of the buckling event, where some steps required one or two repetitions. All other portions of the simulation get along without time step repetitions.

The influence of the spatial discretization on the evolution of the time step size is studied as well. Figure 4.22 reports the evolution of Δt_n over t for the setting $\varepsilon_l^{s,\Delta t} = 5 \cdot 10^{-4}$ and $\varepsilon_l^{f,\Delta t} = 10^{-3}$ for the coarse and the fine mesh. Both curves deviate only very little from each other. Due to the different spatial refinements, both meshes results in different buckling patterns, while the finer mesh includes more localized buckling phenomena. Thus, more localized changes of dynamic behavior occur during the simulation resulting in slightly smaller time step sizes during the phase of increasing time step sizes $t \in [10, 25]$. Besides these local phenomena, the spatial discretization does not influence the adaptive time stepping as expected.

The energy production per time step $\Delta E_\Gamma^{n \rightarrow n+1}$ as discussed in Section 3.2.4 is studied as well. To put it into relation to the overall energy of the system, the kinetic energy E_{kin}^f of the fluid field is computed as well. Since energies of the solid are comparatively small, they are not included into the discussion and the fluid's kinetic energy is used as an indicator for the energy in the entire system. Figure 4.23 shows the kinetic energy E_{kin}^f , the energy production $\Delta E_\Gamma^{n \rightarrow n+1}$ per time step as well as its accumulation E_Γ^{n+1} . The fluid kinetic energy E_{kin}^f is governed by the prescribed inflow velocity and, thus, shows a huge increase when the inflow velocity is raised at $t = 5$. Oscillations occur during the buckling phase, but finally a constant level of kinetic energy is reached when a stationary state is approached. Looking at the interface, energy production per time step $\Delta E_\Gamma^{n \rightarrow n+1}$ as well as its accumulation E_Γ is by far smaller, namely by three to five orders of magnitude. The main accumulation of artificial interface energy happens during the buckling event. Before and after the buckling, interface energy production per time step is around 10^{-6} . This along with the previous analysis by MAYR *et al.* [163] confirm numerically that the amount of energy production due to the mismatch $t_m^f \neq t_m^s$ is negligible,

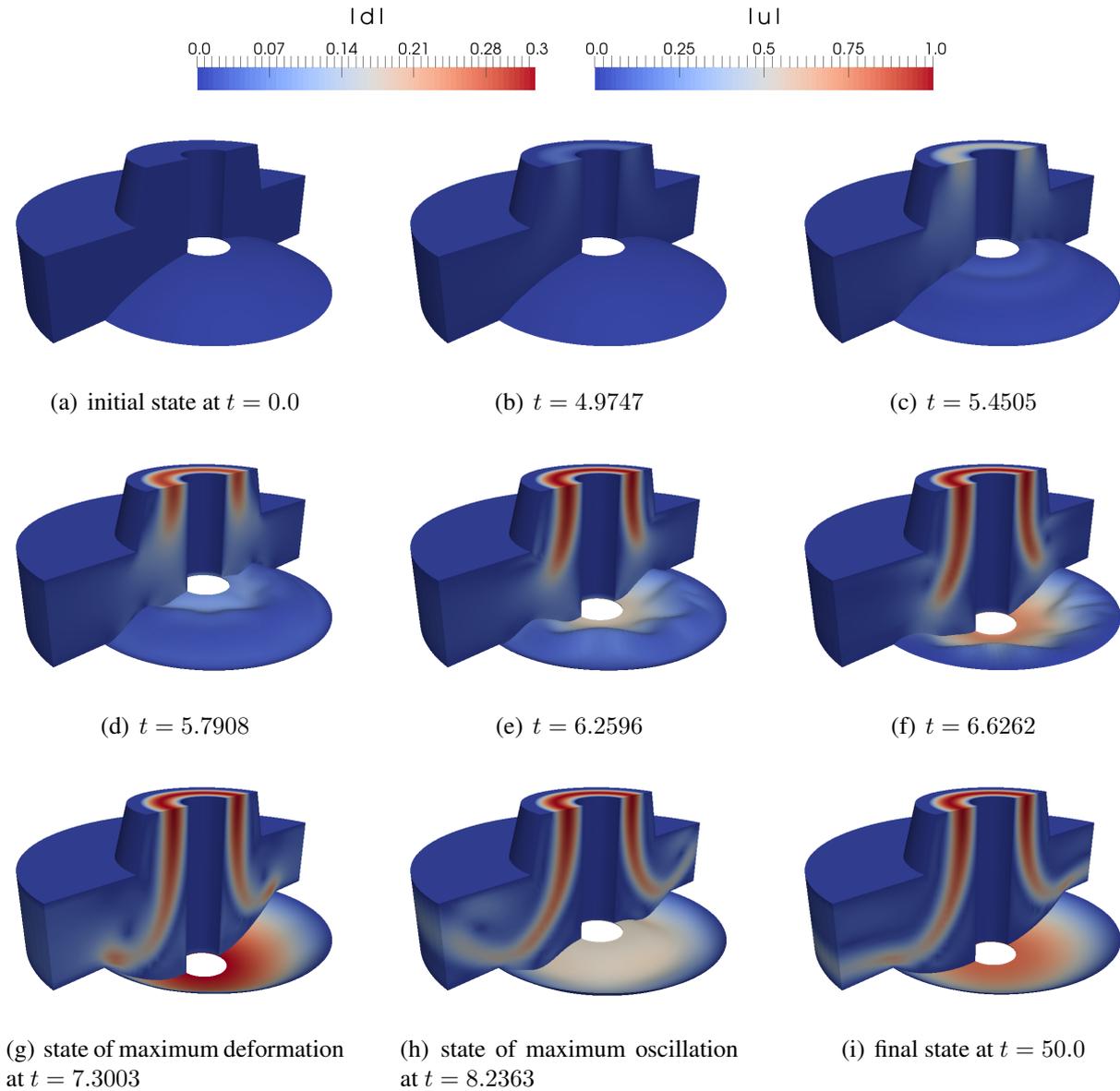


Figure 4.19: Snapshots of the solution of a gasket at selected time steps using the *fine* mesh — The fluid field is colored according to the magnitude of its velocity field denoted by $|u|$ while the structure field's color is encoded with the displacement field magnitude denoted by $|d|$. Color scales are calibrated at maximum deformation at $t = 7.30$. Simulation has been performed with $\varepsilon_l^{S,\Delta t} = 10^{-3}$ and $\varepsilon_l^{F,\Delta t} = 10^{-3}$.

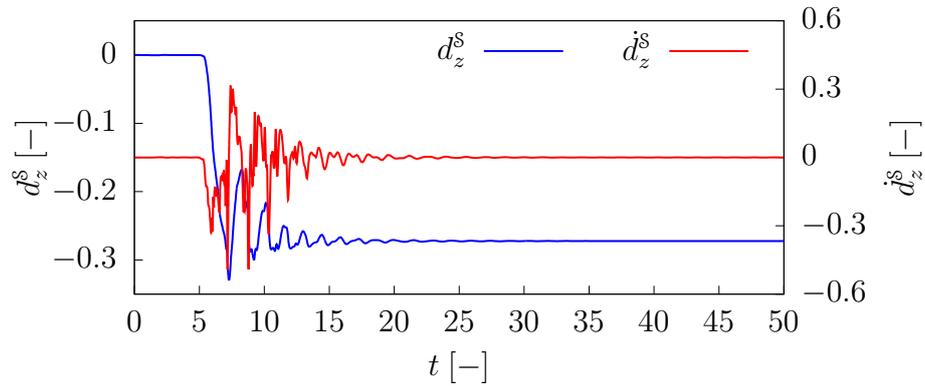
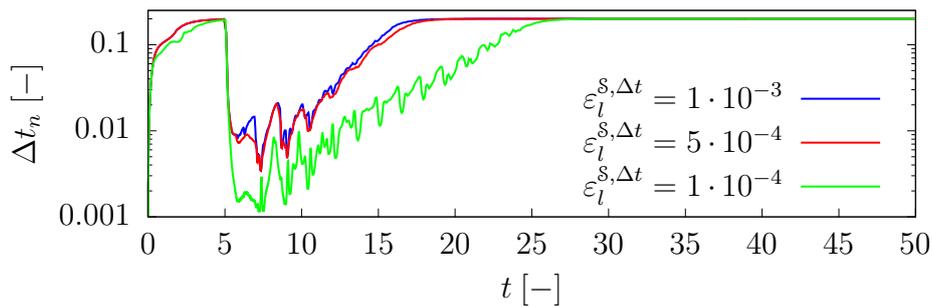
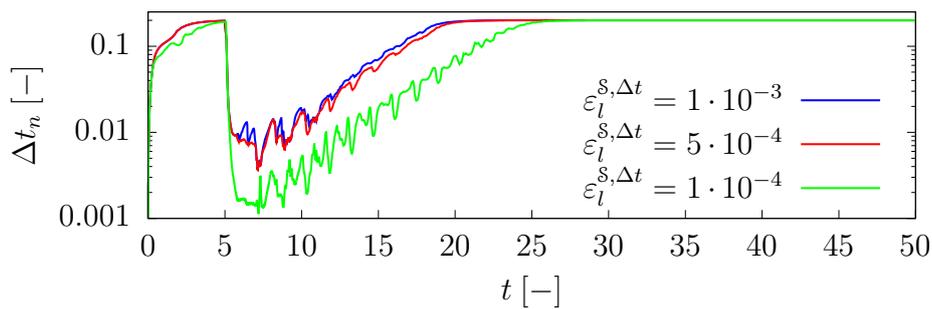


Figure 4.20: Displacement and velocity in the gasket-like example — Starting from resting conditions, large amplitudes in displacement and velocity occur during as soon as the buckling starts. Afterwards, the solid settles in a stable equilibrium configuration. Results are obtained with $\varepsilon_l^{s,\Delta t} = 5 \cdot 10^{-4}$ and $\varepsilon_l^{f,\Delta t} = 10^{-3}$.



(a) Coarse mesh



(b) Fine mesh

Figure 4.21: Evolution of the time step size Δt_n for the gasket-like example — The time step size is increased in the initial phase of the simulation. When the buckling starts, a massive reduction of Δt_n is performed, while it can increase again towards the end of the simulation. Behavior for the coarse mesh (*top*) and the fine mesh (*bottom*) are similar.

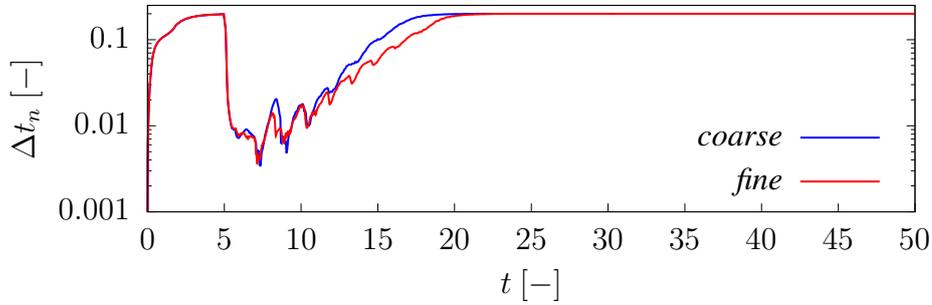


Figure 4.22: Influence of the mesh on the evolution of the time step size Δt_n for the gasket-like example — The time step size evolves very similarly for both meshes. The finer mesh sometimes requires slightly smaller Δt_n due to localized dynamic effects that come along with the better resolved buckling patterns.

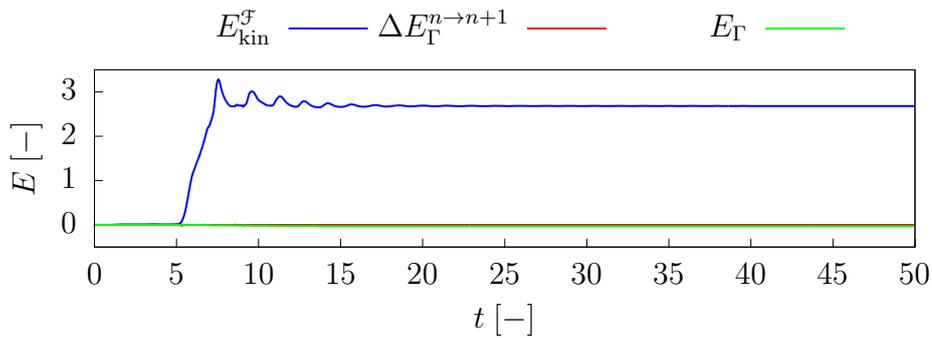


Figure 4.23: Energies in gasket-like example — The kinetic energy $E_{\text{kin}}^{\mathcal{F}}$ of the fluid field largely increases with the increase in inflow velocity at $t = 5$ and ranges in the order of magnitude of 10^0 . At the interface, energy production per time step $\Delta E_{\Gamma}^{n \rightarrow n+1}$ as well as its accumulation E_{Γ} are in the range of 10^{-5} and 10^{-3} and, thus, considered negligible.

especially in practical applications, where energy also heavily depends on viscous effects of the fluid as well as on the dissipation of the numerical scheme at hand.

4.4 Summary

The goal of controlling the temporal discretization error has been addressed in this chapter. After a brief introduction to some fundamental terminology of ODEs, a well-established adaptive time stepping scheme based on *a posteriori* error estimation has been reproduced from literature. Several approaches to practical error estimation have been reviewed.

By combining error estimation in the solid and the fluid field and additionally taking into account the fluid-structure interface, a novel adaptive time stepping scheme for monolithic FSI solvers has been proposed. Guidelines for the choice of all algorithmic parameters have been discussed. Several numerical examples have been used to demonstrate its properties, in particular the ability to guarantee a user-given level of accuracy while limiting the computational cost to the amount that is necessary. Remarkable savings compared to the case of a non-adaptive time stepping scheme could be achieved. The combination of guaranteed accuracy and computational efficiency is considered as highly relevant for practical applications.

5 Solution Methods for the Monolithic System of Equations

The most crucial part of applying monolithic schemes to FSI problems is the solution process. It is common to address the nonlinearity with a `NEWTON` scheme, while the linear systems (3.46) or (3.53) are usually solved with a preconditioned `KRYLOV` solver [89, 112, 142, 163, 227]. Due to the non-symmetry of the system of linear equations, usually the *Generalized Minimal Residual (GMRES) method* by SAAD and SCHULTZ [208] is applied as iterative linear solver. The most difficult part is defining a good preconditioner for GMRES. A variety of approaches has been reported in literature. GEE *et al.* [89] proposed very powerful physics-based block preconditioners based on *algebraic multigrid (AMG) methods*, which are briefly summarized in Sections 5.1.2 and 5.1.3. A re-implementation of the ideas by GEE *et al.* [89] with extensions to monolithic coupling of an arbitrary number of fields in a modular software framework within the in-house code `Baci` has been done by VERDUGO and WALL [240]. HEIL [112] and HEIL *et al.* [113] use block-triangular approximations to the full Jacobian as preconditioner. A preconditioner based on pseudo-solid mesh updates is proposed and analyzed by MUDDLE *et al.* [172]. Several preconditioner designs are briefly sketched by TEZDUYAR and SATHE [227] in the context of space-time finite elements, among them the block-iterative coupling used by GEE *et al.* [89], cf. Section 5.1.2. By extending the work of CROSETTO *et al.* [40], a block preconditioner for the factorized and statically condensed FSI matrix with a `SIMPLE` preconditioner for the fluid subproblem has been proposed recently by DEPARIS *et al.* [53], see also [83].

To ease the discussion of all preconditioning techniques, the linear systems (3.46) and (3.53) for the fluid-handled and structure-handled case are transformed to the general form

$$\begin{bmatrix} \mathcal{S} & & \mathcal{S}^{\mathcal{S}\mathcal{F}} \\ & \mathcal{A} & \mathcal{A}^{\mathcal{G}\mathcal{F}} \\ \mathcal{F}^{\mathcal{F}\mathcal{S}} & \mathcal{F}^{\mathcal{F}\mathcal{G}} & \mathcal{F} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^{\mathcal{S}} \\ \Delta \mathbf{x}^{\mathcal{G}} \\ \Delta \mathbf{x}^{\mathcal{F}} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}^{\mathcal{S}} \\ \mathbf{r}^{\mathcal{G}} \\ \mathbf{r}^{\mathcal{F}} \end{bmatrix} \quad (5.1)$$

by field-wise grouping of unknowns and subsequent rearrangement, such that an arrow-shaped block matrix arises. To ease notation, the distinction of interior and interface degrees of freedom as well as the choice of master and slave have been dropped without loss of generality. Additionally, time step and iteration indices are omitted to simplify notation. The superscripts at the off-diagonal blocks indicate coupling between two fields.

Three FSI-specific preconditioners are discussed in this chapter. At first, the two existing approaches by GEE *et al.* [89] are reviewed briefly in Section 5.1, namely a block-iterative coupling with AMG-based approximations of the block inverses and a fully coupled AMG preconditioner, that incorporates the interface coupling on the coarse levels of a multigrid hierarchy of the coupled problem. Additionally, some extensions of the fully coupled AMG preconditioner, namely dealing with different numbers of levels in each field as well as a smoother based on

a SCHUR complement approximation, are introduced. Both approaches make use of the separation of physical fields and are commonly referred to as *physics-based block preconditioners*. They are very powerful, but exhibit one major drawback, namely the accumulation of error at the fluid-structure interface. This is addressed by proposing a novel *hybrid additive/multiplicative SCHWARZ preconditioner* in Section 5.2. It overcomes the drawbacks of classical physics-based block preconditioning approaches by hybridizing the existing FSI preconditioners of multiplicative SCHWARZ type with an additional additive SCHWARZ preconditioner. It is based on an overlapping domain decomposition with subdomains spanning across the interface, which are then treated with special subdomain solvers to reduce error accumulation at the interface. This chapter concludes with some remarks on practical aspects of the involved linear and nonlinear solvers in a monolithic setting, cf. Section 5.3. A detailed comparison and performance study of all preconditioners is performed later using numerical examples, cf. Chapter 6. Finally, an intermediate summary is given in Section 5.4.

5.1 Physics-Based Multilevel Block Preconditioning for FSI

A variety of preconditioners for block matrices as given in (5.1) is available in literature. Common to all these approaches is the fact that they exploit the block structure of the system matrix. The block structure usually corresponds to the grouping of unknowns of different physical fields, while coupling between the fields is reflected by off-diagonal blocks. Thus, such preconditioners are often referred to as *physics-based block preconditioners*. Some approaches have already been listed at the beginning of this chapter.

After a brief introduction to algebraic multigrid methods, that are core parts of the presented preconditioners, two multi-level block preconditioners by GEE *et al.* [89] are summarized briefly, namely a block-iterative coupling with approximate block inverses in Section 5.1.2 and a fully coupled algebraic multigrid preconditioner in Section 5.1.3, .

5.1.1 Algebraic Multigrid Methods in a Nutshell

One of the most powerful methods for the preconditioning of systems of linear equations that arise from finite element discretizations are *algebraic multigrid (AMG) methods*. Only a very brief summary is given here. For an intuitive presentation of multigrid methods in general and AMG methods in particular see the introductory book by BRIGGS *et al.* [27]. More detailed derivations as well as mathematical foundations are given in the monographs by HACKBUSCH [108, 109] or WESSELING [247] for example. The relation between multigrid methods and *domain decomposition techniques* is discussed by SMITH *et al.* [220] for example. A very recent literature survey on multigrid methods is given in the thesis by WIESNER [249].

The idea behind multigrid methods is to introduce coarse representations of the problem at hand and use solutions of these coarse problems to accelerate the solution process of the original fine level problem. This strategy heavily relies on the fact, that high-frequency components of the error can effectively be reduced by cheap and simple relaxation-type smoothers, e.g. JACOBI or GAUSS–SEIDEL smoothers, while their effect on low-frequency modes is often very poor. By transferring the low-frequency modes to a coarser representation of the problem, they appear as

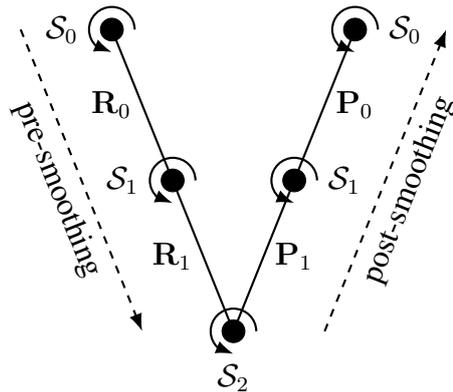


Figure 5.1: Sketch of a 3-level multigrid V-cycle — On each level ℓ , a smoother \mathcal{S}_ℓ is applied to smooth the different components of the error on different levels. On fine and medium levels, one usually applies cheap smoothers, while a direct solve is performed on the coarsest level. Levels are visited in the order indicated by the dashed arrows. Quantities are transferred between the levels by restriction and prolongation operators \mathbf{R}_ℓ and \mathbf{P}_ℓ , respectively.

high-frequency modes relative to the coarser 'grid' and, thus, can again be effectively tackled by relaxation-type smoothers. This process is done recursively until a small enough coarse system of equations is reached that can be treated with a direct solver efficiently. Each coarse representation, referred to as *level* in the context of AMG methods, provides a correction to the solution of the next finer representation. The idea behind multigrid methods can be summarized as applying cheap smoothing techniques on differently coarse representations of the original problem at hand to reduce the different components of the error effectively on different levels of the multigrid hierarchy.

A variety of schemes, in which the levels are visited, is available in literature. In this thesis, *V-cycles* are used exclusively. Figure 5.1 exemplarily shows a 3-level multigrid V-cycle. During the *pre-smoothing* process, the algorithm starts at the fine level and works its way down to the coarsest level, while the *post-smoothing* process proceeds in the opposite direction. The transfer of quantities between the levels is managed by the *level transfer operators*, namely the *restriction operators* \mathbf{R}_ℓ for the transfer from a fine level ℓ to a coarse level $\ell + 1$ and the *prolongation operators* \mathbf{P}_ℓ for the transfer from any coarse level $\ell + 1$ to a fine level ℓ , respectively. On each level, smoothing techniques are applied, that can be completely specified by the user. On the coarsest level, one usually employs a direct solver. Depending on the physical field at hand, specific choices of transfer operators and smoothers are beneficial or even necessary.

Although AMG can be applied as a stand-alone solver for systems of linear equations, it is usually applied to precondition a KRYLOV solver. For elasticity problems, a very popular choice is the application of *smoothed aggregation (SA)* techniques as introduced by VANĚK *et al.* [234–236], BREZINA [26], or MANDEL *et al.* [160]. For convection dominated flow, an energy minimization approach by SALA *et al.* [210] is available which is also referred to as PETROV–GALERKIN *algebraic multigrid (PG-AMG)*. Multigrid transfer operators for nonsymmetric systems based on SCHUR complements and GALERKIN projections have been developed

by WIESNER *et al.* [250]. In this thesis, SA-AMG is used for solid and ALE, while PG-AMG is applied to the fluid's equations. Implementations utilize the ML package [90] of the Trilinos libraries [116].

5.1.2 A Block-Iterative Approach with an Internal Algebraic Multigrid Preconditioner

As given in the thesis by KÜTTLER [145] or by GEE *et al.* [89], a block version of the GAUSS–SEIDEL method, referred to as *block* GAUSS–SEIDEL (*BGS*), can be used as preconditioner for the monolithic system of equations (5.1). It can be achieved by dropping the upper-triangular coupling blocks in (5.1), yielding the *forward BGS* preconditioner

$$\mathcal{M}_{\text{BGS}}^{-1} = \begin{bmatrix} \mathcal{S} & & \\ \mathcal{F}^{\mathcal{FS}} & \mathcal{A} & \\ & \mathcal{F}^{\mathcal{FG}} & \mathcal{F} \end{bmatrix}^{-1}. \quad (5.2)$$

For an efficient application of $\mathcal{M}_{\text{BGS}}^{-1}$, the required block inverses \mathcal{S}^{-1} , \mathcal{A}^{-1} , and \mathcal{F}^{-1} are approximated by field-wise preconditioning operations $\mathcal{M}_{\mathcal{S}}^{-1}$, $\mathcal{M}_{\mathcal{G}}^{-1}$, and $\mathcal{M}_{\mathcal{F}}^{-1}$ based on field-wise AMG hierarchies. In these field-wise hierarchies, a variety of smoothers are available, among them relaxation-based smoothers like damped JACOBI or damped, possibly symmetric GAUSS–SEIDEL methods, CHEBYSHEV polynomials in case of symmetric matrices, or *incomplete LU (ILU) factorizations* with different fill levels, all implemented in Trilinos' `Ipack` package [209]. On the coarse grid, usually a direct solver is applied, most often from the *Unsymmetric MultiFrontal PACKage (UMFPACK)* [43]. Moreover, embedding of the BGS preconditioner into a damped RICHARDSON iteration is straightforward, cf. KÜTTLER [142], but is not utilized in this thesis. Rather an additional GMRES iteration is performed, which is much more effective than sweeping through the preconditioner twice per GMRES iteration.

This preconditioner is denoted by *BGS(AMG)*, since it uses a BGS method on the outside with embedded AMG preconditioners for each physical field. A sketch of the algorithm is given in Figure 5.2. The solid, ALE, and fluid multigrid V-cycles are denoted by \mathcal{S} , \mathcal{G} , and \mathcal{F} , respectively. Arrows indicate the coupling between the fields, which is realized via the BGS method and is incorporated on the fine level only.

Advantages of this approach are as follows: First, it is very easy to implement. The BGS method is very simple and just consists of a few lines of code, while all parts of the algorithm related to multigrid methods can be handed off to a black box multigrid software package. Such software packages provide a variety of pre-implemented and ready-to-use aggregation strategies and level smoothers. In `Baci`, Trilinos' ML package [87] is used for the FSI preconditioner, while a more recent implementation [240] uses the `MueLu` package [188]. Second, the setup cost is comparatively small, since the BGS does not involve any setup at all and the AMG setup done by ML is highly optimized. Third, by using AMG as a black box, the necessary insight is less than when the multigrid related algorithms have to be implemented by the user himself. Fourth, since the field-wise multigrid hierarchies do not interact with each other directly, there is no restriction on the number of levels in each of them. This is beneficial in cases, where for example the solid discretization is much smaller than the fluid and ALE discretization, which

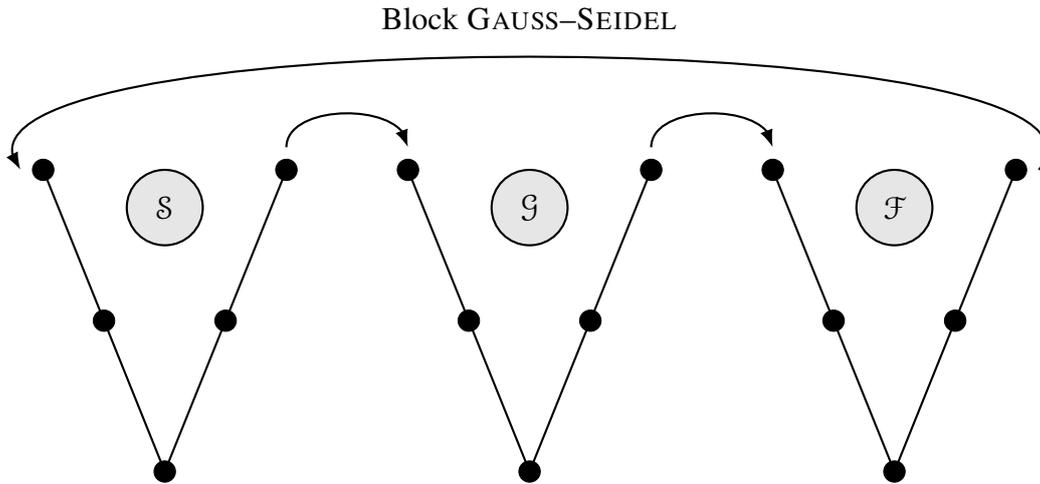


Figure 5.2: Block GAUSS–SEIDEL preconditioner with AMG-based approximate block inverses — A block GAUSS–SEIDEL method is used to transfer information among the fields, while the action of field-wise block inverses is approximated with field-individual multigrid schemes. Data is exchanged between the fields only on the fine level $\ell = 0$. Thus, the numbers of levels $n^{S,\ell}$, $n^{F,\ell}$, and $n^{G,\ell}$ of the solid, fluid, and ALE hierarchies do not have to match. This sketch is inspired by the presentation in [142].

could lead to a larger number of levels $n^{F,\ell}$ and $n^{G,\ell}$ in fluid and ALE hierarchies, respectively, than in the solid hierarchy with $n^{S,\ell}$ levels. Fifth, each hierarchy is allowed to perform a dynamic rebalancing if the coarse level systems are too small to be solved on all involved processors. Such a rebalancing can be crucial to obtain optimal multigrid scalability and performance.

However, this approach also exhibits some drawbacks. The main issue is the fact that the coupling information is not included on the coarse levels of the multigrid hierarchies. By exchanging coupling information among the fields only on the fine level, coupling is addressed by means of the BGS method only and cannot benefit from the powerful multigrid performance. Following the experiment of thoughts as given by GEE *et al.* [89], it can be argued, that even if the block inverses can be computed exactly, the error after one application of the preconditioner concentrates at the fluid-structure interface. To reduce those errors, quite some additional KRYLOV iterations need to be performed. This is expensive, especially with the notion, that one needs to deal with the full system just to reduce the error in a small, but important portion of it.

5.1.3 A Fully Coupled Algebraic Multigrid Preconditioner

To cure the main drawback of the block-iterative coupling, namely the lack of interface coupling on the coarse levels $\ell > 0$ of the multigrid hierarchies, GEE *et al.* [89] inverted the nesting of BGS and field-specific AMG hierarchies to construct an AMG hierarchy for the coupled FSI problem. Its key feature is the inclusion of the FSI coupling terms on the coarse levels. Hence, it is referred to as *fully coupled algebraic multigrid preconditioner*.

The idea of the fully coupled AMG preconditioner for FSI as given by GEE *et al.* [89] or in the thesis by KÜTTNER [142] is illustrated in Figure 5.3. A fully coupled multigrid hierarchy

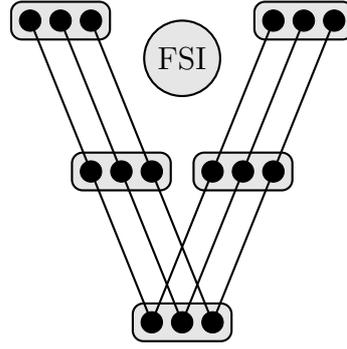


Figure 5.3: Fully coupled AMG preconditioner for FSI — A fully coupled multigrid hierarchy is constructed, that includes FSI coupling terms on the coarse levels as well. This strongly enhances the preconditioning effect, since interface-related errors can be tackled by the coarse grid correction effectively. A BGS method or an approximate SCHUR complement approach are applied as level smoothers. This sketch is inspired by the presentation in [142].

is constructed algebraically, such that the FSI coupling terms are included on the coarse levels as well. FSI-specific smoothers are applied on each level of the fully coupled AMG hierarchy. This strongly enhances the preconditioning effect, since interface-related errors can be tackled by the coarse grid correction effectively. Of course, suitable level smoothers need to be defined, that are able to handle the block system on each level.

Setup of the Fully Coupled AMG Hierarchy

Assuming the existence of field-specific restriction operators $\mathbf{R}_\ell^{\mathcal{S}}$, $\mathbf{R}_\ell^{\mathcal{G}}$, and $\mathbf{R}_\ell^{\mathcal{F}}$ as well as prolongation operators $\mathbf{P}_\ell^{\mathcal{S}}$, $\mathbf{P}_\ell^{\mathcal{G}}$, and $\mathbf{P}_\ell^{\mathcal{F}}$ associated with the level transfer between levels ℓ and $\ell + 1$ for solid, ALE, and fluid field, respectively, a representation of the monolithic system of linear equations (3.39) on level $\ell \in [0, n^\ell - 1]$ can be given as

$$\mathcal{J}_\ell \Delta \mathbf{x}_\ell = -\mathbf{r}_\ell, \quad (5.3)$$

where the multigrid hierarchy is assumed to consist of n^ℓ levels. To ease notation, time step and iteration indices have been dropped. On level $\ell + 1$ with $\ell \in [0, n^\ell - 1]$, the coarsened Jacobian in (5.4) reads

$$\begin{aligned} \mathcal{J}_{\ell+1} &= \begin{bmatrix} \mathbf{R}_\ell^{\mathcal{S}} & & \\ & \mathbf{R}_\ell^{\mathcal{G}} & \\ & & \mathbf{R}_\ell^{\mathcal{F}} \end{bmatrix} \begin{bmatrix} \mathcal{S} & & \mathcal{S}^{\mathcal{S}\mathcal{F}} \\ & \mathcal{A} & \mathcal{A}^{\mathcal{G}\mathcal{F}} \\ \mathcal{F}^{\mathcal{F}\mathcal{S}} & \mathcal{F}^{\mathcal{F}\mathcal{G}} & \mathcal{F} \end{bmatrix}_\ell \begin{bmatrix} \mathbf{P}_\ell^{\mathcal{S}} & & \\ & \mathbf{P}_\ell^{\mathcal{G}} & \\ & & \mathbf{P}_\ell^{\mathcal{F}} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_\ell^{\mathcal{S}} \mathcal{S}_\ell \mathbf{P}_\ell^{\mathcal{S}} & & \mathbf{R}_\ell^{\mathcal{S}} \mathcal{S}_\ell^{\mathcal{S}\mathcal{F}} \mathbf{P}_\ell^{\mathcal{F}} \\ & \mathbf{R}_\ell^{\mathcal{G}} \mathcal{A}_\ell \mathbf{P}_\ell^{\mathcal{G}} & \mathbf{R}_\ell^{\mathcal{G}} \mathcal{A}_\ell^{\mathcal{G}\mathcal{F}} \mathbf{P}_\ell^{\mathcal{F}} \\ \mathbf{R}_\ell^{\mathcal{F}} \mathcal{F}_\ell^{\mathcal{F}\mathcal{S}} \mathbf{P}_\ell^{\mathcal{S}} & \mathbf{R}_\ell^{\mathcal{F}} \mathcal{F}_\ell^{\mathcal{F}\mathcal{G}} \mathbf{P}_\ell^{\mathcal{G}} & \mathbf{R}_\ell^{\mathcal{F}} \mathcal{F}_\ell \mathbf{P}_\ell^{\mathcal{F}} \end{bmatrix}, \end{aligned} \quad (5.4a)$$

while the residual vector $\mathbf{r}_{\ell+1}$ is computed as restriction of the fine level residual vector, reading

$$\mathbf{r}_{\ell+1} = \begin{bmatrix} \mathbf{R}_\ell^{\mathcal{S}} & & \\ & \mathbf{R}_\ell^{\mathcal{G}} & \\ & & \mathbf{R}_\ell^{\mathcal{F}} \end{bmatrix} \begin{bmatrix} \mathbf{r}_\ell^{\mathcal{S}} \\ \mathbf{r}_\ell^{\mathcal{G}} \\ \mathbf{r}_\ell^{\mathcal{F}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_\ell^{\mathcal{S}} \mathbf{r}_\ell^{\mathcal{S}} \\ \mathbf{R}_\ell^{\mathcal{G}} \mathbf{r}_\ell^{\mathcal{G}} \\ \mathbf{R}_\ell^{\mathcal{F}} \mathbf{r}_\ell^{\mathcal{F}} \end{bmatrix}. \quad (5.4b)$$

It is stressed that all building blocks for the coarse-level FSI system of equations are readily available from the single-field multigrid hierarchies. The only tasks to be performed by the user's implementation itself is the construction of the off-diagonal coupling blocks in (5.4a) as well as defining a smoother on each level, that is capable of dealing with the block structure of (5.4).

Note that (5.4) requires the number of levels in solid, fluid, and ALE hierarchies to be equal, i.e. $n^{\mathcal{S},\ell} = n^{\mathcal{F},\ell} = n^{\mathcal{G},\ell} = n^\ell$. Especially in cases of largely differing sizes of solid and fluid discretizations, this seems to be a limiting factor, since the larger field cannot be coarsened as much as necessary. However, this can be easily cured by revisiting the construction of the off-diagonal coupling blocks. For levels $\ell < \min\{n^{\mathcal{S},\ell}, n^{\mathcal{F},\ell}, n^{\mathcal{G},\ell}\}$, coarsening is performed as given in (5.4). As any field reaches its coarsest level prior to the other fields, this field's matrix is not coarsened further. This is achieved by setting restriction and prolongation operators of this field to be the identity \mathbf{I}_ℓ . Only the other fields are coarsened further. Exemplifying the case $n^{\mathcal{S},\ell} < n^{\mathcal{F},\ell} = n^{\mathcal{G},\ell}$, coarsening is performed as given by (5.4) until $\ell = n^{\mathcal{S},\ell} - 1$. Then, further coarsening follows the strategy

$$\begin{aligned} \mathcal{J}_{\ell+1} &= \begin{bmatrix} \mathbf{I}_\ell & & \\ & \mathbf{R}_\ell^{\mathcal{G}} & \\ & & \mathbf{R}_\ell^{\mathcal{F}} \end{bmatrix} \begin{bmatrix} \mathcal{S}_{\ell^{\mathcal{S}}} & & \mathcal{S}_{\ell^{\mathcal{S}}}^{\mathcal{S}\mathcal{F}} \\ & \mathcal{A}_\ell & \mathcal{A}_\ell^{\mathcal{G}\mathcal{F}} \\ \mathcal{F}_\ell^{\mathcal{F}\mathcal{S}} & \mathcal{F}_\ell^{\mathcal{F}\mathcal{G}} & \mathcal{F}_\ell \end{bmatrix} \begin{bmatrix} \mathbf{I}_\ell & & \\ & \mathbf{P}_\ell^{\mathcal{G}} & \\ & & \mathbf{P}_\ell^{\mathcal{F}} \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{S}_{\ell^{\mathcal{S}}} & & \mathcal{S}_{\ell^{\mathcal{S}}}^{\mathcal{S}\mathcal{F}} \mathbf{P}_\ell^{\mathcal{F}} \\ & \mathbf{R}_\ell^{\mathcal{G}} \mathcal{A}_\ell \mathbf{P}_\ell^{\mathcal{G}} & \mathbf{R}_\ell^{\mathcal{G}} \mathcal{A}_\ell^{\mathcal{G}\mathcal{F}} \mathbf{P}_\ell^{\mathcal{F}} \\ \mathbf{R}_\ell^{\mathcal{F}} \mathcal{F}_\ell^{\mathcal{F}\mathcal{S}} & \mathbf{R}_\ell^{\mathcal{F}} \mathcal{F}_\ell^{\mathcal{F}\mathcal{G}} \mathbf{P}_\ell^{\mathcal{G}} & \mathbf{R}_\ell^{\mathcal{F}} \mathcal{F}_\ell \mathbf{P}_\ell^{\mathcal{F}} \end{bmatrix} \end{aligned} \quad (5.5a)$$

and

$$\mathbf{r}_{\ell+1} = \begin{bmatrix} \mathbf{I} & & \\ & \mathbf{R}_\ell^{\mathcal{G}} & \\ & & \mathbf{R}_\ell^{\mathcal{F}} \end{bmatrix} \begin{bmatrix} \mathbf{r}_{\ell^{\mathcal{S}}}^{\mathcal{S}} \\ \mathbf{r}_\ell^{\mathcal{G}} \\ \mathbf{r}_\ell^{\mathcal{F}} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{\ell^{\mathcal{S}}}^{\mathcal{S}} \\ \mathbf{R}_\ell^{\mathcal{G}} \mathbf{r}_\ell^{\mathcal{G}} \\ \mathbf{R}_\ell^{\mathcal{F}} \mathbf{r}_\ell^{\mathcal{F}} \end{bmatrix}. \quad (5.5b)$$

where the level index $\ell^{\mathcal{S}}$ for the solid field is kept constant at $\ell^{\mathcal{S}} = n^{\mathcal{S},\ell} - 1$ while fluid and ALE level indices are progressively incremented with ongoing coarsening until $\ell = n^{\mathcal{F},\ell} - 1 = n^{\mathcal{G},\ell} - 1$. Note that the solid block $\mathcal{S}_{\ell^{\mathcal{S}}}$ in (5.5) is not affected by further coarsening of fluid and ALE fields. Figure 5.4 illustrates this strategy using graphical representations of V-cycles. Of course, a similar strategy can be applied for the cases with more levels in the solid hierarchy than the fluid and ALE hierarchies or even in cases where all three hierarchies have different numbers of levels.

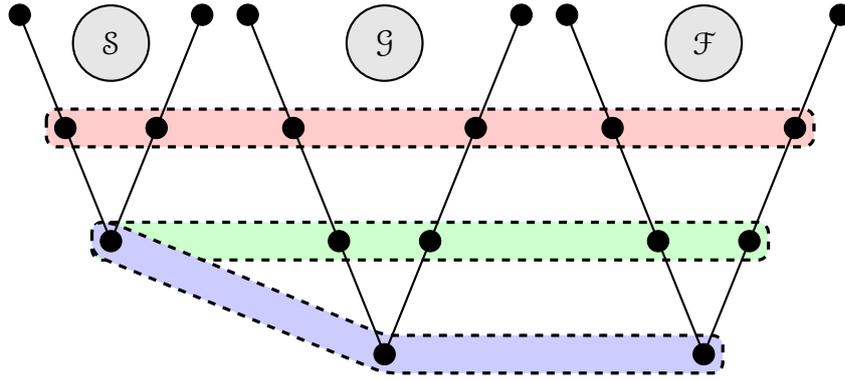


Figure 5.4: Fully coupled multigrid hierarchy with different numbers of levels per field exemplifying $n^{S,\ell} < n^{F,\ell} = n^{G,\ell}$ — For levels $\ell < \min\{n^{S,\ell}, n^{F,\ell}, n^{G,\ell}\} - 1$ (red and green), the coarse problem is constructed via (5.4). After the solid reached its coarsest level, further coarsening happens only in fluid and ALE (purple), while the off-diagonal coupling blocks need to be treated properly, cf. (5.5).

Smoothers for the Fully Coupled AMG Hierarchy

On each level ℓ , a smoother needs to be defined. On the one hand, the BGS method (5.2) can be applied as level smoother, while the actions of the block inverses are approximated with the same field-specific one-level preconditioners that already have been used as level smoothers in the internal AMG hierarchies of the BGS(AMG) approach. On the coarse level, usually a BGS(LU), i.e. a block GAUSS–SEIDEL method with exact block inverses, is preferred over a direct solve on the fully coupled coarse level matrix. The latter would require a transformation of the block sparse matrix storage format into a single sparse matrix format, which is computationally expensive due to a change of the matrix type.

On the other hand, a SCHUR complement approximation can be used as a level smoother. Thereby, the fluid SCHUR complement is approximated via

$$\hat{\mathbf{S}}^F = \mathcal{F} - \mathcal{F}^{\mathcal{FS}} \mathbf{D}_S^{-1} \mathcal{S}^{\mathcal{SF}} - \mathcal{F}^{\mathcal{FG}} \mathcal{D}_G^{-1} \mathcal{A}^{\mathcal{GF}} \quad (5.6)$$

with $\mathbf{D}_S = \text{diag}(\mathcal{S})$ and $\mathcal{D}_G = \text{diag}(\mathcal{A})$ being pure diagonal matrices, that contain the diagonal entries $[\mathcal{S}_{ii}]$ and $[\mathcal{A}_{ii}]$ of the solid and ALE matrix \mathcal{S} and \mathcal{A} , respectively. Application of the SCHUR complement approach on any level $\ell \in [0, n^\ell - 1]$ is done as

$$\begin{aligned} \Delta \mathbf{x}_\ell^F &= - \left(\hat{\mathbf{S}}^F \right)^{-1} \mathbf{r}_\ell^F, \\ \Delta \mathbf{x}_\ell^S &= - \mathcal{S}^{-1} \left(\mathbf{r}_\ell^S + \mathcal{S}_\ell^{\mathcal{SF}} \Delta \mathbf{x}_\ell^F \right), \\ \Delta \mathbf{x}_\ell^G &= - \mathcal{A}^{-1} \left(\mathbf{r}_\ell^G + \mathcal{A}_\ell^{\mathcal{GF}} \Delta \mathbf{x}_\ell^F \right). \end{aligned} \quad (5.7)$$

In practical applications, the inverses in (5.7) are never formed explicitly. As the approximate SCHUR complement is used as preconditioner for an outer KRYLOV method, it is sufficient to approximate the action of the inverses in (5.7) on a vector by any, often cheap, iterative method.

Independently, LANGER and YANG [149] proposed a very similar SCHUR complement preconditioner, but they replace \mathbf{D}_S and \mathcal{D}_G by point-wise block diagonal matrices. They used it as

stand-alone one-level preconditioner, yielding mesh size dependency of the number of GMRES iterations, which had been cured by applying the fully coupled AMG method described in this section. Additionally, they briefly touched the cases where SCHUR complement approximations for the solid and ALE block are constructed, but considered them as not worth to investigate, since their construction is not cheaper than (5.6). Their numerical results show very good performance in terms of GMRES iteration counts. However, parallel performance has not been assessed and a detailed comparison to other state of the art FSI preconditioning methods is missing in their report.

Both the BGS and the approximate SCHUR complement smoothers can be embedded into stationary RICHARDSON iterations. Most often, only a single iteration is performed with a damping factor ω_ℓ . Multiple iterations need to be performed only in very demanding problems. Embedding of the fully coupled AMG preconditioner in a RICHARDSON iteration around the coupled V-cycle is possible, but not applied for similar reason as in the case of BGS(AMG).

Depending on the actual choice of the level smoother, i.e. BGS or approximate SCHUR complement, this preconditioner is referred to as AMG(BGS) or AMG(Schur), respectively. This notation indicates, that the BGS or approximate SCHUR complement are embedded into the fully coupled AMG method. This idea will generally be referred to as AMG(BGS/Schur), when the distinction of the actual level smoother is not important.

Revisiting the experiment of thoughts used to motivate the AMG(BGS/Schur) preconditioner, a certain amount of improvement can be expected, since the multigrid coarse level corrections reflect the interface coupling. However, the basic issue of a block preconditioner that relies on the physics-based block structure of the matrix is still present. Thinking in terms of AMG(BGS), the fine and coarse level coupling is still only addressed by means of the BGS method, even if the block inverses inside the BGS method are computed exactly. Hence, a concentration of error at the fluid-structure interface is still expected, even if it is less pronounced as for the BGS(AMG) approach.

5.2 A Novel Hybrid Additive/Multiplicative SCHWARZ Preconditioner for Surface-Coupled Problems

Both preconditioning approaches presented in Section 5.1 exploit the block structure of the FSI system matrix that is related to the separation of physical fields by the fluid-structure interface. A commonality of all physics-based block preconditioners is the concentration of error at the fluid-structure interface as already indicated in the experiment of thoughts at the end of Section 5.1.2. The present section aims at developing a preconditioner that in particular addresses this issue. This can be achieved by combining the existing physics-based block preconditioners denoted by \mathcal{M}_{MS}^{-1} with an additional preconditioner \mathcal{M}_{AS}^{-1} that is based on a purposely 'non-physics-based' overlapping domain decomposition. By neglecting the location of the interface when generating the domain decomposition, the resulting subdomains span across the fluid-structure interface on purpose. By using high-quality solves, i.e. direct or close-to-direct solves, on the patches across the interface, the accumulated error stemming from the physics-based block preconditioner can be reduced effectively. Of course, the subdomain solves have to be insensitive to the mixed-physics block matrices that are encountered on subdomains spanning

across the FSI interface. Motivations of the notations \mathcal{M}_{MS}^{-1} and \mathcal{M}_{AS}^{-1} will be given below. A preliminary implementation of such an approach has been done by NOLL [177].

After a brief introduction to overlapping domain decomposition and SCHWARZ methods in Section 5.2.1, requirements and setup of the desired domain decomposition are detailed in Section 5.2.2. Technically, combining the preconditioners is realized by embedding them into a RICHARDSON iteration, which will be described in Section 5.2.3. A demonstrative example is then given in Section 5.2.4, while performance evaluations on a larger scale will be carried out in Section 6.1.2.

5.2.1 Overlapping Domain Decomposition and SCHWARZ Methods

Only a very brief introduction of *domain decomposition (DD)* methods and their relation to SCHWARZ methods is given, which is mainly limited to introduce the usual wording related to such methods. For comprehensive and detailed introductions of DD and SCHWARZ methods it is referred to the monographs by QUARTERONI and VALLI [195], SMITH *et al.* [220], or TOSELLI and WIDLUND [228], for example.

In this thesis, only *overlapping DD* methods are used, while the case of *non-overlapping DD* methods is not treated at all. In overlapping DD methods, the entire computational domain Ω is decomposed into M overlapping subdomains Ω_m with $m = 0, \dots, M - 1$. Then, the problem is reformulated as a local DIRICHLET-type problem on each subdomain. Exchange of information among the subdomains happens via the overlap of the subdomains. In parallel computer architectures, subdomains Ω_m are often assigned to a processor m to allow for parallel execution and speed-up of the computation.

Two elementary methods, known as *additive SCHWARZ method* and *multiplicative SCHWARZ method*, will play an important role in defining the FSI preconditioners. Both are based on an overlapping DD. Starting from a matrix representation that groups unknowns according to subdomains, one ends up with an additive SCHWARZ method by dropping all off-diagonal blocks, which equals a block-JACOBI approach. Solutions on all subdomains can be computed simultaneously, since they do not depend on other subdomains. In opposite, multiplicative SCHWARZ methods are obtained by dropping only the upper-triangular off-diagonal blocks, yielding a block-GAUSS-SEIDEL approach. Solving for each subdomain needs to be done one after another, since the lower-triangular off-diagonal blocks couple the subdomains and, thus, require the solution in subdomain $m - 1$ to be known in order to solve on subdomain m . For further details the reader is referred to the literature listed above.

5.2.2 Requirements for the Domain Decomposition

To allow for hybridization of the existing physics-based block preconditioners with the additional preconditioner \mathcal{M}_{AS}^{-1} that specifically tackles error accumulation at the interface, the parallel distribution of the problem among all processors must satisfy certain requirements. They will be discussed briefly in the following.

A typical overlapping domain decomposition for purely physics-based block preconditioners is illustrated in Figure 5.5. The entire computational domain Ω is separated into a solid domain Ω^S and a fluid domain Ω^F by the fluid-structure interface Γ_{FSI} . To speed up computations

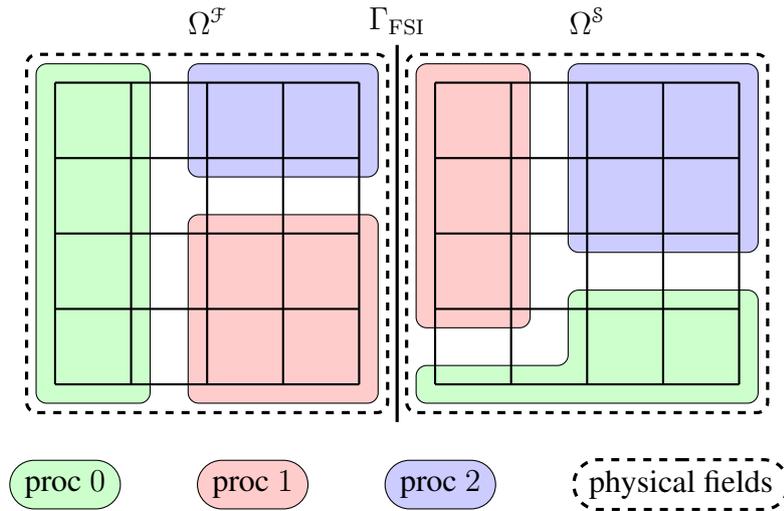


Figure 5.5: Physics-based overlapping domain decomposition of a FSI problem — At the fluid-structure interface Γ_{FSI} , the domain is partitioned into solid and fluid subdomains indicated by dashed lines. Each field can further be distributed among several processors by an overlapping domain decomposition indicated by the colored patches. Overlap of subdomains is not depicted for clarity of presentation.

on parallel hardware architectures, each physical field can be partitioned among n^{core} processors by an overlapping domain decomposition, cf. 'proc 0', 'proc 1', and 'proc 2' in Figure 5.5. For simplicity of illustration, coloring of the subdomains is done based on the 'interior' nodes of each subdomain, while the overlap is not visualized. By passing the solid and fluid domain separately to a partitioning tool like `Zoltan` [23], the partitioner is not aware of the fluid-structure interface. This results in domain decompositions of the solid and the fluid field, where subdomain boundaries at the interface inside the solid domain do not necessarily coincide with subdomain boundaries of the fluid field. Note that every processor handles a portion of each field, i.e. owns nodes of both solid and fluid subdomains.

Remark 5.2.1 *Since the fluid field is solved on a moving domain by means of an ALE description, the ALE field needs to be distributed among the processors as well. Usually, a domain decomposition is performed for the fluid field only and is then transferred to the ALE field to guarantee the same layout. This reduces communication and data exchange among processors throughout the computation. In the sequel, domain decomposition of the ALE field is not mentioned anymore, since it is assumed to be matching with the fluid's decomposition.*

This mismatch of subdomains at the interface can be overcome by basing the partitioning on a *monolithic graph*, that consists of the solid and fluid graphs and also reflects the interface coupling. It is created as the combination of the solid and the fluid graph with additional insertion of off-diagonal coupling entries for the interface coupling. The coupling can be extracted from the mortar projection operator \mathcal{P} defined in (3.43) where a non-zero entry \mathcal{P}_{ij} indicates the coupling between the i th degree of freedom of the slave field to the j th degree of freedom of the master field and vice versa. It is created as an auxiliary graph purely for the purpose of creating the

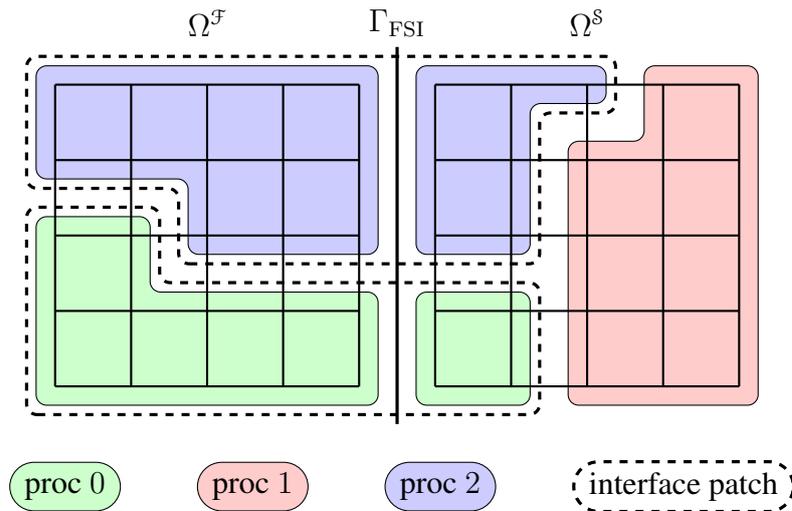


Figure 5.6: Overlapping domain decomposition based on a monolithic graph of a FSI problem — In such a decomposition, subdomains span across the interface like 'proc 0' and 'proc 2'. They are crucial for the effectiveness of the proposed preconditioner. Some processors might not own portions of both fields, e.g. 'proc 1'. Overlap of subdomains is not depicted for clarity of presentation.

desired overlapping DD. If the monolithic graph has been constructed properly, a graph-based partitioner cannot distinguish between solid and fluid field and, thus, will produce subdomains, that are likely to span across the interface as illustrated in Figure 5.6. At the interface, opposite solid and fluid subdomains reside on the same processor, namely 'proc 0' and 'proc 2' in Figure 5.6. These processors, that can be seen as patches spanning across the interface, will play a key role in the design of the proposed preconditioner. On the other hand, some processors might not own a portion of each field, for example 'proc 1' in Figure 5.6, that only owns solid nodes, but no fluid and ALE nodes. Again, coloring of the subdomains is done based on the 'interior' nodes of each subdomain, while the overlap is not visualized for simplicity of illustration. A physics-based overlapping DD and one based on a monolithic graph are compared to each other in Figure 5.7 using a one-dimensional setting. The mesh (*bottom*) is divided in solid and fluid portions Ω^s and Ω^f by the fluid-structure interface Γ_{FSI} . The physics-based overlapping DD (*middle*) partitions each physical field into three subdomains with the interface coinciding with a subdomain boundary. By partitioning of a monolithic graph (*top*), only three subdomains are generated in total, where one of them spans across the interface, namely the one associated with 'proc 1'. In particular this subdomain is supposed to address error accumulation at the interface, that results from the physics-based block preconditioners.

Remark 5.2.2 A fundamental difference between the physics-based overlapping domain decomposition and the one obtained from a monolithic graph is as follows: In the former one, every processor owns nodes of every physical field. In the latter one, only the subdomains that span across the interface have their share of the solid and the fluid field, while those subdomains away from the interface contain portions of either the solid or the fluid and the ALE field only.

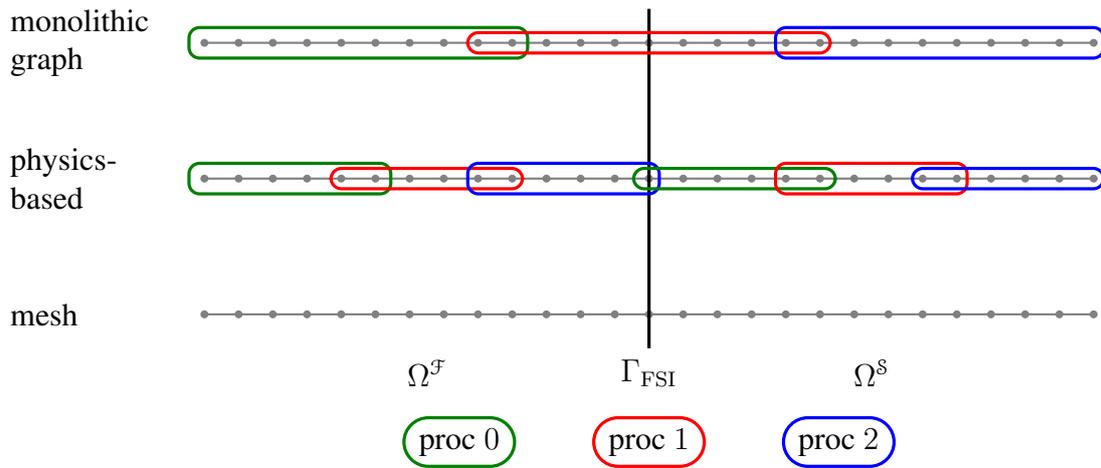


Figure 5.7: Overlapping domain decompositions for FSI preconditioners — An exemplary one-dimensional FSI mesh (*bottom*) is partitioned by a physics-based overlapping DD (*middle*) and an overlapping DD based on a monolithic graph (*top*). In the latter case, the subdomain associated with 'proc 1' spans across the interface Γ_{FSI} .

This matters in certain aspects of the implementation that are related to the layout of the MPI communicator.

The implementation in the in-house research code `Baci` [243] heavily relies on the fact that all processors own portions of all physical fields, since many steps throughout the computation are done field-by-field-wise, e.g. the residual evaluation. If one processor does not contain any solid nodes for example, this processor is idle during the solid's residual evaluation. This is not an optimal design when thinking in terms of parallel performance. However, loss of performance is outweighed by the speed-up due to the new preconditioner as can be seen in the performance evaluation in section 6.1.2. This issue can be cured by a more sophisticated, but appropriate design of the MPI communicators. By introducing sub-communicators that group processors according to the number and type of fields that they own and applying a suitable load-balancing strategy, it can be avoided that single processors are idle. As this requires major restructuring of the setup of the MPI communicators, this is subject to future work.

Alternatively, one could come up with an overlapping domain decomposition that on the one hand exhibits interface-spanning subdomains to satisfy the requirements for the hybrid preconditioner, but on the other hand is such that every processor owns solid, fluid, and ALE nodes. Such an approach has been tested by NOLL [177]. However, creation of this type of overlapping domain decomposition turned out to be a very difficult and tricky task especially in the case of a large number of subdomains.

Using the overlapping domain decomposition based on a monolithic graph, the linear system (5.1) is rather presented as

$$\mathbf{Ax} = \mathbf{b} \quad (5.8)$$

with \mathbf{A} , \mathbf{x} and \mathbf{b} replacing the system matrix \mathcal{J}_ℓ , the solution increment vector $\Delta \mathbf{x}_\ell$, and the right-hand side vector $-\mathbf{r}_\ell$. This also indicates, that the block structure is of no importance.

Sorting all unknowns by their affiliation to subdomains yields the matrix representation

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{00} & \mathbf{A}_{01} & \cdots & \mathbf{A}_{0n} \\ \mathbf{A}_{10} & \mathbf{A}_{11} & \cdots & \mathbf{A}_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{n0} & \mathbf{A}_{n1} & \cdots & \mathbf{A}_{nn} \end{bmatrix} \quad (5.9)$$

distributed among n subdomains, where n usually equals the number of processors n^{core} . Matrices \mathbf{A}_{ii} are restrictions of the global matrix \mathbf{A} to processor i , while the off-diagonal matrices \mathbf{A}_{ij} account for the coupling between the local subproblems on processors i and j . All processor-local matrices in (5.9), especially the off-diagonal ones, are sparse.

5.2.3 Setup and Application of the Preconditioner

To setup the combined preconditioner, two building blocks are necessary, namely any of the already existing physics-based block preconditioners $\mathcal{M}_{\text{MS}}^{-1}$ from Section 5.1 plus the additional preconditioner $\mathcal{M}_{\text{AS}}^{-1}$. Construction, setup, and implementation of the physics-based block preconditioners has already been detailed in Section 5.1. The additional preconditioner $\mathcal{M}_{\text{AS}}^{-1}$ is obtained by dropping all off-diagonal coupling blocks in (5.9), which results in an additive SCHWARZ preconditioner

$$\mathcal{M}_{\text{AS}}^{-1} = \begin{bmatrix} \mathbf{A}_{00} & & & \\ & \mathbf{A}_{11} & & \\ & & \ddots & \\ & & & \mathbf{A}_{nn} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}_{00}^{-1} & & & \\ & \mathbf{A}_{11}^{-1} & & \\ & & \ddots & \\ & & & \mathbf{A}_{nn}^{-1} \end{bmatrix}. \quad (5.10)$$

In order to tackle error accumulation due to physics-based block preconditioners and to benefit from the domain decomposition, where subdomains span across the interface on purpose, the subdomain solve \mathbf{A}_{ii}^{-1} on processor i needs to be of a type that is insensitive to mixed-physics block matrices. This rules out BGS-type methods as used in physics-based preconditioners, but opens the field to *direct* solvers. Such high quality solves on subdomains, that span across the interface, enable an efficient reduction of errors remaining from the application of a physics-based block preconditioner.

Commonly used direct solvers are often based on LU factorizations, that decompose the matrix \mathbf{A} of the linear system into a lower triangular matrix \mathbf{L} and an upper triangular matrix \mathbf{U} such that $\mathbf{A} = \mathbf{LU}$ [44, 166, 232]. However, matrices \mathbf{L} and \mathbf{U} are no longer sparse even for sparse \mathbf{A} , which usually makes their computation too expensive in terms of time and memory consumption. This can be remedied by approximations usually referred to as *incomplete LU (ILU)* factorizations [37, 165, 207]. Approximation quality of ILU factorizations can be controlled by choosing a fill level k , which is then denoted by $\text{ILU}(k)$. The cheapest method is $\text{ILU}(0)$, that proceeds as in a full LU factorization, but drops all entries in \mathbf{L} and \mathbf{U} , that are not in the sparsity pattern of \mathbf{A} . Larger fill levels $k > 0$ allow for additional fill-in in \mathbf{L} and \mathbf{U} , which brings them 'closer' to the exact factorizations. Often, $\text{ILU}(0)$ is sufficient for a strong preconditioning effect, while additional fill-in does not pay off when having in mind the increased setup

and storage costs [209] for fill levels $k > 0$. In this thesis, such incomplete LU factorizations are performed by Trilinos' `Itpack` package [209]. The cost of the factorizations is directly related to the size of the subdomains. To allow for a fast setup of the additive SCHWARZ preconditioner, sizes of subdomains should not be too large. In the numerical example in Section 6.1.2, they are chosen in the range of $n^{\text{dof}}/\text{core} \approx 7600$.

Now, light can be shed on the notations. The physics-based block preconditioners $\mathcal{M}_{\text{MS}}^{-1}$ from Section 5.1 are formally of multiplicative SCHWARZ type, which has been indicated by the subscript MS. Respectively, the notation $\mathcal{M}_{\text{AS}}^{-1}$ of the additional preconditioner with its subscript AS refers to additive SCHWARZ methods.

Remark 5.2.3 *The initial intention in the design of the preconditioner was to address error accumulation particularly at the fluid-structure interface only as the error in the interior of each subdomain is effectively addressed by the preconditioner $\mathcal{M}_{\text{MS}}^{-1}$. Thus, one could perform subdomain solves only on those subdomains that have their share of the interface. All other subdomains away from the interface can be dropped, i.e., their local subdomain solves are replaced by an identity matrix \mathbf{I} . Assuming 3 subdomains in total and assuming that only processors 0 and 2 contain a portion of the interface, the additive SCHWARZ preconditioner would read*

$$\mathcal{M}_{\text{AS}}^{-1} = \begin{bmatrix} \mathbf{A}_{00}^{-1} & & \\ & \mathbf{I} & \\ & & \mathbf{A}_{22}^{-1} \end{bmatrix}.$$

However, this does not result in savings of setup cost, as factorizations of \mathbf{A}_{ii} are performed simultaneously on each processor. Processors with \mathbf{I} would be idle, waiting for processors with a portion of the interface to finish their factorizations. Hence, in this thesis all processors perform local solves \mathbf{A}_{ii}^{-1} , which does not result in additional setup cost, but strengthens the effect of the preconditioner even away from the interface.

The physics-based block preconditioner $\mathcal{M}_{\text{MS}}^{-1}$ and the additional additive SCHWARZ preconditioner $\mathcal{M}_{\text{AS}}^{-1}$ are chained together to form the *hybrid additive/multiplicative SCHWARZ preconditioner*. Equally, the shorter expression *hybrid SCHWARZ preconditioner* is used in this thesis. It is applied as a symmetric multiplicative SCHWARZ method, reading

$$\mathcal{M}_{\text{HS}}^{-1} = \mathcal{M}_{\text{AS}}^{-1} \circ \mathcal{M}_{\text{MS}}^{-1} \circ \mathcal{M}_{\text{AS}}^{-1} \quad (5.11)$$

where the additive SCHWARZ preconditioner is applied before and after the physics-based block preconditioner. In GMRES iteration k , the preconditioner (5.11) is applied to the linear system (5.8) via three stationary RICHARDSON iterations

$$\begin{aligned} \mathbf{x}_{\text{I}}^k &= \mathbf{x}_0^k + \omega_{\text{AS}} \mathcal{M}_{\text{AS}}^{-1} (\mathbf{b} - \mathbf{A} \mathbf{x}_0^k) \\ \mathbf{x}_{\text{II}}^k &= \mathbf{x}_{\text{I}}^k + \omega_{\text{MS}} \mathcal{M}_{\text{MS}}^{-1} (\mathbf{b} - \mathbf{A} \mathbf{x}_{\text{I}}^k) \\ \mathbf{x}_{\text{III}}^k &= \mathbf{x}_{\text{II}}^k + \omega_{\text{AS}} \mathcal{M}_{\text{AS}}^{-1} (\mathbf{b} - \mathbf{A} \mathbf{x}_{\text{II}}^k) \end{aligned} \quad (5.12)$$

with damping parameters ω_{AS} and ω_{MS} and the initial guess \mathbf{x}_0^k . Intermediate steps after the first and second RICHARDSON iteration are denoted by \mathbf{x}_{I}^k and \mathbf{x}_{II}^k , respectively, while the final

result of the preconditioning operation is referred to as $\mathbf{x}_{\text{III}}^k$. Basically, it is possible to perform multiple iterations of each of the three RICHARDSON iterations in (5.12) and also to wrap (5.12) in a RICHARDSON iteration itself. However, these possibilities are not exploited for practical applications. Additionally, damping parameters are chosen as $\omega_{\text{AS}} = 1$ and $\omega_{\text{MS}} = 1$ in this thesis.

Remark 5.2.4 *One-level additive SCHWARZ methods are known to result in increased iteration counts, when the number of subdomains is increased [220]. Thus, the additive SCHWARZ preconditioner $\mathcal{M}_{\text{AS}}^{-1}$ is never applied as the only preconditioner without any multigrid preconditioner. It is always applied in the hybrid setting together with a physics-based multigrid preconditioner as given in (5.11) to enable mesh independency as demonstrated in the numerical example in Section 6.1.2.*

5.2.4 Proof of Concept and Demonstration of Improved Error Reduction

To demonstrate the basic principle behind the proposed hybrid preconditioner numerically, a reduced version of the well-known pressure wave example is studied. For a detailed description of the example see Section 6.1. A very coarse mesh is used. The solid portion consists of 5904 unknowns, while fluid and ALE use 15908 and 11931 degrees of freedom, respectively. The total number of unknowns is 33743. The problem is solved on 4 processors using an overlapping domain decomposition based on a monolithic graph of the coupled problem.

For simplicity, only the linear system of equations in the first nonlinear iteration of the first time step is considered. This system can be seen as exemplary for all time steps of the simulation, since iteration counts are rather constant throughout the entire simulation as will be shown in Section 6.1.2. The effectiveness of the hybrid preconditioner is assessed by comparing error reduction through different preconditioners. On the one hand, a purely physics-based block-iterative preconditioner, cf. Section 5.1.2, is applied. Assuming exact block inverses for each block within the BGS method, errors after application of the preconditioner are only due to the BGS method. Exact block inverses are computed via a direct solver, namely a LU decomposition. Thus, this preconditioner is referred to as BGS(LU). On the other hand, the newly proposed hybrid preconditioner is configured as follows: The additive SCHWARZ portion uses direct LU-based solves for each subdomain, while the interior multiplicative SCHWARZ method is the aforementioned BGS(LU) approach to augment comparability. It is referred to as H-BGS(LU).

Two tests are performed: First, the linear system of equations is solved exactly. Efficiency of the preconditioner is assessed in terms of the number of GMRES iterations required to reach machine precision. The pure BGS(LU) method requires 41 GMRES iterations, while the hybrid approach only takes 11 GMRES iterations until convergence to machine precision.

Second, the number of GMRES iterations is limited as detailed below. The effect of preconditioning is evaluated by the achieved relative residual reduction as well as the remaining error, i.e. the deviation of the approximate GMRES solution from the exact solution obtained in the first test. For the hybrid preconditioner, only a single GMRES iteration is performed to reduce error reduction to the effect of preconditioning as much as possible. One sweep of H-BGS(LU) consists of three applications of LU-type preconditioners, namely the pre- and

post-application of the additive SCHWARZ preconditioner plus one sweep of BGS(LU) in between. To achieve comparability, three GMRES iterations are performed with pure BGS(LU) to also apply a LU-based method three times in total. The hybrid preconditioner achieves a relative residual reduction $\|\mathbf{r}_{\text{lin}}^1\|_2 / \|\mathbf{r}_{\text{lin}}^0\|_2 = 6.9 \cdot 10^{-3}$, while the pure BGS(LU) allows for a reduction up to $\|\mathbf{r}_{\text{lin}}^3\|_2 / \|\mathbf{r}_{\text{lin}}^0\|_2 = 5.2 \cdot 10^{-2}$, only. A visualization of the error is shown in Figure 5.8. For the BGS(LU) preconditioner, the error after three GMRES iterations is plotted in Figure 5.8(a). In the solid, the error is at the order of 10^{-4} (*left*) with slightly larger values at the fluid-structure interface. The accumulation of error at the interface is even more pronounced for the fluid velocity (*middle*) and fluid pressure (*right*), which are of order 10^0 and $10^3 - 10^4$, respectively. The same analysis for the hybrid preconditioner H-BGS(LU) is shown in Figure 5.8(b), showing that massive error reductions could be achieved. The maximum error in solid displacements is close to the fluid-structure interface, but at the order of 10^{-8} , which resembles a reduction by four orders of magnitude. Similar reductions are reported for errors in fluid velocities and fluid pressure, whose maximum values are now at the order of 10^{-4} and 10^{-1} , respectively. A graphical representation of these reductions is given in Figure 5.8(c). The circular geometry is cut in half. The upper half reports the errors for pure BGS(LU), the lower half those for H-BGS(LU). Color scales are calibrated such that they span the combined range of errors of BGS(LU) and H-BGS(LU). The significant reductions of the error by the hybrid preconditioner compared to the purely physics-based one can be seen clearly.

Summing up, the idea behind the hybrid additive/multiplicative SCHWARZ preconditioner could be confirmed numerically. Huge gains in error reduction could be achieved in this simple example. Further performance analysis on a larger scale will be performed in Section 6.1.2.

5.3 Practical Aspects of Monolithic Linear and Nonlinear Solvers for Coupled Problems

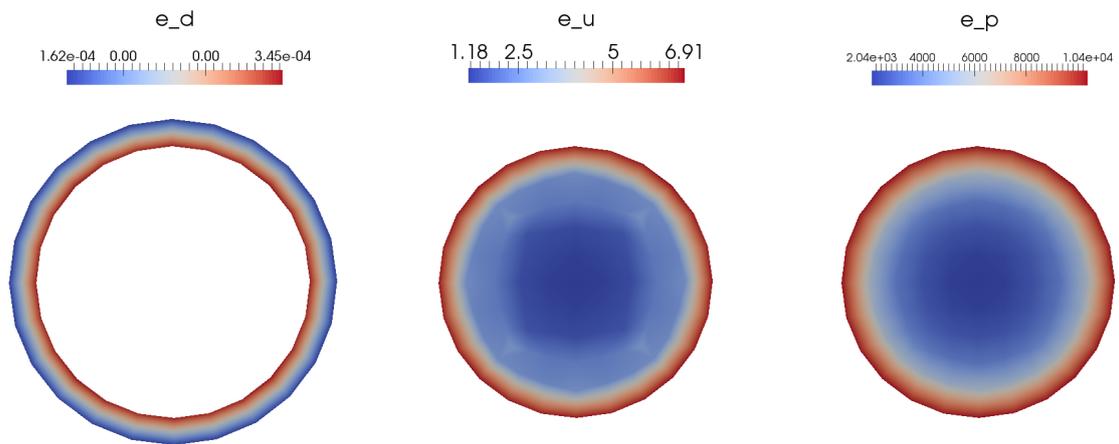
In this section, practical aspects of monolithic solvers for coupled problems are discussed, namely the type of convergence check as well as the type of norms to be used. Both linear and nonlinear solvers are addressed.

Since both the nonlinear and the linear solver are iterative schemes, appropriate stopping criteria have to be provided to decide whether the iterative procedure can be stopped or needs to be continued. Basically, one can stop iterating if the numerical solution is *close enough* to the exact solution. Lacking the knowledge of the exact solution, one usually aims at reducing the norm of a residual vector below a tolerance ε and accepts the numerical solution associated with that residual as the final approximation to the exact solution. In addition, the norm of the solution increment $\Delta \mathbf{x}$ can be monitored and included into the convergence check.

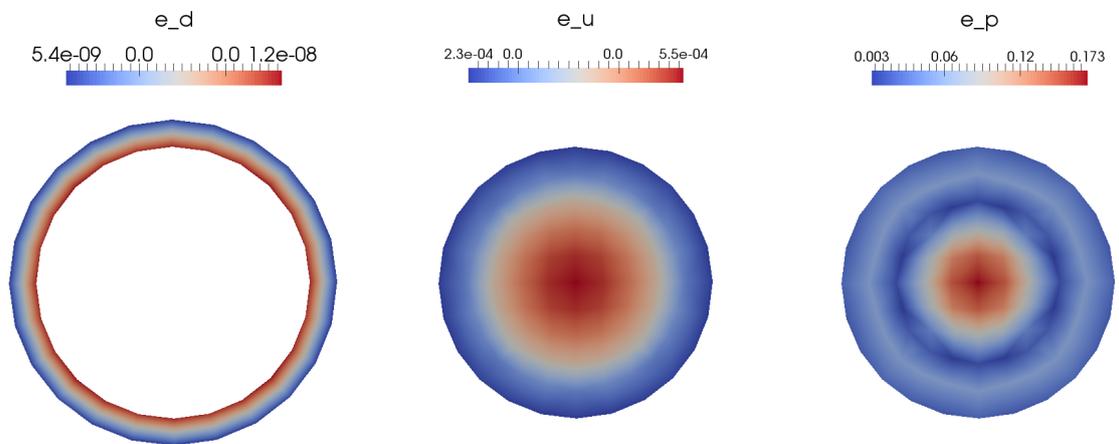
Common norms are EUCLIDIAN vector norms, namely the length-scaled 2-norm $\|(\bullet)\|_2$ as already introduced in (4.32) and the inf-norm $\|(\bullet)\|_\infty$ given as

$$\|(\bullet)\|_\infty = \max \{ |(\bullet)_m| \}$$

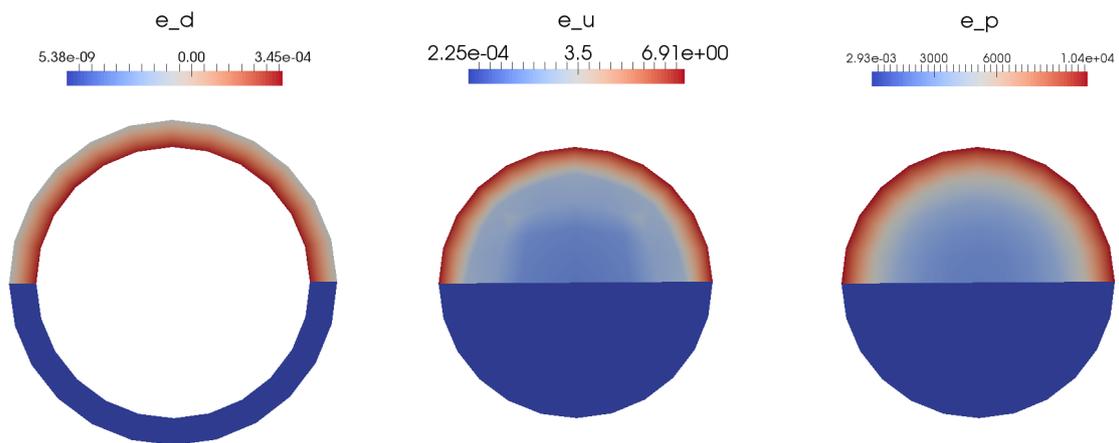
with $m = 1, \dots, M$ and M being the number of entries in the vector (\bullet) . In opposite to the 2-norm, a length-scaling is not necessary for the inf-norm, since its calculation does not involve a summation of all entries.



(a) Error after 3 GMRES iterations with BGS(LU) preconditioning



(b) Error after 1 GMRES iteration with H-BGS(LU) preconditioning



(c) Comparison of BGS(LU) (top) and H-BGS(LU) (bottom)

Figure 5.8: Comparison of the error reduction by BGS(LU) and H-BGS(LU) — Errors after three GMRES iterations with BGS(LU) preconditioning are compared to errors after one GMRES iteration with H-BGS(LU) preconditioning. Errors in solid displacements, fluid velocities, and fluid pressures denoted by e_d , e_u , and e_p , respectively, are visualized on a cross section of the pressure wave example. The hybrid preconditioner is more effective than the purely physics-based BGS(LU) approach.

5.3.1 Convergence Check for the Nonlinear Solver

In the context of solving a system of nonlinear equations, the stopping criterion is usually based on the comparison of a norm of the nonlinear residual vector \mathbf{r} to a user-given tolerance $\varepsilon_{\mathbf{r}}^{\text{nl}}$. The stopping criterion is satisfied if

$$\|\mathbf{r}\| \leq \varepsilon_{\mathbf{r}}^{\text{nl}}.$$

The norm of the solution increment can be included in the convergence check via additionally asking for

$$\|\Delta\mathbf{x}\| \leq \varepsilon_{\Delta\mathbf{x}}^{\text{nl}}.$$

For both convergence checks, 2-norm and inf-norm are applicable. In this thesis, the nonlinear convergence check is always based on absolute norms. In practical applications, often weighted norms [198] or a combination of absolute and relative tolerance [134] seem to be useful.

In multi-physics applications like the FSI problem at hand, both the global solution and residual vector are assembled based on solution and residual vectors of each field involved in the problem, cf. e.g. (3.8) for the global FSI residual. However, there is no guarantee that the portions from each field are somehow balanced, neither w.r.t. size nor w.r.t. their magnitude. It may happen — and this is usually the case — that solid and fluid field differ significantly in size and magnitude of their residual vectors. While differences in size are due to geometric dimensions and spatial discretization, discrepancy in magnitude can have several reasons. On the one hand, different systems of units may be used in both fields, but even with the same system of units differences in physical properties may lead to differences in the magnitude of the residual vector. On the other hand, the initial residual vector depends on the initial guess of the solution vector. This initial guess can be of varying quality in both fields, which might lead to a small residual contribution of one field, if its initial guess is very good, while the other field produces a large residual due to a less accurate initial guess.

Having in mind the possibly huge variations of the contributions to the global residual vector, it seems to be dangerous to judge about convergence based on norms of the global residual vector, only. Especially when using a 2-norm, the residual norm might be dominated by one of the involved fields such that no control over the other fields can be guaranteed. This dominating effect can be either based on the length scaling included in the 2-norm or based on the different magnitudes of the field residuals. Though, even application of the inf-norm might be problematic since choosing a single tolerance does not reflect for possible variations in magnitude of the field residuals. Similar arguments hold for testing the solution increment vector.

To circumvent the aforementioned issues, a more sophisticated convergence check is performed in this thesis that reflects the contributions of the single fields as well as the coupling between them. The nonlinear monolithic residual as well as the monolithic solution increment vector are decomposed into physics-based portions, namely

- all entries related to the solid's displacement degrees of freedom: $\mathbf{r}^{\mathcal{S}}, \Delta\mathbf{d}^{\mathcal{S}}$
- all entries related to the fluid's velocity degrees of freedom: $\mathbf{r}_{\mathbf{u}}^{\mathcal{F}}, \Delta\mathbf{u}^{\mathcal{F}}$
- all entries related to the fluid's pressure degrees of freedom: $\mathbf{r}_p^{\mathcal{F}}, \Delta p^{\mathcal{F}}$
- all entries related to the fluid-structure interface: $\mathbf{r}_{\Gamma_{\text{FSI}}}^{\text{ma}}, \Delta\mathbf{x}_{\Gamma_{\text{FSI}}}^{\text{ma}}$ with $\text{ma} \in \{\mathcal{S}, \mathcal{F}\}$ depending on the choice of master and slave side of the mortar coupling

Similarly as for error estimation in the adaptive time stepping, cf. Section 4.2.1, the actual subdivisions depend on the choice of master and slave side of the mortar method. If exemplarily choosing the solid as the master field, the solid portion contains all solid degrees of freedom including those located on the fluid-structure interface. Then, the fluid's velocity portion only accounts for the interior fluid velocity degrees of freedom away from the interface. However, the pressure residual takes all pressure degrees of freedom into account, since the pressure is not affected by the interface condensation process. Finally, the solid's interface degrees of freedom are extracted into a separate set to account for the central role of the fluid-structure interface. In case of the fluid being the master side, this strategy stays the same, however, the interface is assigned to the fluid field, respectively. For each of these physics-based portions, both 2-norm and inf-norm are required to satisfy user-given tolerances, that may be different for each vector portion. To achieve convergence, all individual tests must be passed at the same time, which is equivalent to tie all individual tests together with a logical AND relation.

The FSI solver in `Baci` [243] does not implement the nonlinear solver by itself, but employs the `NOX` package from the *Trilinos* project [116]. This library also provides a framework for convergence tests. Besides testing residual and increment norms as described above, additional tests for valid numbers and the number of nonlinear iterations are performed. An exemplary screen output is given in Figure 5.9. Four groups associated with solid, interface, fluid velocity and fluid pressure degrees of freedom can be seen, denoted by `DISPL`, `GAMMA`, `VELOC`, and `PRESS`, respectively. Each group tests 2-norm and inf-norm of its residual (denoted by `residual`) as well as its solution increment vector (denoted by `update`). At the very bottom, the 2-norms of the global residual and solution increment vectors are given, which show the difference between the global norms and the field-wise norms. The test is passed if and only if all individual tests are satisfied at the same time, resembling a combination of the tests by a logical AND.

Choosing all these tolerances is up to the user. The computational engineer can select meaningful tolerances, where the influence of the system of units, the problem size, and the desired accuracy need to be taken into account. Usually, physical insight into the problem is helpful. General rules cannot be provided. A possible strategy is outlined in [198].

5.3.2 Convergence Check for the Linear Solver

The linear system of equations is solved with the preconditioned GMRES method by `SAAD` and `SCHULTZ` [208], using the implementation in the `Aztec` package [115] from the *Trilinos* library [116]. Convergence is tested by means of a relative residual norm, reading

$$\frac{\|\mathbf{r}_{\text{lin}}^i\|_2}{\|\mathbf{r}_{\text{lin}}^0\|_2} \leq \varepsilon^{\text{lin}}$$

with $\|\mathbf{r}_{\text{lin}}^i\|_2$ denoting the 2-norm of the linear residual in GMRES iteration i which is normalized with the initial residual norm $\|\mathbf{r}_{\text{lin}}^0\|_2$. The base tolerance ε^{lin} is given by the user with typical values being in the range of $10^{-4} - 10^{-5}$. Its interplay with the nonlinear convergence tolerances needs to be considered to obtain a reasonable value.

With progress of the nonlinear solver, the nonlinear residual \mathbf{r}^k probably consists of small entries, which might approach the limit of machine precision. In such scenarios, it might be

```

*****
-- Status Test Results --
**.....OR Combination ->
**.....Finite Number Check (Two-Norm F) = Finite
**.....AND Combination ->
  Converged....AND Combination ->
    Converged....DISPL residual Two-Norm = 2.508e-11 < 1.000e-08
    Converged....DISPL residual Max-Norm = 3.143e-10 < 1.000e-08
    Converged....DISPL update Two-Norm = 2.302e-10 < 1.000e-08
    Converged....DISPL update Max-Norm = 1.030e-09 < 1.000e-08
  **.....AND Combination ->
    Converged....GAMMA residual Two-Norm = 3.546e-11 < 1.000e-09
    Converged....GAMMA residual Max-Norm = 3.143e-10 < 1.000e-09
    Converged....GAMMA update Two-Norm = 2.306e-10 < 1.000e-09
    **.....GAMMA update Max-Norm = 1.030e-09 < 1.000e-09
  **.....AND Combination ->
    Converged....VELOC residual Two-Norm = 4.605e-12 < 1.000e-08
    Converged....VELOC residual Max-Norm = 3.642e-11 < 1.000e-08
    **.....VELOC update Two-Norm = 6.304e-08 < 1.000e-08
    **.....VELOC update Max-Norm = 1.372e-06 < 1.000e-08
  **.....AND Combination ->
    Converged....PRESS residual Two-Norm = 1.395e-15 < 1.000e-08
    Converged....PRESS residual Max-Norm = 2.162e-14 < 1.000e-08
    **.....PRESS update Two-Norm = 1.981e-05 < 1.000e-08
    **.....PRESS update Max-Norm = 8.414e-05 < 1.000e-08
  **.....Number of Iterations = 4 < 10
*****

*****
-- Nonlinear Solver Step 4 --
||F|| = 4.389e-08 step = 1.000e+00 dx = 5.445e-03
*****

```

Figure 5.9: Exemplary screen output of nonlinear convergence test for monolithic FSI in `Baci` — Based on Trilinos’ `NOX::StatusTest`, a combination of field convergence tests for residuals and solution increments related to solid displacements (`DISPL`), interface quantities (`GAMMA`), fluid velocities (`VELOC`), and fluid pressures (`PRESS`) can be performed. Additional checks for finite numbers and iteration counts are included.

very expensive or even unfeasible to converge the linear solver to its base tolerance ε^{lin} . This is remedied by adapting the convergence test to

$$\frac{\|\mathbf{r}_{\text{lin}}^i\|_2}{\|\mathbf{r}_{\text{lin}}^0\|_2} \leq \max \left\{ 1, \frac{\beta^{\text{lin}}}{\|\mathbf{r}^k\|_2} \right\} \cdot \varepsilon^{\text{lin}}.$$

The scalar factor β^{lin} is usually chosen in the range of $10^{-2} - 10^{-3}$. It loosens the effective tolerance for the linear convergence check in case that the nonlinear residual norm $\|\mathbf{r}^k\|_2$ is smaller than β^{lin} , such that the linear solver is required to converge to a tolerance, that is less tight with ongoing convergence of the nonlinear solver. This strategy saves computational time and avoids aiming at accuracies in the linear solver that are infeasible to achieve.

5.4 Summary

In this chapter, key components of monolithic solvers for FSI problems have been discussed. Focus was put on the preconditioning of the monolithic system of linear equations. Existing physics-based block preconditioners tailored to FSI problems have been reviewed briefly. To address one of their drawbacks, namely the accumulation of error at the fluid-structure interface, a novel hybrid additive/multiplicative SCHWARZ preconditioner has been proposed. It combines the existing physics-based approaches with an additional additive SCHWARZ preconditioner. The latter one is based on an overlapping domain decomposition that purposely consists of subdomains spanning across the fluid-structure interface. Through the application of incomplete LU factorizations as subdomain solvers in this additional preconditioner, error accumulation can be reduced effectively. A simplified example has been shown to proof the basic concept. Finally, some remarks on the convergence check for iterative linear and nonlinear solvers have been given where specifics of monolithic solution processes for coupled multi-physics problems have been addressed.

6 Numerical Examples

Besides the numerical examples in Section 4.3, where the focus was put on demonstration and discussion of the newly proposed adaptive time stepping scheme, two three-dimensional examples are shown in the sequel. First, a pressure wave through an elastic tube is used as a benchmark problem for the monolithic FSI solver, where time integration and preconditioning techniques will be examined. Second, real-world applicability of the proposed framework will be demonstrated by the example of pulsatile blood flow through a patient-specific *abdominal aortic aneurysm* (AAA), where the adaptive time stepping will play an important role again. In both examples, efficiency will be of particular interest.

Simulations have been performed with the in-house research code `Baci` [243]. Details on the fluid stabilization have already been given at the beginning of Section 4.3. Specifications of the used hardware resources can be found in Appendix B.

6.1 Pressure Wave through an Elastic Tube

As a benchmark problem for monolithic FSI solvers, the well-known pressure wave through an elastic tube, originally proposed in [91], is studied. It is designed to mimic hemodynamic conditions, especially w.r.t. to the material densities with the ratio $\rho^s/\rho^f \approx 1$. MAYR *et al.* [163] used this example to discuss the influence of different time integration schemes on the solution. Superiority of monolithic FSI solvers in comparison to partitioned solution schemes has been demonstrated by KÜTTLER *et al.* [143]. A detailed analysis of the performance of the linear solvers has been performed by GEE *et al.* [89] where classical versions of the FSI-specific AMG preconditioners from Section 5.1 have been applied. Simulations with non-matching interface discretizations have been reported in literature, where KLÖPPEL *et al.* [138] used the dual mortar method to deal with non-matching grids at the interface, while DEPARIS *et al.* [55] and FORTI [83] use the INTERNODES approach [52]. It is often used as a benchmark for partitioned [8, 50, 57, 77–79] and monolithic solvers [9, 12, 40, 53, 65, 142, 148, 149] among others.

The geometry is depicted in Figure 6.1. The solid tube is clamped at both ends. The fluid is initially at rest. For the duration of $3 \cdot 10^{-3}$ s, it is loaded with a surface traction $\bar{\mathbf{h}}^f = 1.3332 \cdot 10^4 \text{ g} \cdot \text{cm}/\text{s}^2$ in z -direction at $z = 0$. At $z = \ell$, fluid velocities are prescribed to zero, meaning that the tube is closed at that end. As a result, a pressure wave travels along the tube's longitudinal axis and is reflected at the closed end of the tube. The constitutive behavior of the structure is modelled by a ST.-VENANT-KIRCHHOFF material. The fluid is assumed to be an incompressible NEWTONIAN fluid. The actual constitutive parameters are listed in Table 6.1.

The solid is discretized with `Hex8 F-bar` elements [47], while the fluid utilizes `P1P1` elements with residual-based stabilization as detailed at the beginning of this chapter. Different meshes are used, some of them with matching grids at the interface, some with non-matching

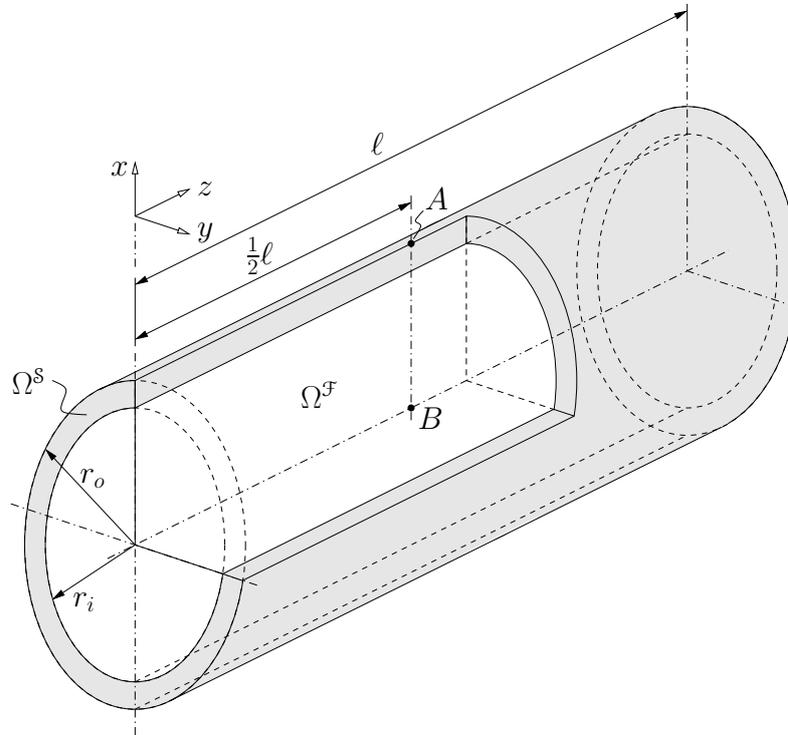


Figure 6.1: Geometry of pressure wave through an elastic tube — A solid tube (outer radius $r_o = 0.6$ cm, inner radius $r_i = 0.5$ cm, length $l = 5.0$ cm) is clamped at both ends and is filled with an incompressible NEWTONian fluid that is initially at rest.

Table 6.1: Material parameters for the pressure wave example

Fluid			Solid		
dynamic viscosity	μ_{dyn}^F	0.03 g/(cm · s)	YOUNG's modulus	E^S	$3.0 \cdot 10^6$ g/(cm · s ²)
density	ρ^F	1.0 g/cm ³	POISSON's ratio	ν^S	0.3
			density	ρ^S	1.2 g/cm ³

Table 6.2: Matching and non-matching meshes for the pressure wave example — Numbers of degrees of freedom per field and in total are reported. Meshes *pw1* – *pw6* use matching interface discretizations, while grids do not match at the interface for meshes *pw7* – *pw9*. By running each mesh on a specific number of cores n^{core} , the load per core can be kept roughly constant.

Mesh ID	$n^{\text{S,dof}}$	$n^{\text{F,dof}}$	$n^{\text{S,dof}}$	$n_{\text{total}}^{\text{dof}}$	n^{core}	$n^{\text{dof}}/\text{core}$
<i>pw1</i>	22896	56180	42135	121211	16	7576.7
<i>pw2</i>	45312	113516	85137	243965	32	7623.9
<i>pw3</i>	90000	228300	171225	489525	64	7648.8
<i>pw4</i>	185328	453420	340065	978813	128	7647.0
<i>pw5</i>	274680	680596	510447	1465723	192	7634.0
<i>pw6</i>	362304	906204	679653	1948161	256	7610.0
<i>pw7</i>	243936	689476	517107	1450519	56	25902.1
<i>pw8</i>	584640	1917124	1437843	3939607	140	28140.1
<i>pw9</i>	1804608	5410948	4058211	11273767	448	25164.7

grids. They differ in element size and, thus, result in different numbers of degrees of freedom. Meshes *pw1* – *pw6* with matching interface discretizations and meshes *pw7* – *pw9* with non-matching interface discretizations are detailed in Table 6.2. Mesh independency has been studied by MAYR *et al.* [163]. Based on their results, all meshes used in this thesis are considered as fine enough to render mesh independency of the solution. Two different hardware platforms are used. Meshes *pw1* – *pw6* are ran on the *Opteron* partition of an in-house cluster, while meshes *pw7* – *pw9* are solved on SuperMUC’s *Phase 2* of the Leibniz Supercomputing Center, Garching, Germany. Hardware details are summarized in Appendix B. The load per core is kept approximately constant at $\approx 7620 n^{\text{dof}}/\text{core}$ and $\approx 26400 n^{\text{dof}}/\text{core}$, respectively.

The exemplary mesh *pw7* is depicted in Figure 6.2. Displacement and pressure fields obtained with mesh *pw7* are shown in Figure 6.3. The pressure wave travels along the longitudinal axis of the tube, resulting in radial expansion of the tube’s wall. Locations of maximum fluid pressure and maximum radial expansion of the solid coincide. In Figure 6.3, the generalized- α method has been used for solid and fluid field with spectral radii $\rho_{\infty}^{\text{S}} = 0.8$ and $\rho_{\infty}^{\text{F}} = 0.7$, respectively.

6.1.1 On the Influence of Time Integration Schemes

Two phenomena, that are related to time integration, are studied in this section. First, the influence of different time integration schemes on the solution is discussed. Second, the energy production at the fluid-structure interface is monitored to confirm the theoretical findings of Section 3.2.4 numerically.

Damping Properties of Different Time Integration Schemes

A central aspect of the presented FSI solver is the freedom of choice of time integration schemes in solid and fluid field. This freedom as well as the effect of different combinations of time

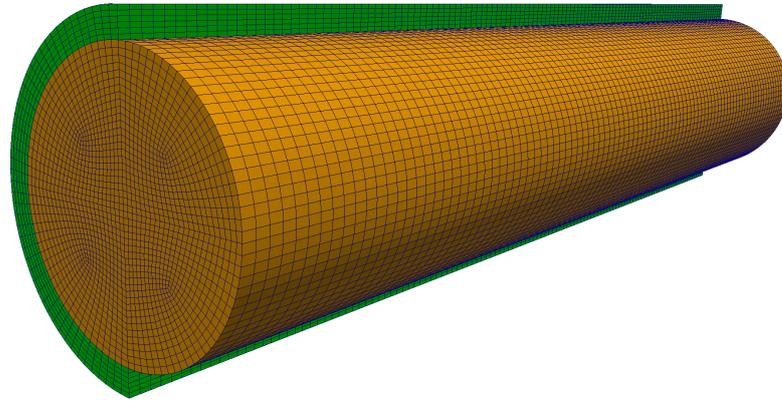


Figure 6.2: Mesh $pw7$ of pressure wave example — Solid and fluid are meshed with structured, pure `Hex` meshes with non-matching interface discretizations. The finer meshes $pw8$ and $pw9$ are then generated via global mesh refinement.

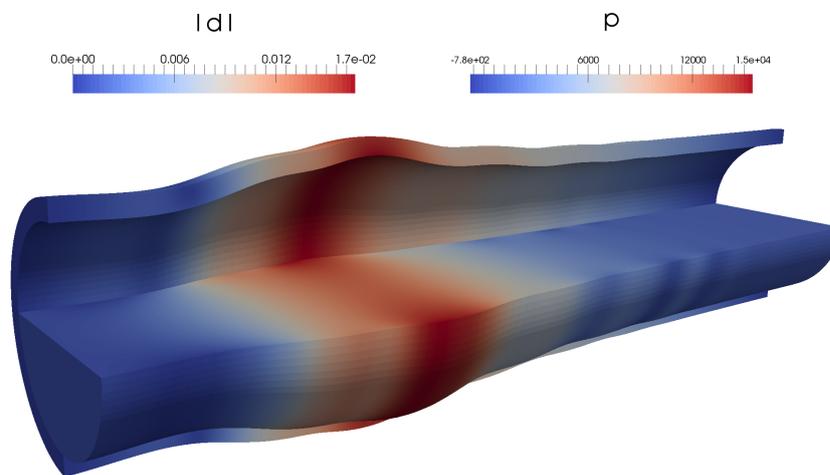


Figure 6.3: Cut view of displacement and pressure field in pressure wave at time $t = 0.005$ s — The solid domain is colored according to its displacement magnitude, denoted by $|d|$, while the fluid is colored according to the fluid pressure, denoted by p . Displacements are magnified by a factor of 10 for the purpose of visualization.

integration schemes with various time integration parameters in both fluid and structure field is now studied. A similar analysis has already been performed by MAYR *et al.* [163].

In the solid, the generalized- α method [39] is applied. In the fluid field, both the generalized- α method [129] and the one-step- θ scheme with various choices for their time integration parameters $\rho_\infty^{\mathcal{F}}$ and $\theta^{\mathcal{F}}$ are used.

The choice of time integration schemes and especially their numerical dissipation is studied in terms of solution quantities, namely the x -component $d_x^{\mathcal{S}}(t)$ of the solid's displacement at location A on the tube's outer surface and the fluid pressure $p^{\mathcal{F}}(t)$ at the tube's center point B . Locations of points A and B are indicated in Figure 6.1. Solution curves for different combinations of time integration schemes with various time integration parameters in both the fluid and the structure field are depicted in Figure 6.4. When both fields are discretized with generalized- α schemes, the influence of the actual parameter choice is rather small. The pressure field shows some fluctuations, when no or only little numerical dissipation is imposed. These fluctuations vanish the better, the more numerical dissipation is introduced into the system. However, the overall behavior of the solution is not affected by numerical dissipation. This changes dramatically, when one-step- θ time integration is utilized in the fluid field. Then, only the choice $\theta^{\mathcal{F}} = 0.5$ is free of numerical dissipation. The larger the value of $\theta^{\mathcal{F}}$, the more numerical dissipation is involved. Again, numerical dissipation reduces the fluctuations in the pressure field. Simultaneously, the amplitudes in the displacements as well as in the pressure are reduced significantly by a larger amount of numerical dissipation, i.e. the solution changes a lot. This becomes particularly evident in the displacement solution around times $t = 0.013$ s and $t = 0.017$ s, where local features of the undamped curves cannot be reproduced due to large numerical dissipation.

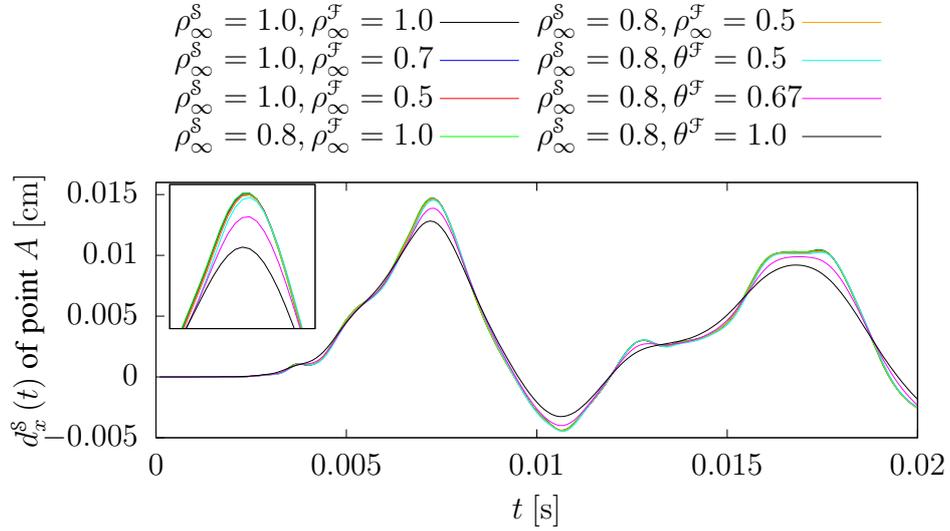
As can be seen from this comparison, numerical dissipation of one field does also highly affect the solution of the other field due to the interface coupling. Thus, numerical dissipation and its impact on the coupled problem needs always to be considered when choosing the time integration schemes for the solid and the fluid field.

Energy Considerations at the Fluid-Structure Interface

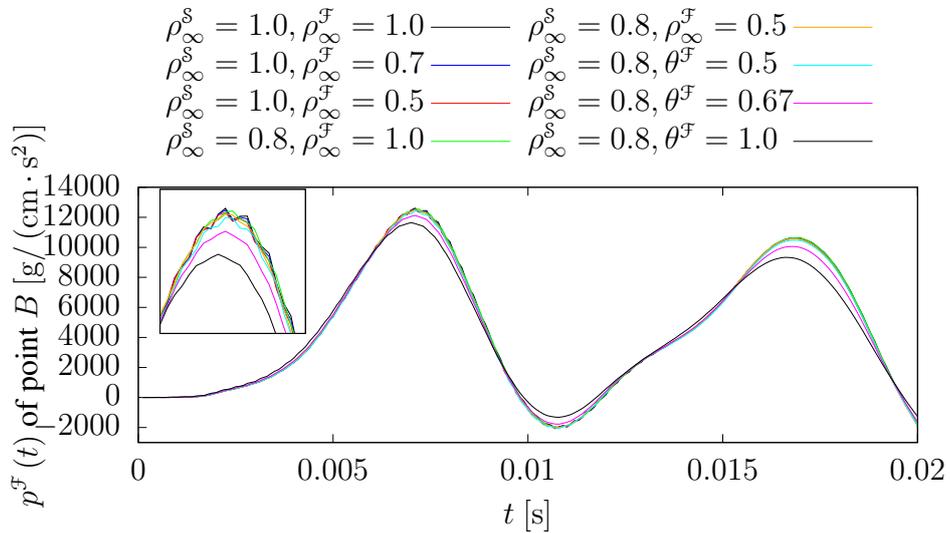
The production of spurious energy at the interface as given in (3.36) and (3.38) is studied numerically using meshes *pw1* and *pw7* with matching and non-matching interface discretizations, respectively. For comparison, the total energy of the solid, i.e. its elastic and kinetic energy, is used as indicator for the system's total energy. The fluid's kinetic energy is not considered since the fluid field exhibits only very little flow localized at the interface and its amount is not worth mentioning. Due to the implementation at hand, Hex8 elements using *enhanced assumed strains (EAS)* are used in the solid instead of Hex8 F-Bar elements to allow for the calculation of the solid's internal and kinetic energy.

Using the generalized- α method in the fluid and the solid field, the influence of the time discretization is studied with two cases that address the issue of possibly mismatching generalized midpoints $t_m^{\mathcal{S}}$ and $t_m^{\mathcal{F}}$:

- *Case I:* spectral radii are chosen equal, namely $\rho_\infty^{\mathcal{S}} = \rho_\infty^{\mathcal{F}} = 0.7$
- *Case II:* spectral radii differ, namely $\rho_\infty^{\mathcal{S}} = 0.8$ and $\rho_\infty^{\mathcal{F}} = 0.7$



(a) Radial displacement at location A



(b) Fluid pressure at location B

Figure 6.4: Various combinations of time integration schemes for the pressure wave example — For the structure field, generalized- α time integration with its spectral radius ρ_{∞}^s has been utilized. In the fluid field, either generalized- α time integration or a one-step- θ scheme have been employed, denoted by their parameters ρ_{∞}^f and θ^f , respectively. Close-ups for the first peak are shown revealing the damping effect of the one-step- θ scheme with $\theta^f > 0.5$.

In *case I*, energy production per time step at the interface is expected to vanish since $\rho_\infty^s = \rho_\infty^f$ and, thus, $a = b$ yielding $\Delta E_T^{n \rightarrow n+1} = 0$, cf. (3.36). Numerically, this expectation is confirmed up to machine precision. Consequently, energy accumulation (3.38) does not take place.

In *case II*, a certain amount of energy production per time step is expected. However, it should be small compared to the systems total energy. The solid's total energy is at the order of 10^2 , while its actual value shows minor variations over time. The production of interface energy as given in (3.36) and (3.38) ranges at the order of 10^{-2} and 10^0 , respectively, and, thus, are remarkably smaller. In combination with the fluid's viscous damping and the numerical dissipation of the time integration scheme, it is assumed that the production of artificial energy at the interface does not spoil stability of the solution.

These results are the same for both meshes. Thus, the mortar projections at the interface do not affect the energy production at all.

6.1.2 Discussion of FSI Preconditioners

The subsequent analysis of the proposed preconditioners is divided into two parts. First, the classic multigrid block preconditioners, that heavily rely on the separation of physics at the fluid-structure interface, are applied to solve problems with non-matching grids, cf. Section 6.1.2. A series of matching grids is used to study the behavior of the newly proposed hybrid additive/multiplicative SCHWARZ preconditioner, that combine the physics-based block preconditioners with an additional interface preconditioner to overcome drawbacks due to the separation of physics, cf. Section 6.1.2.

Efficiency and performance are assessed in terms of iteration counts and time measurements of preconditioner setup and linear solving. Since iteration counts are inherent to the algorithm, this number does not depend on the used hardware as well as its workload during the computation. Though, time measurements depend very much on the hardware at hand, e.g. differences in nominal frequency of the CPU or memory access bandwidth, and especially on the workload on the used cluster during the measurement. Especially on clusters, where output of results is written via the same network used for internode communication of data during a parallel computation, timings can be inaccurate if the computation of interest is delayed by other computations on the same hardware, that overload the internode communication with writing output. In this thesis, especially time measurements taken on the in-house cluster's *Opteron* partition, cf. Appendix B, suffer from the latter issue. Having in mind these possible inaccuracies, time measurements will be seen as a useful indicator for performance, but can not be considered as reliable and exact measurement data. Summing up, iteration counts are considered as more reliable, even if time to solution is of greater interest to the user in practical applications.

Physics-based AMG Block Preconditioners for FSI

The physics-based block preconditioners BGS(AMG), AMG(BGS), and AMG(Schur) from Section 5.1 are compared to each other. Performance is assessed in terms of GMRES iterations per time step, preconditioner setup time, and total solver time.

Configurations of the multigrid hierarchies and smoothers for each field are summarized in Table 6.3. Due to symmetry of their respective matrices, solid and ALE use CHEBYSHEV polynomials of degrees 6 and 12 as the fine and medium level smoothers, respectively. Three

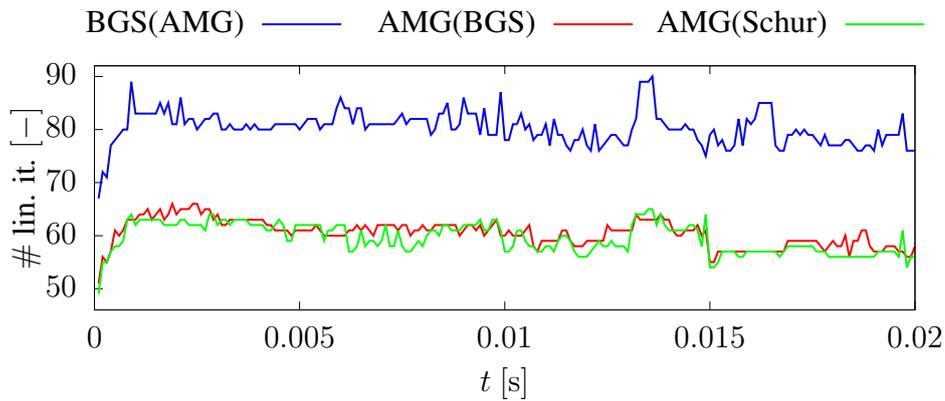
Table 6.3: Settings for multigrid hierarchies for pressure wave example — Solid and ALE use CHEBYSHEV smoothers on fine and medium level, while the fluid employs three sweeps of damped symmetric GAUSS–SEIDEL (SGS) on the fine level and incomplete LU with fill level 0 on intermediate levels. All fields use a direct solver on the coarsest level.

Field	Fine level	Medium levels	Coarse level
\mathcal{S}	CHEBYSHEV(6)	CHEBYSHEV(12)	LU
\mathcal{F}	SGS(3×0.69)	ILU(0)	LU
\mathcal{G}	CHEBYSHEV(6)	CHEBYSHEV(12)	LU

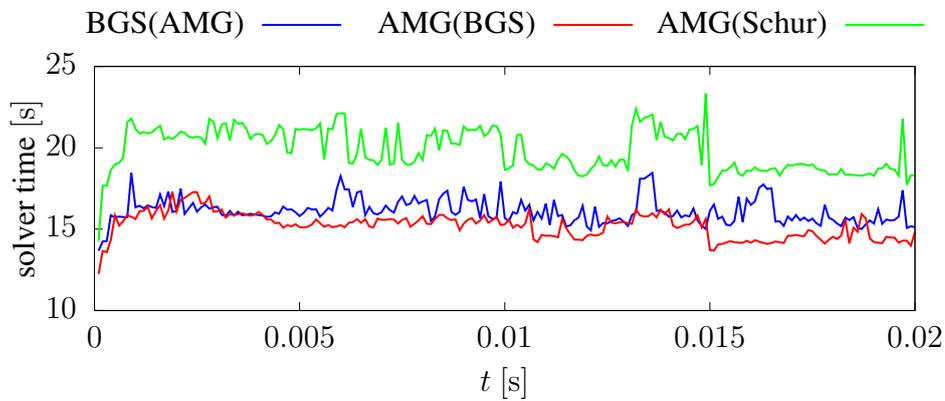
sweeps of damped symmetric GAUSS–SEIDEL (SGS) are applied to the fluid’s fine level problem, while on the medium levels an incomplete LU factorization with fill level 0 is employed. Coarse level problems of all fields are treated with a direct solver. Solid and ALE use smoothed aggregation transfer operators, while the PETROV–GALERKIN AMG scheme [210] is applied for the fluid problem. Coarsening is done until the size of the coarse level problem is below 7500 unknowns per field. The resulting multigrid hierarchies have three levels for the coarser meshes $pw7$ and $pw8$, while four levels are needed for the finest mesh $pw9$. The preconditioner is created once at the beginning of each time step and then re-used in every nonlinear iteration of that time step.

The convergence check is performed as outlined in Section 5.3. For the nonlinear solver, absolute norms of residual vectors and solution vector increments are required to be smaller than 10^{-6} for solid and fluid field, while a tolerance of 10^{-7} is demanded at the interface. The linear solver uses the relative tolerance $\epsilon^{\text{lin}} = 10^{-5}$ in combination with $\beta^{\text{lin}} = 10^{-3}$.

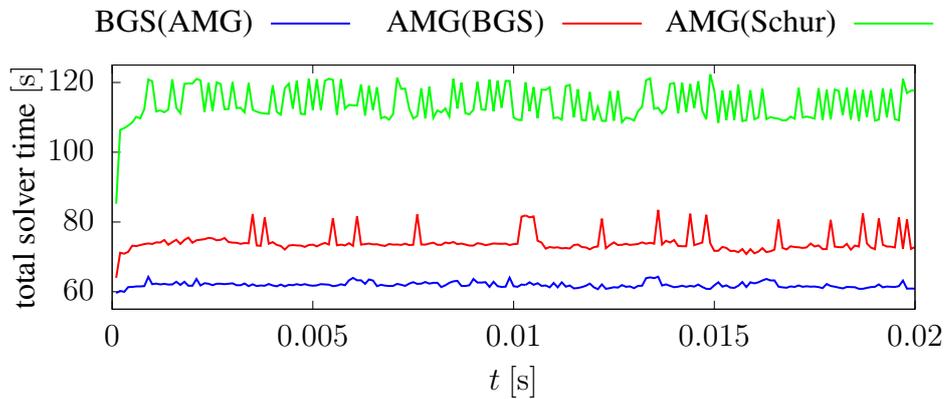
Exemplifying the finest mesh $pw9$ with approximately $11.3 \cdot 10^6$ unknowns, Figure 6.5 reports iteration counts and time measurements associated with the linear solver. Numbers of linear iterations per time step as well as accumulated time measurements for setup of the preconditioner as well as solving the linear system do not vary over simulation time a lot. This corresponds to the very similar dynamic behavior of the problem in every time step without any specific or outstanding events. Considering the three different preconditioning approaches, the BGS(AMG), where the coupling conditions are enforced on the fine level only, needs significantly more iterations than the two other methods, where the interface coupling terms are also included in the coarse level problems, cf. Figure 6.5(a). However, no clear difference in iteration counts can be seen between AMG(BGS) and AMG(Schur). This indicates a similarly strong action of the coarse level couplings despite the actual choice of the level smoothers. However, when looking at setup and iteration timings, the picture is quite different. The BGS(AMG) approach, where only the setup of the field hierarchies needs to be done, is cheaper in terms of setup cost than AMG(BGS) and AMG(Schur), where additional steps to build the coupled coarse level problems are necessary on top of the same setup cost as in BGS(AMG). Assuming an already computed preconditioner and considering solely the time spent in the linear solver, cf. Figure 6.5(b), one observes proportionality between the number of iterations and the time spent for iterations for every preconditioning method. In terms of pure solver time, BGS(AMG) performs similarly as AMG(BGS). The most expensive approach in terms of solver timings is the AMG(Schur) method, which is due to the implementation at hand. When summing up setup



(a) Number of linear iterations per time step



(b) Time for iterating of the linear solver



(c) Time for preconditioner setup and iterating of the linear solver

Figure 6.5: Iteration counts and timings for physics-based block preconditioners applied to mesh $pw9$ of pressure wave example — Iterations, setup and iteration timings are almost constant over all time steps. Iteration counts are lower for preconditioners with FSI coupling on coarse levels. Overall time for the linear solver, cf. Figure 6.5(c), is dominated by the setup cost of the preconditioner.

Table 6.4: Averages of iteration counts and solver time per time step for physics-based block preconditioners in pressure wave example — Iteration counts of the linear solver are not sensitive w.r.t. mesh refinement for all types of preconditioners. A similar, but less strong statement can be made about the time spent in the linear solver, having in mind possible inaccuracies of the time measurements.

(a) Avg. # of linear iterations per time step				(b) Average linear solver time per time step			
Mesh ID	<i>pw7</i>	<i>pw8</i>	<i>pw9</i>	Mesh ID	<i>pw7</i>	<i>pw8</i>	<i>pw9</i>
n^{core}	56	140	448	n^{core}	56	140	448
BGS(AMG)	70.1	65.0	80.3	BGS(AMG)	10.4 s	14.8 s	16.1 s
AMG(BGS)	55.0	48.0	51.0	AMG(BGS)	9.3 s	11.8 s	12.3 s
AMG(Schur)	55.5	48.0	49.0	AMG(Schur)	12.9 s	15.4 s	14.3 s

cost of the preconditioner and time spent in the linear solver, cf. Figure 6.5(c), AMG(Schur) is again the most expensive method due to huge setup cost of the SCHUR complement operator, but also its increased timings for the linear solver, cf. Figure 6.5(b). The setup is expensive since it involves copying large parts of the system matrix, which spoils time measurements. Although iteration counts of the BGS(AMG) are remarkably higher than those of AMG(BGS) and even with the slightly larger time that BGS(AMG) spends in the linear solver, cf. Figure 6.5(b), it is the cheapest approach in total, since it can balance larger iteration counts with the cheapest setup and application cost of all examined preconditioners, cf. Figure 6.5(c).

For mesh *pw9*, iteration counts and timings do not vary much over time. This can also be observed for the coarser meshes *pw7* and *pw8*. Thus, the further discussion uses time-averaged values. Such averaged values for the total number of linear iterations per time step as well as the total time for the linear solve, i.e. the sum of the setup time of the preconditioner and the time for linear iterations, are reported in Table 6.4. For all preconditioners, the number of iterations is constant over mesh refinement as expected for preconditioners based on multigrid techniques. This statement holds for all considered preconditioners. Regarding the time spent in the linear solver, it seems that BGS(AMG) and AMG(BGS) show a slight increase with mesh refinement, while AMG(Schur) does not. However, having only three measurement points is a too small data base to come up with a conclusive statement. Overall, all preconditioners are considered satisfactory in keeping iteration counts and solution time constant with ongoing mesh refinement.

In the research code `Baci` [243], BGS(AMG) and AMG(BGS) are more favorable than AMG(Schur) due to comparable and faster times to solution despite the differences in iteration counts. Performance optimization of the AMG(Schur) implementation is subject to future work.

Novel Hybrid Additive/Multiplicative SCHWARZ Preconditioners for FSI

To assess performance improvements, that can be achieved by the newly proposed hybrid additive/multiplicative SCHWARZ preconditioner, cf. Section 5.2, the new preconditioner is applied

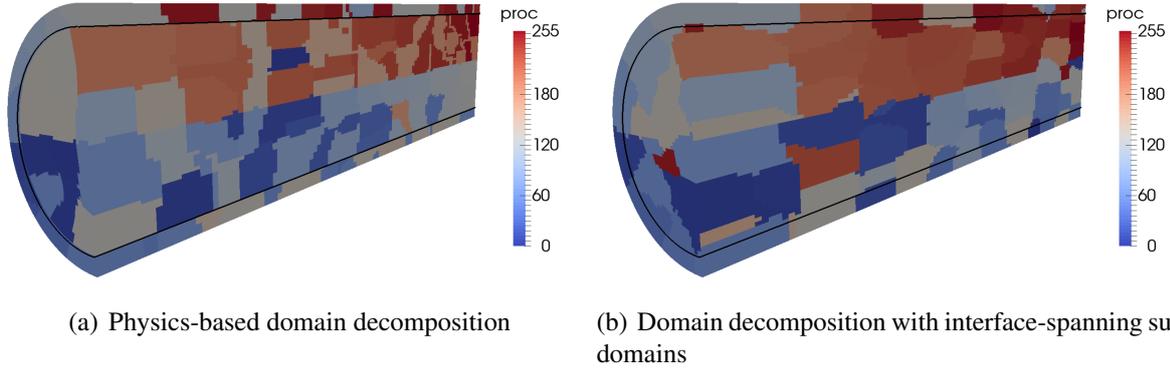


Figure 6.6: Domain decompositions for physics-based and hybrid preconditioners for pressure wave example — The cut view shows the parallel distribution of the entire problem among 256 cores. The position of the interface is indicated by black lines. *Left*: In the initial parallel layout, where solid and fluid domain have been distributed independently, subdomains do not span across the interface. *Right*: After performing the redistribution based on a monolithic graph, some subdomains contain solid or fluid portions only, while only a certain number of subdomains contains solid and fluid portions. The latter span across the interface and, thus, allow for the application of the hybrid additive/multiplicative SCHWARZ preconditioner.

together with the existing preconditioners from Section 5.1 using the RICHARDSON scheme as given in (5.12). The problem is solved on the series of meshes $pw1 - pw6$ as detailed in Table 6.2 to study the influence of mesh refinement and increased core counts. Thereby, the load per core is kept constant, yielding the notion of a *weak scaling* study. The core counts are chosen such that the average load per core is roughly $7620 n^{\text{dof}}/\text{core}$ for each mesh, such that the local subdomains are of a size that can be treated with incomplete LU or LU factorizations very efficiently. This is crucial in case of the hybrid preconditioner. Otherwise, the additional setup cost for the additive SCHWARZ portion could spoil setup times and, thus, render the hybrid approach to be non-competitive at all.

A prerequisite for the application of the hybrid additive/multiplicative SCHWARZ preconditioner is an overlapping domain decomposition with subdomains that span across the fluid-structure interface, cf. Section 5.2.2. Initial and final distributions for mesh $pw6$ are shown in Figure 6.6 for a total core count of $n^{\text{core}} = 256$. Starting from an initial, field-wise distribution as shown in Figure 6.6(a), a *monolithic* graph containing solid and fluid graphs is built. This is passed to the hyper-graph partitioner package *Zoltan* [23] to obtain a parallel layout as it is required. The final parallel layout exhibits subdomains that span across the fluid-structure interface, cf. Figure 6.6(b), and, thus, is able to overcome the concentration of error at the interface in case of the purely physics-based preconditioners.

In this study, the following preconditioner configurations are studied: The additive SCHWARZ part of the combined preconditioner applies an ILU(0) locally on each subdomain. It is applied before and after the AMG preconditioner, that formally is a multiplicative SCHWARZ method. The setup of the physics-based multiplicative SCHWARZ preconditioners is summarized in Table 6.5. Their configurations are similar to the ones from the previous section. The only differ-

Table 6.5: Settings for multigrid hierarchies embedded in the hybrid preconditioner — Solid and ALE use CHEBYSHEV smoothers on the fine and medium levels, while the fluid employs three and six sweeps of damped symmetric GAUSS–SEIDEL (damping parameter $\omega = 0.69$) on the fine and medium levels, respectively. All fields use a direct solver on the coarsest level.

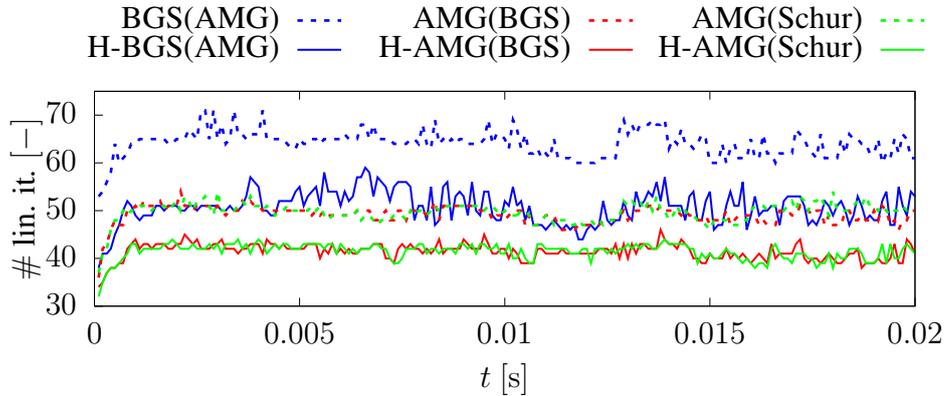
Field	Fine level	Medium levels	Coarse level
\mathcal{S}	CHEBYSHEV(6)	CHEBYSHEV(12)	LU
\mathcal{F}	SGS(3×0.69)	SGS(6×0.69)	LU
\mathcal{G}	CHEBYSHEV(6)	CHEBYSHEV(12)	LU

ence is the application of six sweeps of damped symmetric GAUSS–SEIDEL in the medium levels in the fluid’s AMG hierarchy, since doing ILU in the additive and multiplicative SCHWARZ part of the preconditioner seems to be inefficient due to the higher setup cost. The preconditioner is created once at the beginning of each time step and then re-used in every nonlinear iteration of that time step. Convergence tolerances for the linear and nonlinear solvers are chosen as in the previous example with purely physics-based preconditioning. Only one domain decomposition is computed for each mesh $pw1 - pw6$ and then used for all studied preconditioners to foster comparability of performance results independent of the domain decomposition.

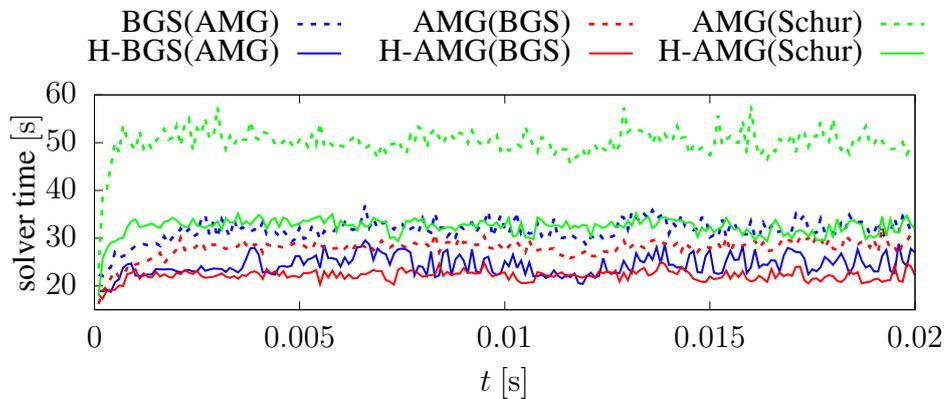
Iteration counts, pure linear solver time, and total linear solver time are reported in Figure 6.7 for the finest mesh $pw6$. Solid lines represent the hybrid additive/multiplicative SCHWARZ preconditioner denoted by the prefix ‘H-’, while dashed lines indicate the classic, purely physics-based block preconditioners for the sake of comparison. The additional additive SCHWARZ preconditioner enhances the preconditioner such that the number of linear iterations is reduced remarkably in every configuration, cf. Figure 6.7(a). A very similar picture can be seen w.r.t. the timings of the linear solver. Looking at the pure solver time, cf. Figure 6.7(b), the reduced number of iterations in case of the hybrid preconditioner results in a significant reduction of the time, that is spent in the linear solver. These massive savings can also be seen in the total timings of the linear solver, that also include the setup cost of the preconditioner, cf. Figure 6.7(c). Since the setup cost of the hybrid preconditioner are larger than of the pure physics-based block preconditioners, the relative savings are lower than in the pure solver time, but still remarkable. These time measurements confirm, that the additional setup cost are amortized by far by the stronger preconditioning effect of the additional additive SCHWARZ step. Although respective diagrams for the coarser meshes $pw1 - pw5$ are omitted for brevity of presentation, it is stressed that they show similar behavior.

Time measurements of the preconditioner setup are not reported separately in Figure 6.7, but rather are included in the total timings, cf. Figure 6.7(c). Due to the additional ILU(0) factorization, that needs to be computed to construct the additive SCHWARZ preconditioner, setup cost are expected to increase in case of the hybrid preconditioner. Due to inaccurate time measurements, the additional cost cannot be determined exactly, but this increase seems to be small since the local subdomain are of small size which allows for a fast computation of the ILU(0) factorization.

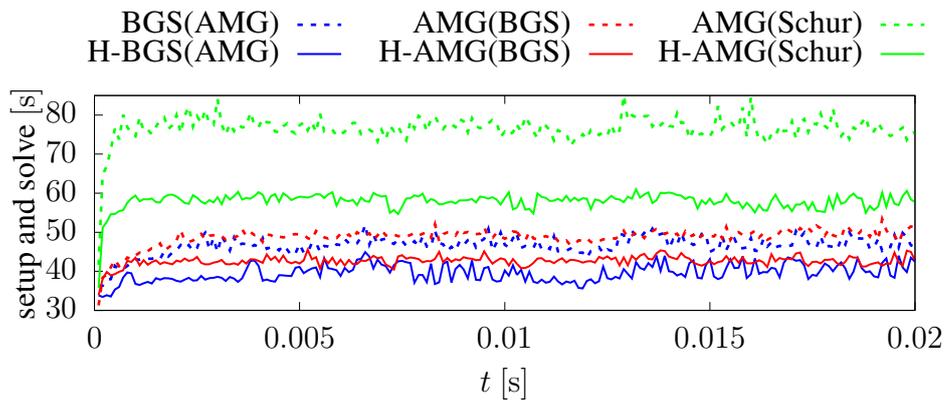
For each mesh $pw1 - pw6$ and all purely physics-based as well as all hybrid preconditioners, the averaged number of GMRES iterations per time step, the average time spent in the linear



(a) Number of linear iterations per time step



(b) Time for iterating of the linear solver



(c) Time for preconditioner setup and iterating of the linear solver

Figure 6.7: Iteration counts and timings for hybrid preconditioners applied to mesh $pw6$ of pressure wave example on 256 cores — Iterations, setup and iteration timings are almost constant over all time steps. Application of the hybrid preconditioner leads to a reduction of iterations and solver time for every type of underlying physics-based block preconditioner.

solver, as well as relative savings due to the hybrid preconditioner are reported in Table 6.6. A comparison among all meshes allows for studying the influence of the mesh size and the core count. Considering the number of GMRES iterations per time step, they remain rather constant with ongoing mesh refinement for all preconditioning approaches as it is expected for multigrid algorithms. Timings of the linear solver exhibit only slight increases, when refining the mesh. When increasing the problem size by a factor of 16, the timings increase only by a factor of 5. This is considered as acceptable, since the problem at hand as well as the implementation are designed to solve engineering problems and not to specifically target optimal solver performance as its only purpose. There are several reasons for these increases: Due to the fully coupled AMG preconditioner, ML's internal rebalancing can not be applied. This leads to a coarse level system, that is far too small to be solved efficiently on a large number of cores and whose solution requires much communication among all processors. Hardware limitations and communication patterns surely contribute to increased timings as well.

To assess its efficiency, iteration counts and solver timings with and without the novel hybrid preconditioner are compared in Table 6.6, where also relative savings of iterations and solver time are reported. These savings are considered as remarkable and totally worth the additional effort during setup of the preconditioner. The extra setup cost is just governed by the size of the subdomains, since the ILU factorization on each core can be performed in parallel independently of each other. If the load per core is kept constant, the additional setup cost is independent of the problem size or the number of processors, respectively.

For the local subdomain solves involved in the additive SCHWARZ part of the hybrid preconditioner, larger fill levels than ILU(0) have been tested. Though, the huge increase in setup cost cannot be amortized by the improved quality of the preconditioner. If the local subdomains are sufficiently small, an exact direct solve seems to be a viable choice, however it is outperformed by the ILU(0) option. Summing up, ILU(0) seems to be a good trade-off between setup cost and effectiveness of the preconditioner. Based on numerical experiments, it is preferred above ILU with higher fill levels and also above direct solves for each subdomain.

6.2 Pulsatile Blood Flow through a Patient-specific Abdominal Aortic Aneurysm

To demonstrate applicability of the proposed formulation and solver framework to complex and patient-specific geometries, the pulsatile blood flow through an *Abdominal Aortic Aneurysm* (AAA) with an outer diameter of 7.5 cm is considered. Model generation starts from *computer tomography* (CT) images and follows the procedure outlined by MAIER *et al.* [159], whereas meshing is done with `Cubit v13.2`. The solid phase consists of the *intraluminal thrombus* (ILT) and the *arterial wall*, while the fluid domain covers the *lumen*. The reconstructed geometry is depicted in Figure 6.8.

The arterial wall utilizes the hyperelastic material model by RAGHAVAN and VORP [196] with constitutive parameters $\alpha_{\text{wall}}^s = 0.174$ MPa, $\beta_{\text{wall}}^s = 1.881$ MPa, and POISSON's ratio $\nu_{\text{wall}}^s = 0.49$. The ILT employs a coupled form of the compressible NEO-HOOKEAN model [118] with the YOUNG's modulus $E_{\text{ilt}}^s = 0.1044$ MPa and POISSON's ratio $\nu_{\text{ilt}}^s = 0.45$. Density of both arterial wall and ILT is chosen as $\rho^s = 1.0$ g/cm³. All values of the solid's constitutive parameters are taken from MAIER *et al.* [159]. Blood uses the modelling assumption of a NEWTONian

Table 6.6: Averages of iteration counts and time measurements per time step for comparison of classic, physics-based block preconditioners and the newly proposed hybrid preconditioner in the pressure wave example — Only one domain decomposition is computed for each mesh $pw1$ – $pw6$ and then used for all preconditioners to avoid spurious influence of the domain decomposition. Iteration counts of the linear solver are not sensitive w.r.t. mesh refinement for all types of preconditioners. Timings of the linear solver exhibit only slight increases with mesh refinement due to the implementation. Relative savings due to the hybrid preconditioner are reported. These savings are considered as remarkable.

(a) Average # of linear iterations per time step						
Mesh ID n^{core}	$pw1$	$pw2$	$pw3$	$pw4$	$pw5$	$pw6$
BGS(AMG)	68.6	65.5	56.6	57.0	59.7	64.1
H-BGS(AMG)	46.1	49.7	42.8	44.8	47.8	50.7
Savings	32.8%	24.1%	24.4%	21.4%	19.9%	20.9%
AMG(BGS)	52.6	54.3	41.9	42.4	44.7	49.1
H-AMG(BGS)	37.0	42.0	33.4	35.0	37.4	41.4
Savings	29.7%	22.7%	20.3%	17.5%	16.3%	15.7%
AMG(Schur)	50.6	53.9	41.7	43.3	45.1	49.7
H-AMG(Schur)	36.0	40.5	33.0	35.0	37.7	41.4
Savings	28.9%	24.9%	20.9%	19.2%	16.4%	16.7%
(b) Average linear solver time per time step						
Mesh ID n^{core}	$pw1$	$pw2$	$pw3$	$pw4$	$pw5$	$pw6$
BGS(AMG)	6.4 s	12.4 s	17.4 s	20.8 s	27.8 s	31.6 s
H-BGS(AMG)	5.7 s	11.0 s	15.5 s	18.2 s	21.1 s	24.4 s
Savings	10.9%	11.3%	10.9%	12.5%	24.1%	22.8%
AMG(BGS)	5.6 s	11.4 s	16.5 s	19.1 s	28.8 s	28.0 s
H-AMG(BGS)	5.0 s	9.6 s	13.8 s	15.8 s	19.4 s	22.2 s
Savings	10.7%	15.8%	16.4%	17.3%	32.6%	20.7%
AMG(Schur)	9.9 s	18.1 s	26.6 s	37.8 s	47.2 s	50.1 s
H-AMG(Schur)	8.2 s	15.8 s	23.0 s	24.9 s	29.8 s	32.4 s
Savings	17.2%	12.7%	13.5%	34.1%	36.9%	35.3%

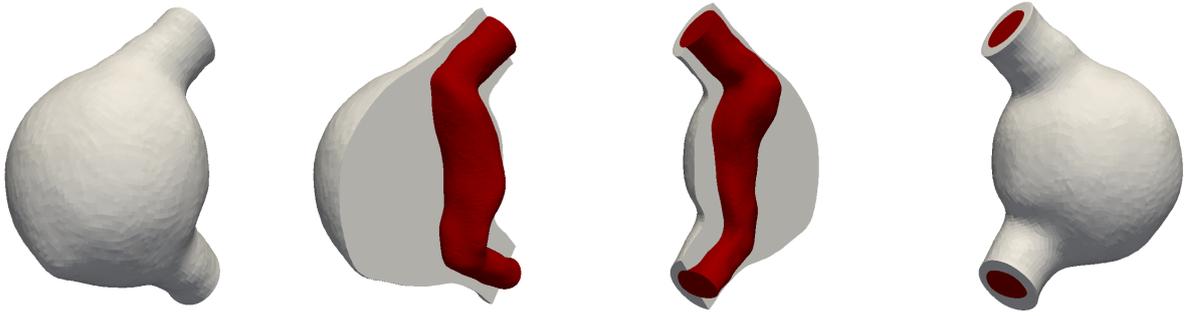


Figure 6.8: Geometry of a patient-specific AAA — The solid phase consists of the *intraluminal thrombus* (ILT) and the *arterial wall*. It is colored in gray. The fluid domain covers the *lumen* and is colored in red. Pulsatile fluid inflow is prescribed at the top, while the outflow boundary is at the bottom.

fluid with dynamic viscosity $\mu_{\text{dyn}}^{\mathcal{F}} = 4 \cdot 10^{-3} \text{ Pa} \cdot \text{s}$ and density $\rho^{\mathcal{F}} = 1.0 \text{ g/cm}^3$ [128]. The ALE mesh motion is determined by a LAPLACEian smoothing procedure, which is suitable since only small deformations are expected.

ILT and arterial wall are clamped at inflow and outflow cross sections. A visco-elastic embedding is used to mimic the surrounding tissue. It is realized with springs and dashpots attached normal to the wall surface in material configuration. Spring stiffness and dashpot viscosity are chosen uniformly as $k_{\text{emb}} = 10 \text{ g}/(\text{mm}^2 \cdot \text{s}^2)$ and $c_{\text{emb}} = 100 \text{ g}/(\text{mm}^2 \cdot \text{s})$, respectively, taken from [169]. At the fluid inflow area $A_{\text{in}}^{\mathcal{F}} = 267,018 \text{ mm}^2$, a time-dependent parabolic velocity profile is prescribed. The time curve is based on measurement data [225] and results in a flow rate $Q_{\text{in}} = 3.01/\text{min}$, which seems to be in the physiological range. It is depicted in Figure 6.9. At the beginning of the heart cycle, the inflow velocity is immediately increased to its peak value at time $t \approx 0.09 \text{ s}$. In this short period of time, almost all transport of blood volume happens. This peak is followed by a brief plateau at low inflow velocities. Then, the inflow direction is even reverted during a short period of prescribed backflow with its peak at time $t \approx 0.37 \text{ s}$. Starting with approximately $t > 0.6 \text{ s}$ only very low and constant inflow velocities are prescribed. Although the downstream portion of the vascular system is not resolved geometrically, it is included into the model via a traction boundary condition at the fluid outflow area. The prescribed time-dependent traction value varies between the diastolic level $p_{\text{dia}}^{\mathcal{F}} = 86.9 \text{ mmHg}$ and the systolic level $p_{\text{sys}}^{\mathcal{F}} = 121 \text{ mmHg}$. Its time curve is based on results by ISMAIL *et al.* [128] and is also depicted in Figure 6.9. Its peak plateau is in the range of approximately $0.15 \text{ s} \leq t \leq 0.31 \text{ s}$, i.e. it is slightly delayed to the velocity curve. Its peak value is located at $t = 0.18 \text{ s}$. An even more accurate modelling can be achieved by the application of patient-specifically calibrated *Windkessel* models [128], but limitations in the implementation of the computational framework prevented their application in combination with the adaptive time stepping scheme. Both velocity and pressure curves are periodic in time with a cycle time of $T_c = 1.0 \text{ s}$.

To account for the fact that the stack of CT images have been taken *in vivo*, i.e. in the presence of blood pressure, a prestressing phase up to the diastolic blood pressure $p_{\text{dia}}^{\mathcal{F}}$ using a *modified updated LAGRANGEan formulation (MULF)* proposed by GEE *et al.* [88] is prefixed to the actual transient simulation. During this prestressing phase, the fluid inflow and outflow boundary

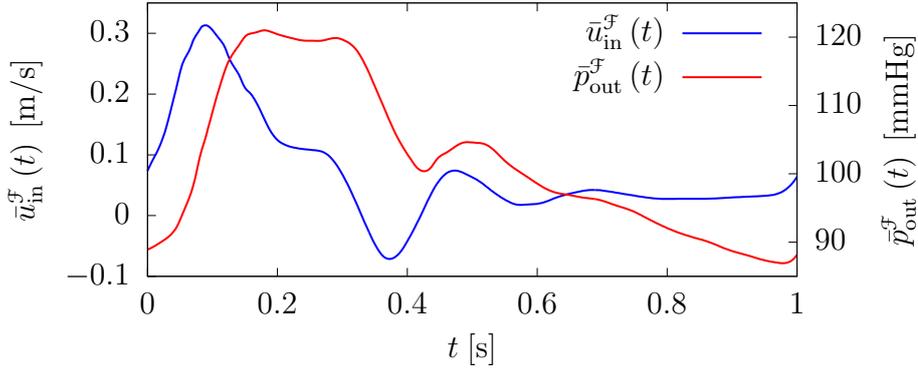


Figure 6.9: Time curves for physiological fluid boundary conditions of AAA— The inflow velocity time curve $\bar{u}_{\text{in}}^{\mathcal{F}}(t)$ is based on measurement results published in [225] and on the computational example in [128]. The temporal evolution $\bar{p}_{\text{out}}^{\mathcal{F}}(t)$ of the pressure level at the outflow boundary is analog to numerical results in [128]. Both curves are periodic in time with a cycle time of $T_c = 1.0$ s.

conditions are smoothly increased to their initial values at the beginning of the heart cycle, while the solid is treated with the MULF algorithm.

Meshing is performed as follows: Starting with a unstructured quadrilateral mesh of the fluid-structure interface of the lumen, a purely hexahedral boundary layer for the fluid is created. The remaining fluid domain is filled with tetrahedral elements, while a single layer of pyramid elements is used at the transition to the hexahedral boundary layer. The fluid’s interface mesh is then imprinted on the lumen’s FSI surface. Additionally, the lumen’s outer surface is meshed with unstructured quadrilaterals. As in the fluid domain, the bulk of the ILT is meshed with tetrahedra while pyramids are inserted at the transition to the quadrilateral surface meshes. Finally, the arterial wall is extruded using linear hexahedral elements. The mesh is depicted in Figure 6.10. It consists of 211827 solid, 328548 fluid, and 246411 ALE degrees of freedom, yielding a total number of unknowns of $n_{\text{total}}^{\text{dof}} = 786786$. The problem is ran on 28 cores of SuperMUC’s *Phase 2* partition, cf. Appendix B.

Time integration is performed by means of the generalized- α method with spectral radii $\rho_{\infty}^{\mathcal{S}} = 0.8$ and $\rho_{\infty}^{\mathcal{F}} = 0.5$ in solid and fluid field, respectively. Adaptive time stepping is applied based on error estimation in the fluid field only, using the comparison to an auxiliary ADAMS–BASHFORTH-2 scheme. The solid is not taken into account for error estimation since only very little deformation is expected. Moreover, dynamics of the problem are solely driven by the fluid flow. The local error tolerance is related to the peak fluid inflow velocity as $\varepsilon_l^{\mathcal{F}, \Delta t} = 2.0 \approx 0.5\% \cdot \hat{u}_{\text{in}}^{\mathcal{F}}$ as suggested in Section 4.2.3. The time step size is limited to the range $\Delta t_{\text{min}} = 10^{-5} \text{ s} \leq \Delta t_n \leq 0.1 \text{ s} = \Delta t_{\text{max}}$. The remaining algorithmic parameters related to adaptive time stepping have been chosen as $\kappa_{\text{min}} = 0.1$, $\kappa_{\text{max}} = 2.0$, and $\kappa_s = 0.9$. Averaging of increasing time step sizes (4.33) is applied using $\gamma_{n+1} = 0.3$ and $\gamma_n = 0.7$. The number of time step repetitions is limited to five.

The fully-coupled AMG-based FSI preconditioner detailed in Section 5.1.3 is applied. Settings are summarized in Table 6.7. A three-level multigrid hierarchy is built, where SA-AMG is

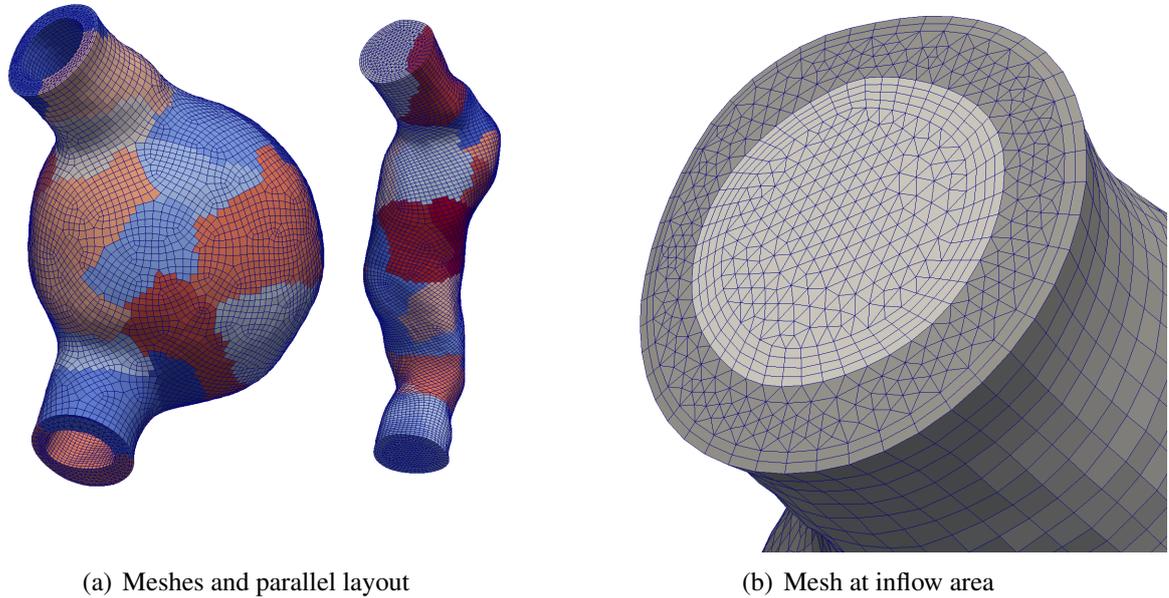


Figure 6.10: Mesh of a patient-specific AAA — *Left*: Meshes of solid and fluid domain are shown. Colors indicate the domain decomposition among 28 processors. *Right*: The close-up of the mesh at the inflow area shows the tetrahedra-based solid mesh, where the arterial wall is extruded as pure hexahedral mesh (dark grey). The fluid features a purely hexahedral boundary layer. Its interior is meshed with tetrahedral elements (light grey).

Table 6.7: Settings for fully-coupled AMG-based preconditioner for solution of AAA — Solid and ALE use cheap CHEBYSHEV smoothers on fine and medium level, while the fluid employs three sweeps of damped symmetric GAUSS–SEIDEL on the fine level and incomplete LU with fill level 0 on intermediate levels. All fields use a direct solver on the coarsest level. On each level, a BGS smoother is applied.

Field	Level 0	Level 1	Level 3
\mathcal{S}	Chebyshev(6)	Chebyshev(12)	LU
\mathcal{F}	SGS(3×0.69)	ILU(0)	LU
\mathcal{G}	Chebyshev(6)	Chebyshev(12)	LU

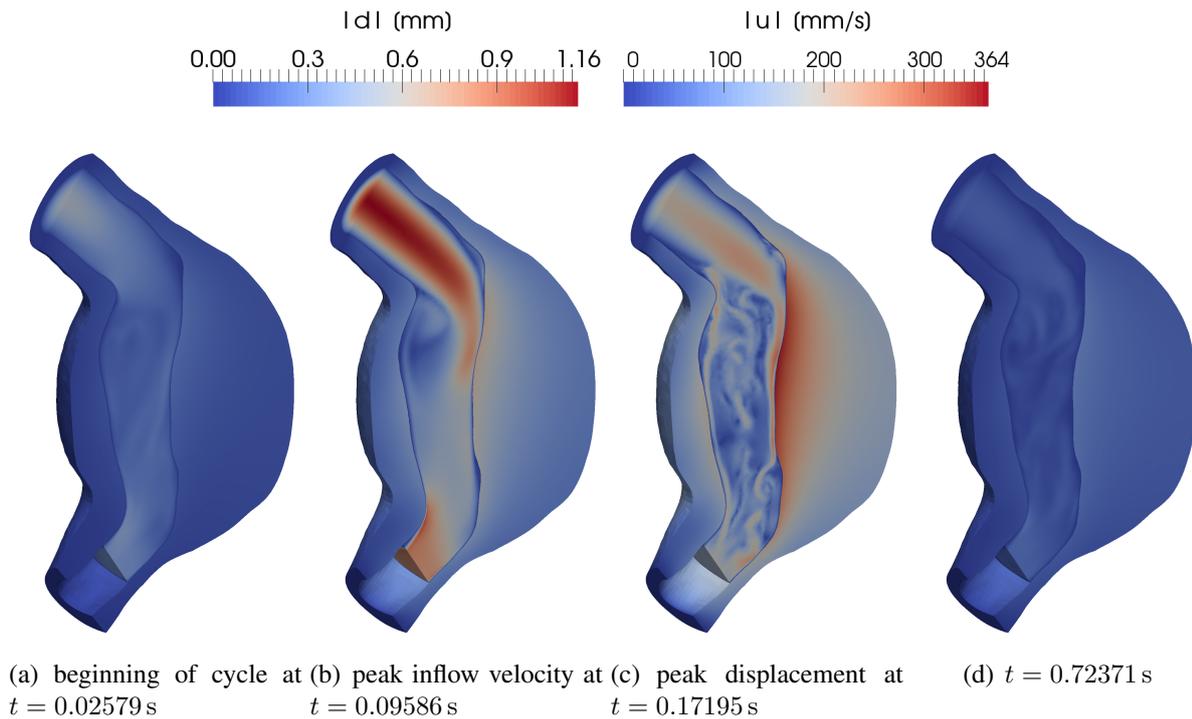


Figure 6.11: Snapshots of the solution of a pulsatile blood flow through an AAA — The fluid field is colored according to the magnitude of its velocity field denoted by $|u|$ while the structure field's color is encoded with the displacement field magnitude denoted by $|d|$. The fluid color scale is calibrated at maximum inflow velocity at $t = 0.09586$ s, while the solid color scale is based on maximum deformation at $t = 0.17195$ s. The pulsatile motion of ILT and arterial wall is further detailed in Figure 6.12.

used for solid and ALE, while a PG-AMG is applied to the fluid. For the linear solver, a relative tolerance of $\varepsilon^{\text{lin}} = 10^{-5}$ in combination with $\beta^{\text{lin}} = 10^{-3}$ is required, while the nonlinear iterative solver is considered as converged as soon as field-related residual and solution increment norms are below 10^{-6} and those related to the interface below 10^{-7} .

Figure 6.11 shows cuts through the AAA to illustrate flow patterns and internal deformation. The lumen is cut prior to the outlet to allow for a visualization of the displacement of the interior boundary of the ILT, i.e. the FSI interface. Starting from an initial state with only little flow and almost no deformation, cf. Figure 6.11(a), the state with the peak inflow velocity is reached quickly at time $t = 0.09586$ s, cf. Figure 6.11(b). The prescribed inflow can be seen clearly, which is then deflected by the ILT. Due to a larger diameter of the lumen inside the AAA, velocities are lower there and increase only towards the narrowed outlet. The maximum deformation of ILT and arterial wall is reached at time $t = 0.17195$ s, cf. Figure 6.11(c), where the largest displacements occur directly at the fluid-structure interface. However, deformation of the wall is visible as well, even if it takes smaller values than in the ILT due to its higher stiffness. Towards the end of the heart cycle, as only a low and almost constant inflow velocity is prescribed, fluid flow in the domain almost vanishes, cf. Figure 6.11(d). Additionally, only

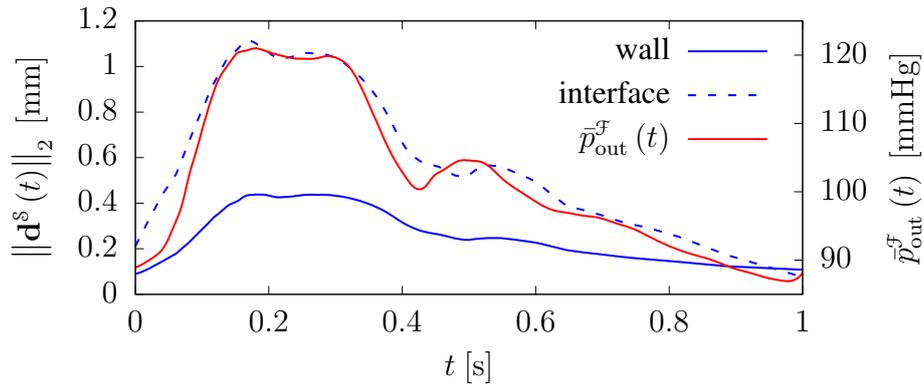


Figure 6.12: Displacement of the artery throughout the heart beat — The absolute displacement of two exemplary solid nodes, one located on the outer surface of the arterial wall and the other on the fluid-structure interface, is governed by the fluid pressure p^f , that in turn follows the prescribed pressure $\bar{p}_{\text{out}}^f(t)$ at the outflow cross section. The influence of the fluid velocity field on the pulsatile motion of the ILT and the arterial wall seems to be rather small.

small displacements occur due to the fluid pressure being closer to the diastolic than the systolic pressure level. Overall, a periodic state has been reached.

The pulsatile motion of ILT and arterial wall can be seen in Figure 6.12, where the absolute displacements of two exemplary solid nodes are related to the prescribed pressure at the outflow cross section. Both nodes are picked at about half length of the aneurysm in regions, where large deformations occur. One is located on the fluid-structure interface and the other one on the outer surface of the arterial wall. Temporal evolution of their displacement magnitudes follows the time curve of the prescribed outflow pressure $\bar{p}_{\text{out}}^f(t)$. Solid deformation seems to be governed by the fluid's pressure rather than by the fluid's velocity field. This fact stresses the importance of the applied prestressing technique, that is crucial to obtain physiologically correct pressure levels.

The evolution of the time step size Δt_n is reported in Figure 6.13. A variation of the time step size can be clearly observed. At the beginning of the heart cycle, the time step size is reduced due to the transient behavior stemming from the peak of the inflow velocity. Towards the end of the heart cycle, i.e. $t > 0.6$ s, a low and almost constant inflow velocity is prescribed. Additionally, dynamic behavior from the the initial velocity peak already vanished. Thus, in increase of the time step size by roughly an order of magnitude is possible. Repetition of time steps is necessary only for the last five time steps of the heart cycle where the inflow velocity already increases a little bit, cf. Figure 6.9. All other reductions of Δt_n can be handled by the safety factor κ_s without the need of time step repetitions. As the beating motion of the heart and, thus, the pulsatile blood flow exhibit periodic behavior, such a periodicity can also be observed for the evolution of the time step size.

To assess computational efficiency, computational savings compared to the case with constant time step size and the same accuracy requirements are quantified using (4.35). Taking into account the smallest time step size used during the computation, $\Delta t_n = 1.5873 \cdot 10^{-4}$ s, and

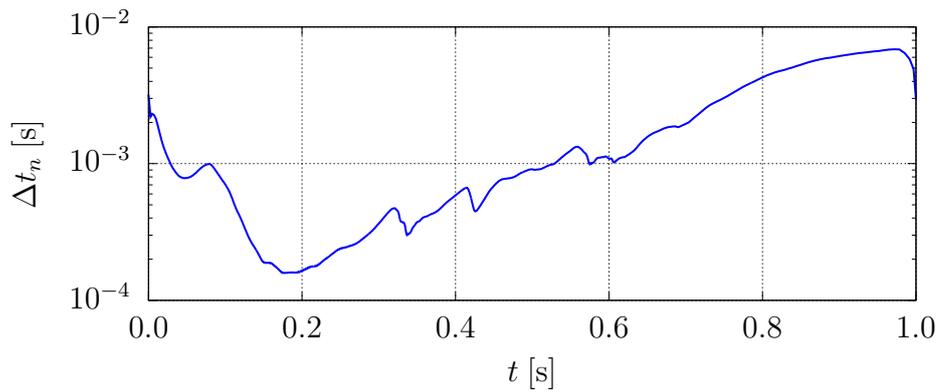


Figure 6.13: Evolution of time step size Δt_n for pulsatile flow through AAA — After reduction of Δt_n during the initial phase of the heart cycle with high inflow velocities and even a prescribed backflow, the time step size is increased towards the end of the heart cycle, where only low inflow velocities are prescribed and dynamics already calmed down.

the number of adaptive time steps $n_{\text{ada}} = 1616$ needed for one heart cycle, the computational savings evaluate to 74.5% which is a tremendous gain in efficiency.

7 Concluding Remarks and Outlook

Achievements of this thesis are summarized briefly before an outlook on possible future research directions is given.

Summary of Achievements

In this thesis, a monolithic framework for the solution of fluid-structure interaction problems has been described. It allows for the numerical simulation of the interaction of an incompressible fluid flow with solid bodies undergoing finite deformation. To deal with the deformation of the fluid domain, an ALE observer has been used for the fluid field, while the solid domain has been treated with a purely LAGRANGEan description. At the fluid-structure interface, kinematic continuity as well as dynamic equilibrium of interface traction fields are required. The coupling conditions have been enforced weakly via a LAGRANGE multiplier field.

Spatial discretization of the solid and the fluid field has been performed by means of the finite element method. The meshes of the solid and the fluid field do not need to match at the interface since a dual mortar method has been used for the discretization of the LAGRANGE multiplier field. Due to the dual shape functions, a static condensation of the LAGRANGE multiplier degrees of freedom could be performed. This led to two variants of coupling algorithms, namely one with a fluid-handled interface motion and another one with a structure-handled interface motion. For both cases, the final systems of equations have been derived and presented in full detail.

For time integration, fully implicit, single-step, and single-stage time integration schemes are applied in both fields. The basic goal of choosing the solid and fluid time integration scheme freely and independently from each other to allow for tailoring to the field-specific needs has been achieved. For this purpose, a temporal interpolation of the interface traction fields has been incorporated in a temporally consistent manner. The artificial energy production at the FSI interface due to different time integration schemes in the solid and the fluid field has been studied theoretically and numerically. This energy production is shown to vanish for time step size refinement, i.e. $\Delta t \rightarrow 0$, or for certain choices of time integration schemes, namely $t_m^{\mathcal{F}} = t_m^{\mathcal{S}}$. Numerical examples have been presented, that demonstrate optimal temporal convergence rates, study this freedom of choice of time integration schemes, and show the negligibility of the artificial energy production at the FSI interface in practical simulations.

To control the accuracy of the time integration of the coupled problem, a novel adaptive time stepping scheme for monolithic FSI solvers has been proposed. It is based on *a posteriori* error estimation. Quantities from the solid and the fluid field as well as from the fluid-structure interface are taken into account to adapt the time step size to the current needs of the simulation. Overall, a user-given level of accuracy can be guaranteed. The adaptive time stepping scheme has been applied to several academic examples but also to the simulation of the pulsatile blood

flow through a patient-specific abdominal aortic aneurysm. Guaranteed accuracy as well as remarkable gains in efficiency have been demonstrated.

Starting from existing physics-based AMG block preconditioners for FSI problems, a novel hybrid additive/multiplicative SCHWARZ preconditioner has been developed. It combines the powerful multigrid performance of the existing physics-based preconditioners with an additional additive SCHWARZ preconditioner, that is specifically designed to tackle error accumulation at the fluid-structure interface. This was achieved by generating an overlapping domain decomposition with subdomains that span across the interface. Incomplete LU factorizations that are insensitive to the block structure of the matrix stemming from the different physical fields have been found to be a viable choice as subdomain solvers. They seem to represent a good trade-off between setup cost and quality of the result.

A thorough comparison of the presented preconditioning techniques has been performed. The influence of the problem size has been studied. Summing up, the application of the novel hybrid additive/multiplicative SCHWARZ preconditioner results in significant reductions of the number of linear iterations as well as the time that is spent in the linear solver. These savings have been shown to be independent of the mesh size and the number of cores used for the computation. Furthermore, a pulsatile blood flow through a patient-specific abdominal aortic aneurysm has been simulated. Thereby, the adaptive time stepping as well as FSI-specific AMG-based preconditioners have been applied. Overall, applicability of the proposed monolithic solver with all its features to real-world problems could be demonstrated. Even in such complex problems, a speed-up by a factor of 4 was achieved.

Outlook and Future Work

In future work, several aspects could be studied: On the one hand, questions of the handling of non-matching interface discretizations could be addressed. In the current approach with the dual mortar method, the evaluation of the mortar integrals is very expensive. This is not a drawback in the present work as the mortar matrices need to be evaluated only once in the beginning of the simulation and then are stored and used throughout the entire computation. However, this works only for problems with large but still moderate distortion of the fluid mesh. Especially in applications where the solid body performs a rotatory motion inside the fluid domain, mesh distortion often becomes very large. To prevent the breakdown of the simulation, a relaxation of the ALE mesh is necessary. This can be realized by releasing the mesh interface coupling in tangential direction as proposed by KLÖPPEL [137], but at the additional cost of re-evaluating the mortar coupling matrices. Speed-up can be achieved by fast integration techniques as discussed by FARAH *et al.* [68], but this does not cure the original problem. Alternatively, one could employ a method whose coupling operators can be evaluated very fast and cheaply. The INTERNODES approach by DEPARIS *et al.* [52, 55] could be interesting in that regard. Although it requires the evaluation of the coupling operators in the spatial configuration, their setup is very fast since only mass matrix-like operators need to be integrated on the interface in the spatial configuration. A third remedy is described in the thesis by SHAHMIRI [218] where a hybrid fixed-grid/ALE approach is employed. To allow for arbitrary movement of the solid body, a thin layer of ALE-based fluid is wrapped around the solid body such that the FSI problem can be treated with the same methods described in the present thesis. However, this ALE-based fluid

is then moved over the EULERian background fluid. Hence, the mortar matrices do not change throughout the simulation, but additional costs and difficulties arise from the fluid-internal interface. Applications that can benefit from such approaches range from a very small scale like in the numerical analysis of red blood cells [137] to the very large scale like simulations of wind turbines [13, 14].

With regard to the adaptive time stepping scheme, two extensions are of interest. First, an incorporation of more sophisticated and hopefully more accurate error estimators could increase accuracy even further. Additionally, the concept of an 'error density per unit time', that has been briefly mentioned in section 4.2.3, is worth further investigation.

The newly proposed hybrid additive/multiplicative SCHWARZ preconditioner produced satisfactory results and speed-ups of the computation. However, some details of that approach should be investigated further. In the current approach, the additional additive SCHWARZ preconditioner is applied before and after the physics-based block preconditioner, cf. (5.11). By applying it only either before or after the physics-based block preconditioner, the computational time for a single application of the additive SCHWARZ preconditioner could be saved. However, it is not clear *a priori* how this modification influences the quality of the preconditioner. An alternative modification would be to keep the order of applications as given in (5.11), but to reduce or even drop the amount of pre- and/or post-smoothing on the AMG hierarchy's fine level of the physics-based block preconditioner. Both modifications need to be evaluated by comparing savings in application time to a possible increase of numbers of linear iterations due to a reduced effect of the preconditioner. Especially the variant that omits the application of the additive SCHWARZ preconditioner after the physics-based block preconditioner might lead to additional overall savings. Looking at the implementation in `Baci`, the issue of single cores being idle during some stages of the computation needs to be addressed as indicated in Remark 5.2.2.

Finally, the overall nesting of all components of the solution process could be changed. Currently, the nonlinearity is treated by a NEWTON scheme, that employs a KRYLOV solver in every nonlinear iteration, which in turn uses linear multigrid methods for its preconditioning. With such a setup, it is well known that the simulation spends most of its wall clock time inside the linear solver, but requires only a few residual evaluations per time step. Having in mind recent trends in hardware architecture, namely fine-grained parallelism with CPUs connected to many GPU-like cores, evaluations of the residual vector can be sped up a lot due to their perfect parallel scalability. To exploit this fact in the solution of a nonlinear problem, the nesting as described above could be flipped as follows: A nonlinear multigrid method can be put on the outside to address the nonlinear character of the problem. The NEWTON scheme, an approximate variant of it, or other nonlinear schemes can be used as level smoothers inside the nonlinear multigrid scheme. Additionally, the outer multigrid scheme can be embedded into a nonlinear KRYLOV accelerator like *nonlinear GMRES* [48, 179, 246] or *ANDERSON acceleration* [1] to improve convergence. By using any other method than NEWTON's method to deal with nonlinear problems, the number of iterations of the nonlinear solver and, thus, the number of residual evaluations is expected to increase or even to increase a lot. However, exactly these residual evaluations can be done extremely fast. Combined with the fact, that the effort of solving large systems of linear equations might be reduced by such an approach, huge overall savings might be possible. BRUNE *et al.* [30] show results of such an approach applied to pure elasticity or fluid flow problems, but using them in the setting of fluid-structure interaction problems seems to be a novel approach.

A Mesh Motion Algorithms for the ALE Grid

As an ALE observer is used to describe the flow field on a deforming domain, evidently a strategy for moving the nodes of the ALE mesh is needed. In a FSI context, its basic task is to extend the interface motion, i.e. interface displacement and velocity, into the interior of the fluid domain, such that grid displacements and grid velocities at interior nodes can be computed purely based on prescribed boundary information as given by (2.1). The problem of ALE mesh motion is a purely artificial problem. It is not related to the physics of the coupled problem, i.e. it does not influence the coupling itself. Its sole purpose is to define a proper mesh for the evaluation of the fluid problem.

A brief, but detailed and comprehensive overview on ALE methods is given by DONEA *et al.* [62]. ALE methods are most often used in the context of finite volume or finite element based spatial discretizations. An application to meshless methods is described in [183], for example.

Although in this thesis the ALE method is used to describe the fluid field on a deforming domain, it can also be applied to other types of problems. In solid mechanics, ALE methods are very appealing in scenarios with very large displacements and distortions like fracture, crash, metal forming, impact, and explosion, where a Lagrangian description usually fails [62]. Especially problems including plasticity are often treated with an ALE description, e.g. [2, 4, 5, 84–86, 93, 120, 122, 156, 157, 201, 202, 215].

In the following, some ALE mesh motion techniques are summarized briefly. First, some basic requirements are discussed. Afterwards, an overview of possible mesh moving algorithms is given, where those three that are used throughout this thesis are described briefly. Finally, a large deformation benchmark test case is proposed and used to compare and evaluate these ALE mesh motion algorithms.

A.1 Requirements

First, some desired properties are discussed which will later be used to assess the quality and the efficiency of mesh motion algorithms. This collection of requirements is based on comments in [62, 242]. The ALE field is a purely artificial and auxiliary field, that is just used to enable fluid computations on a deforming domain, but does not alter the physical problem at hand. Its influence on the computational workflow should be small, i.e. mesh motion should be a cheap and automatic process during the simulation. In addition, remeshing should be circumvented for two reasons. On the one hand, it would introduce additional errors due to data transfer from the old to the new mesh. On the other hand, remeshing usually requires user interaction, that should be avoided. Further requirements arise since the quality of the fluid field solution crucially depends on the grid displacement and velocity fields. Even in the presence of large

deformation, one aims at avoiding excessive element distortion, which at least degrades the quality of the fluid solution, but also can lead to the failure of the simulation, if distortions result in invalid element shapes or deformations like inverted or self-penetrating elements. Moreover, a spatially smooth grid velocity field \mathbf{u}^g is necessary since it directly enters the convective term in the NAVIER–STOKES equations (2.4). Finally, certain features of the mesh like boundary layers or local refinements should be preserved throughout the simulation.

A.2 Mesh Motion Algorithms

Especially in the presence of large deformations, a robust mesh motion algorithm is crucial in practical applications. In this section, some mesh motion techniques are reviewed very briefly. A literature survey on ALE mesh update procedures is given by DONEA *et al.* [62]. They distinguish two basic mesh update techniques. The category of *mesh regularization* comprises all techniques that aim at preserving mesh regularity and avoid mesh entanglement during the computation. *Mesh adaptation techniques* usually involve some notion of error indication to allow, for instance, to concentrate a finer mesh in regions of steep solution gradients. When thinking in these two categories, the concept of mesh regularization is applied in this thesis exclusively.

A further classification has been done in the thesis by WALL [242], where a basic distinction into *continuous* and *discontinuous* mesh motion algorithms is made. On the one hand, continuous methods usually deform huge portions of the mesh. If mesh deformation can roughly be estimated *a priori*, *heuristic* approaches like in [60] or *interpolation methods* like in [6, 94, 97, 114, 136, 153, 213, 229] can be applied. For more complex problems, general approaches might be necessary that can be seen as mesh regularization techniques in the sense of DONEA *et al.* [62]. On the other hand, discontinuous approaches are often used, when large deformations occur, but are limited to simple geometric features of the domain like e.g. rotating machinery parts in a turbine. These approaches allow for relative motion between two meshes and enable data exchange at an interface in case of surface coupling or in a region of overlap in case of volume coupling. A mortar-based approach for such problems of surface-coupled type has been developed by KLÖPPEL [137] for the simulation of human red blood cells. Referring to the classification by WALL [242], solely continuous mesh motion schemes are used in this thesis.

Three different mesh regularization strategies are applied, namely LAPLACEian smoothing, a springs model and an elasticity approach which are briefly described in the sequel. A thorough study of available implementations in the research code `Baci` [243] has been performed by BRUDER [29].

A.2.1 LAPLACEian Smoothing

Positioning of nodes by solving a LAPLACE equation is a very popular method for ALE mesh motion. It has originally been proposed by WINSLOW [252]. LÖHNER and YANG [158] distinguish between coordinate and velocity smoothing techniques with nodal displacements or velocities as the primary unknowns. In this thesis, only smoothing of the nodal displacements is pursued, whereas the grid velocity \mathbf{u}^g is reconstructed afterwards by the means of finite differences as also done by FÖRSTER [81].

To avoid steep gradients and to arrive at smooth mesh displacement or velocity fields, the LAPLACE equation

$$\nabla_0 \cdot \kappa^g \nabla_0 \underline{\mathbf{d}}^g = \underline{\mathbf{0}} \quad (\text{A.1})$$

with essential boundary conditions

$$\underline{\mathbf{d}}^g = \bar{\underline{\mathbf{d}}}^g \quad \text{on } \Gamma_D^g$$

is solved. Note that the mesh motion problem is a pure DIRICHLET problem, i.e. $\Gamma_D^g = \partial\Omega_0^g$. A finite element formulation can be derived by applying the standard procedures as they have already been outlined in Section 3.2. After stating a weak form, the spatial discretization (3.11a) is inserted and the finite element formulation can be developed. Due to the linearity of (A.1), the stiffness matrix needs to be evaluated only once in the beginning of the simulation. This makes such an approach very appealing from a computational point of view.

However, the simplicity of the LAPLACEian smoothing comes along with a reduced robustness in certain scenarios. For example, meshes usually feature small elements in the vicinity of interfaces or boundaries, e.g. boundary layers, or other types of refinement in special areas of interest. Moreover, these areas often undergo large deformations and, thus, these small elements are endangered to reach inadmissible deformation first. By allowing for a spatially varying diffusivity κ^g , one can shift the deformation away from the small elements towards the larger ones that are far away from the interface or boundary. This greatly enhances robustness of the mesh motion in presence of large deformation. Different strategies have been proposed in literature. LÖHNER and YANG [158] use a larger κ^g close to the interface or boundary and decrease its value with increasing distance. Alternatively, the diffusivity κ^g can be chosen to scale inversely with the element size or element Jacobian. However, in this thesis a constant diffusivity κ^g is chosen throughout all computations. If mesh motion by LAPLACEian smoothing fails, the more powerful elasticity model is used, cf. Section A.2.3. In addition, the LAPLACEian smoothing has one major drawback, namely its restriction to domains [62]. If it is applied to non domains, interior nodes might be moved outside the domain, yielding an inadmissible mesh. However even for domains, the mesh might undergo extensive local squeezing. Overall, moving the mesh using a LAPLACEian smoother has certain advantages w.r.t. simplicity and efficiency, but its practical applicability is limited especially for complex domains and large deformations. Thus, in this thesis LAPLACEian smoothing is applied only just for scenarios with small or moderate or uniform mesh motion, that do not exhibit large amounts of shear.

A.2.2 Springs Model

Another class of methods interprets the finite element mesh as a network of linear springs instead of continuum elements. This spring analogy has been originally proposed by BATINA [11] for two-dimensional airfoil simulations on purely triangular unstructured meshes, where the vertices of the computational grid are connected with rectilinear springs. Although this approach prevents collision of nodes, it suffers from possible interpenetration of neighboring elements, which can be overcome by additional torsional springs as proposed by FARHAT *et al.* [71]. Extensions to three dimensions have been done in [49, 259]. A graphical representation of the two-dimensional case is shown in Figure A.1.

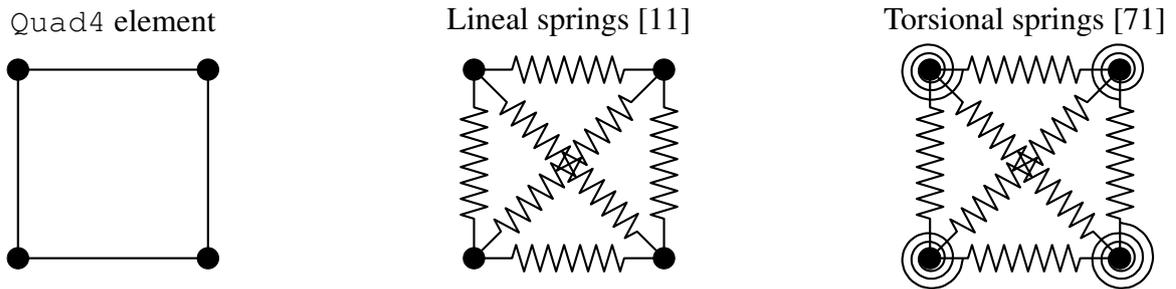


Figure A.1: Springs analogy in two dimensions for ALE mesh motion — All nodes of a finite element (*left*) are connected with rectilinear springs (*middle*). Additionally, torsional springs (*right*) can be placed at each vertex to increase robustness in presence of large deformations.

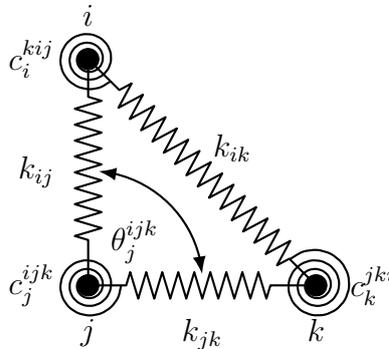


Figure A.2: Configuration of rectilinear and torsional springs in an exemplary triangular element — The vertices i , j , and k are connected with rectilinear springs. At each vertex, a torsional spring is attached, acting on the angle between the two edges sharing that specific vertex.

In the rectilinear springs model, nodes are connected with linear springs. The spring stiffnesses k_{ij} are chosen inversely proportional to the length ℓ_{ij} of the supporting edge between the vertices i and j [11]. Following the improvements by FARHAT *et al.* [71], a torsional spring is attached at vertex j between the two edges, that connect the vertices i , j , and k while sharing vertex j , see Figure A.2. Its stiffness is denoted by c_j^{ijk} with the subscript j specifying the node, that the torsional spring is attached to, and the superscript ijk indicating the angle between the edges ij and jk , that is used to calculate the torque of this spring [71], cf. Figure A.2. The torsional stiffness tends to infinity if the angle θ_j^{ijk} approaches its bounds 0 or π . This helps to prevent interpenetration of neighboring elements [71] which enhances the robustness of the mesh moving algorithm in presence of large deformations. A numerical solution algorithm is sketched by FARHAT *et al.* [71].

Its simplicity and its broad range of applicability make the spring model very appealing in many scenarios and, thus, it is often used in literature. However, it might result in distorted meshes, that are not acceptable for the computation of accurate fluid field solutions, especially in the presence of large deformations, cf. BRUDER [29].

A.2.3 Elasticity Model

The most expensive, but also most robust approach for mesh motion is an elasticity model. Therefore, a reduced static version of (2.8) is used, reading

$$\nabla_0 \cdot \underline{\underline{\mathbf{P}}} = \underline{\underline{\mathbf{0}}} \quad \text{in } \Omega_0^g \quad (\text{A.2})$$

with the first PIOLA-KIRCHHOFF stress tensor as defined in (2.9). All remaining quantities needed to evaluate (A.2) are defined as in the solid case, cf. Section (2.2).

In principle, a variety of strain energy functions Ψ can be used to model the constitutive behavior. Due to their availability in the research code `Baci` [243], BRUDER [29] studied different nonlinear hyperelastic materials, namely ST.-VENANT-KIRCHHOFF (SVK), MOONEY-RIVLIN (MR), NEO-HOOKE (NH), and *logarithmic* NEO-HOOKE (*logNH*), cf. [24, 118]. The MR material includes constitutive parameters c_1^g , c_2^g , and c_3^g with a penalty parameter-like notion to enforce incompressibility. Despite the SVK material's capabilities for large displacements and rigid body motions, it is said to be not suitable for large distortions. The NH material contains terms that penalize cases where the determinant of the Jacobian becomes small. Its logarithmic variant reacts even stronger when subjected to shear. The latter three can be parametrized using a YOUNG's modulus E^g and a POISSON's ratio ν^g . For further details on these material laws, see e.g. the textbooks by BONET and WOOD [24] or HOLZAPFEL [118]. A detailed numerical analysis of the evolution of the strain energy functions in case of ALE mesh motion applications has been performed by BRUDER [29].

In very demanding cases, when the stiffening effect of a constitutive law is not sufficient to cope with mesh distortion, the ALE domain can be subdivided into subdomains with different constitutive behavior. In this thesis, the same constitutive model is used for every subdomain, but different material parameters are assigned to them. Usually, stiffer parameters are chosen close to the interface or moving boundary and weaker ones in the bulk field. This allows for conservation of element shapes close to the interface and makes it attractive to push large amounts of the mesh deformation into the bulk field. This strategy has been pursued in the buckling of a cylindrical shell under fluid loading in Section 4.3.1.

The nonlinear character of (A.2) requires an evaluation of the residual vector and its linearization in every nonlinear iteration. In scenarios with only moderate mesh motion, it is often sufficient to update the stiffness matrix only once per time step or even only once in the beginning of the simulation. Since an exact satisfaction of (A.2) is not necessary and since the ALE field does not alter the physical model, violations due to a modified NEWTON-RAPHSON scheme with an inexact Jacobian matrix and insufficient number of iterations do not harm the overall solution procedure, if sufficient mesh quality can be reached. Thus, even in the nonlinear case, ALE mesh motion is not included into the nonlinear convergence check, cf. Section 5.3.1.

A.3 A Comparison of ALE Mesh Motion Algorithms

To compare robustness of the presented methods, a test case is studied that is driven by a prescribed rotation. It has been specifically designed to trigger difficult mesh motion scenarios including large deformation and strong shear components close to the moving boundary. Robustness is assessed in terms of achievable rotation angle before inadmissible element shapes

occur. Mesh distortion is considered inadmissible as soon as a single element shows $\det \mathbf{J} \leq 0$ with \mathbf{J} being the Jacobian matrix for the isoparametric mapping evaluated at each node of the element. For quadrilaterals or hexahedra, mesh quality is judged by the quality measure $D^{\mathcal{G}}$ of ODDY *et al.* [178], reading

$$D^{\mathcal{G}} = \sum_{i=1}^{n^{\dim}} \sum_{j=1}^{n^{\dim}} c_{ij}^2 - \frac{1}{n^{\dim}} \left(\sum_{k=1}^{n^{\dim}} c_{kk} \right) \quad (\text{A.3})$$

with

$$c_{ij} = \frac{1}{\det \mathbf{J}} \sum_{k=1}^{n^{\dim}} J_{ki} J_{kj}$$

and $n^{\dim} \in \{2, 3\}$ being the number of spatial dimensions. This measure is evaluated at every node and the largest nodal value in the element is used as the element's quality measure [127, 178]. It accounts for both shearing and stretching, while being invariant to rigid body motions as well as the element size [213]. It results in $D^{\mathcal{G}} = 0$ for squared elements and increases with increasing distortion. Unfortunately, an actual geometric interpretation is not straightforward. For some numerical tests, see the work by HYUN and LINDGREN [127] and SARRATE and HUERTA [213], who use measure (A.3) to develop mesh smoothing techniques.

The test case is designed as follows: A squared domain with an inner rectangular whole is considered, cf. Figure A.3. The inner rectangle is subject to a prescribed rotation

$$\bar{\mathbf{d}}^{\mathcal{G}}(t) = \begin{bmatrix} \cos \omega t & -\sin \omega t \\ \sin \omega t & \cos \omega t \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x(\cos \omega t - 1) - y \sin \omega t \\ x \sin \omega t + y(\cos \omega t - 1) \end{bmatrix}_{xy}$$

with the angular velocity $\omega = \pi/360^\circ$, while all nodes on the outer boundary are fixed. Using a pseudo-time t with an artificial time step size $\Delta t = 0.5$, the rotation angle is increased in increments of 0.5° . A similar test has previously been performed by BRUDER [29] who compared the presented mesh motion schemes as well as their implementations in the in-house research code `Baci` [243].

The springs model as well as the elasticity approach with different material laws are compared to each other. LAPLACEian smoothing isn't included in the comparison, since it is known to not be qualified for non domains [62]. Table A.1 provides an overview of the compared methods. Figure A.4 shows the maximally deformed state, that can be achieved with each model. The springs model allows only for small deformation. In addition, element distortion concentrates at the corners of the moving rectangle. Using an elasticity model enables larger rotation angles. The more the constitutive law resists against shearing deformation, the larger deformation is acceptable. The SVK model, which is said to be unsuitable for large distortions, shows similar behavior as the springs approach. Distinctively larger deformation is possible with NH, logNH and MR constitutive laws. As can be seen from the most right column in Table A.1, an increasing value of POISSON's ratio $\nu^{\mathcal{G}}$ enables larger rotation angles, since it immediately increases shear resistance. The parameter $c_3^{\mathcal{G}}$ of the MR material shows a similar effect. Additionally, element distortion does not concentrate at the vertices of the moving boundary, but rather happens more inside the bulk field.

Table A.1: Methods involved in the comparison of ALE mesh motion algorithms — The springs approach is compared to the elasticity model with various constitutive laws. The rightmost column indicates the rotation angle, at which the first element in the mesh exhibited inadmissible distortion. Clearly, elasticity models with high resistance against shear are most robust in this test case.

ID	Model	Parameters	Max. rotation angle
<i>ale01</i>	Springs	—	27.0°
<i>ale02a</i>	Solid with SVK	$E^S = 1.0, \nu^S = 0.0$	18.0°
<i>ale02b</i>	Solid with SVK	$E^S = 1.0, \nu^S = 0.3$	21.0°
<i>ale02c</i>	Solid with SVK	$E^S = 1.0, \nu^S = 0.45$	27.5°
<i>ale02d</i>	Solid with SVK	$E^S = 1.0, \nu^S = 0.49$	35.0°
<i>ale03a</i>	Solid with NH	$E^S = 1.0, \nu^S = 0.0$	35.0°
<i>ale03b</i>	Solid with NH	$E^S = 1.0, \nu^S = 0.3$	49.0°
<i>ale03c</i>	Solid with NH	$E^S = 1.0, \nu^S = 0.45$	78.5°
<i>ale04a</i>	Solid with logNH	$E^S = 1.0, \nu^S = 0.0$	35.0°
<i>ale04b</i>	Solid with logNH	$E^S = 1.0, \nu^S = 0.3$	43.0°
<i>ale04c</i>	Solid with logNH	$E^S = 1.0, \nu^S = 0.45$	57.5°
<i>ale04d</i>	Solid with logNH	$E^S = 1.0, \nu^S = 0.49$	128.0°
<i>ale05a</i>	Solid with MR	$c_1^S = 1.0, c_2^S = 1.0, c_3^S = 1.0$	37.5°
<i>ale05b</i>	Solid with MR	$c_1^S = 1.0, c_2^S = 1.0, c_3^S = 10.0$	40.0°
<i>ale05c</i>	Solid with MR	$c_1^S = 1.0, c_2^S = 1.0, c_3^S = 10.0$	60.5°
<i>ale05d</i>	Solid with MR	$c_1^S = 1.0, c_2^S = 1.0, c_3^S = 500.0$	96.0°
<i>ale05e</i>	Solid with MR	$c_1^S = 1.0, c_2^S = 1.0, c_3^S = 1000.0$	100.5°

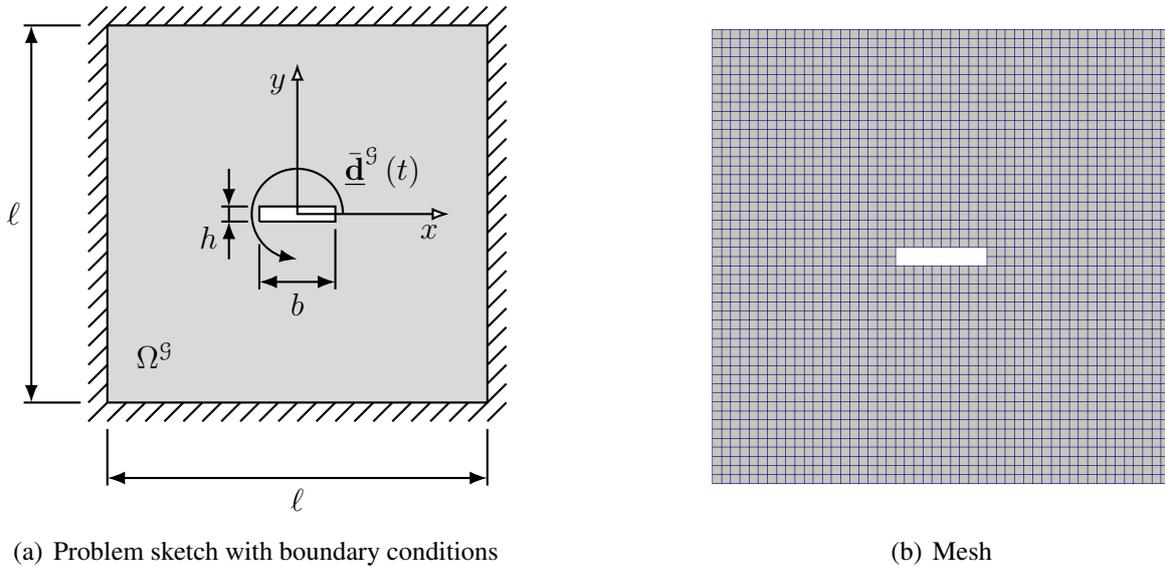


Figure A.3: Square domain with rotating rectangular slot — *Left*: A two-dimensional squared domain ($\ell = 5.0$) is fixed at its outer boundaries, while its inner rectangular slot ($h = 0.2$, $b = 1.0$) undergoes a prescribed rotation $\bar{\mathbf{d}}^S(t)$. *Right*: Grid for comparison of mesh motion techniques.

Although increasing values of ν^S or c_3^S seem to be beneficial for the mesh motion, a trade-off needs to be done since they might cause numerical trouble. The more the material approaches the incompressible limit $\nu^S \rightarrow 0.5$, the harder the linear system is to solve. In the present example, results for values $\nu^S > 0.49$ could not be computed since they either led to floating point exceptions during the computation or to convergence issues.

Finally, some recommendations can be given:

- **LAPLACEian smoothing**: Although LAPLACEian smoothing can only be applied to domains, it is still an attractive choice due to its simplicity and low numerical effort. In cases of mild and uniform deformations like in the pressure wave example or in the computational analysis of AAA, LAPLACEian smoothing is a good and very reasonable choice. Basically, it can be used as a parameter-free scheme. However, if necessary more sophisticated variants are available, cf. for example [158].
- **Springs**: Due to its purely geometric character, the springs model is easy to implement. It can be used very well in the case of mild and uniform deformations. Especially scenarios with low or even without shear, but large elongation or compression can be handled without any problems like in the one-dimensional example presented in Section 3.4.
- **Elasticity model**: The most robust approach in presence of large deformation is definitely the elasticity model, especially with logNH or MR constitutive laws. It shows good preservation of mesh regularity in the vicinity of the interface or moving boundary. However, the nonlinearity of the approach results in increased numerical effort since the stiffness

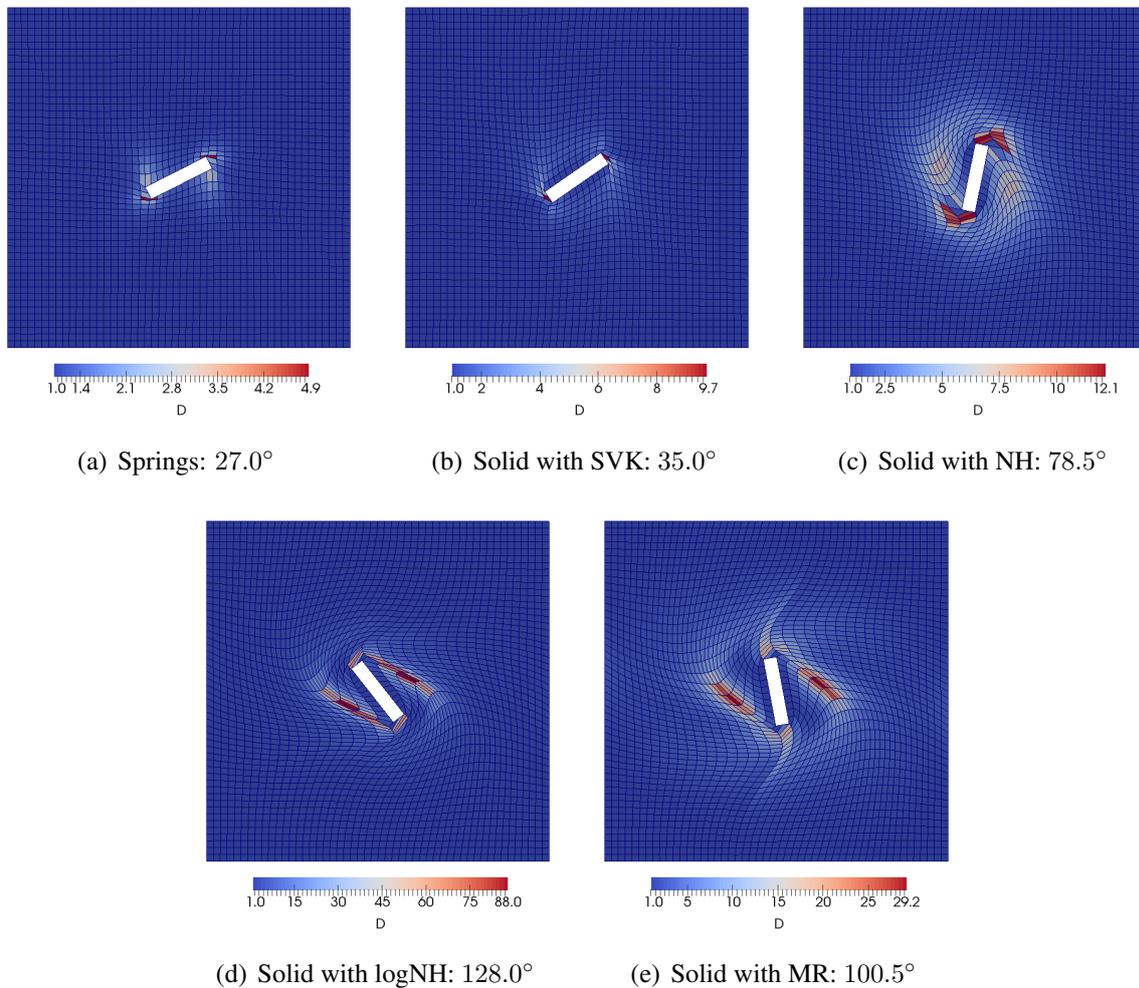


Figure A.4: Final deformations of different ALE mesh motion schemes in a rotation test — The state of maximum rotation in counterclockwise direction is shown, i.e. the state just before at least one element in the mesh undergoes an inadmissible distortion. Elements are color-coded w.r.t. the quality measure D^S , cf. (A.3), where large values (red) stand for highly distorted elements. The solid models with high shear resistance, namely logarithmic NEO-HOOKE and MOONEY-RIVLIN materials, allow for the largest deformation while still maintaining acceptable mesh quality, especially in the vicinity of the moving boundary.

matrix needs to be evaluated in every nonlinear iteration. This cost is not negligible at all, but might be the only choice if all other approaches fail.

B Hardware and Computational Resources

In-house Cluster

The in-house linux-based cluster is jointly operated by the Institute for Computational Mechanics¹ and the Mechanics & High Performance Computing Group², both at Technical University of Munich. It consists of two partitions, denoted by *Opteron* and *Xeon*.

- The *Opteron* partition consists of 1408 *AMD Opteron 6128 Magny Cours* cores organized in 88 nodes with 2 octocore CPUs each. Nominal frequency is 2.0 GHz. Each node has a memory of 32 GB.
- The *Xeon* partition consists of 128 *Intel Xeon E5-2670 SandyBridge* cores organized in 8 nodes with 2 octocore CPUs each. Nominal frequency is 2.6 GHz. Each node has a memory of 32 GB.

Communication between nodes is handled via an Infiniband network (Mellanox ConnexX, 40 GBit/s).

SuperMUC Phase 2 at Leibniz Rechenzentrum

The *SuperMUC Petascale System*³ at Leibniz Supercomputing Centre in Garching, Germany, consists of two installations, named *Phase 1* and *Phase 2*. In this thesis, computational nodes of *Phase 2* have been used. It consists of 86016 *Haswell Xeon Processor E5-2697 v3* cores, grouped into 3072 nodes with 2 processors with 14 cores each. Nominal frequency is 2.6 GHz. Per core, 2.3 GB of memory are available. Nodes are connected via a *Infiniband FDR14* network.

¹www.lnm.mw.tum.de

²www.mhpc.mw.tum.de

³<https://www.lrz.de/services/compute/supermuc/systemdescription/>, visited on June 1st 2016

Bibliography

- [1] D. G. Anderson. Iterative Procedures for Nonlinear Integral Equations. *Journal of the Association of Computing Machinery*, 12(4):547–560, 1965.
- [2] F. Armero and E. Love. An arbitrary Lagrangian–Eulerian finite element method for finite strain plasticity. *International Journal for Numerical Methods in Engineering*, 57(4):471–508, 2003.
- [3] U. M. Ascher and L. R. Petzold. *Computer methods for ordinary differential equations and differential algebraic equations*. SIAM, Philadelphia, PA, USA, 1998.
- [4] H. Askes, A. Rodríguez-Ferran, and A. Huerta. Adaptive analysis of yield line patterns in plates with the arbitrary Lagrangian–Eulerian method. *Computers & Structures*, 70(3):257–271, 1999.
- [5] H. Askes and L. J. Sluys. Remeshing strategies for adaptive ALE analysis of strain localisation. *European Journal of Mechanics - A/Solids*, 19(3):447–467, 2000.
- [6] J. L. F. Aymone, E. Bittencourt, and G. J. Creus. Simulation of 3D metal-forming using an arbitrary Lagrangian–Eulerian finite element method. *Journal of Materials Processing Technology*, 110(2):218–232, 2001.
- [7] S. Badia, F. Nobile, and C. Vergara. Fluid–structure partitioned procedures based on Robin transmission conditions. *Journal of Computational Physics*, 227(14):7027–7051, 2008.
- [8] S. Badia, F. Nobile, and C. Vergara. Robin–Robin preconditioned Krylov methods for fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 198(33–36):2768–2784, 2009.
- [9] S. Badia, A. Quaini, and A. Quarteroni. Modular vs. non-modular preconditioners for fluid–structure systems with large added-mass effect. *Computer Methods in Applied Mechanics and Engineering*, 197(49–50):4216–4232, 2008.
- [10] A. T. Barker and X.-C. Cai. Scalable parallel methods for monolithic coupling in fluid–structure interaction with application to blood flow modeling. *Journal of Computational Physics*, 229(3):642–659, 2010.
- [11] J. T. Batina. Unsteady Euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis. *AIAA Journal*, 29(3):327–333, 1991.
- [12] Y. Bazilevs, V. Calo, Y. Zhang, and T. Hughes. Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38(4–5):310–322, 2006.
- [13] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T. E. Tezduyar. 3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids*, 65(1–3):207–235, 2011.
- [14] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65(1-3):236–253, 2011.
- [15] F. B. Belgacem. The Mortar finite element method with Lagrange multipliers. *Numerische Mathematik*, 84(2):173–197, 1999.
- [16] T. Belytschko, W. K. Liu, B. Moran, and K. I. Elkhodary. *Nonlinear Finite Elements for Continua and Structures*. Wiley, Chichester, West Sussex, 2nd edition, 2014.
- [17] T. Belytschko, H.-J. Yen, and R. Mullen. Mixed methods for time integration. *Computer Methods in Applied Mechanics and Engineering*, 17–18:259–275, 1979.

- [18] T. B. Belytschko and J. M. Kennedy. Computer models for subassembly simulation. *Nuclear Engineering and Design*, 49(1):17–38, 1978.
- [19] T. B. Belytschko, J. M. Kennedy, and D. F. Schoeberle. Quasi-Eulerian Finite Element Formulation for Fluid-Structure Interaction. *Journal of Pressure Vessel Technology*, 102(1):62–69, 1980.
- [20] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.
- [21] P. B. Bochev, M. D. Gunzburger, and R. B. Lehoucq. On stabilized finite element methods for the Stokes problem in the small time step limit. *International Journal for Numerical Methods in Fluids*, 53(4):573–597, 2007.
- [22] P. B. Bochev, M. D. Gunzburger, and J. N. Shadid. On inf-sup stabilized finite element methods for transient problems. *Computer Methods in Applied Mechanics and Engineering*, 193(15–16):1471–1489, 2004.
- [23] E. G. Boman, U. V. Çatalyürek, C. Chevalier, and K. D. Devine. The Zoltan and Isorropia Parallel Toolkits for Combinatorial Scientific Computing: Partitioning, Ordering and Coloring. *Scientific Programming*, 20(2):29–150, 2012.
- [24] J. Bonet and R. D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 2nd edition, 2008.
- [25] B. Bornemann. *Time Integration Algorithms for the Steady States of Dissipative Non-Linear Dynamic Systems*. PhD thesis, Imperial College London, 2003.
- [26] M. Brezina. *Robust Iterative Methods on Unstructured Meshes*. PhD thesis, University of Colorado at Denver, 1997.
- [27] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, 2nd edition, 2000.
- [28] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1):199–259, 1982.
- [29] L. Bruder. A non-linear ALE mesh moving strategy for large-displacement fluid-structure interaction problems. Bachelor’s Thesis, Technische Universität München, 2014.
- [30] P. Brune, M. Knepley, B. Smith, and X. Tu. Composing Scalable Nonlinear Algebraic Solvers. *SIAM Review*, 57(4):535–565, 2015.
- [31] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Ltd, 2nd edition, 2008.
- [32] V. Carey, D. Estep, A. Johansson, M. Larson, and S. Tavener. Blockwise Adaptivity for Time Dependent Problems Based on Coarse Scale Adjoint Solutions. *SIAM Journal on Scientific Computing*, 32(4):2121–2145, 2010.
- [33] P. Causin, J. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194(42–44):4506–4527, 2005.
- [34] F. Ceschino. Evaluation de l’erreur par pas dans les problèmes différentiels. *Chiffres*, 5:223–229, 1962.
- [35] F. Ceschino and J. Kuntzmann. *Numerical solution of initial value problems*. Prentice-Hall, Englewood Cliffs, 1966.
- [36] C.-K. Choi and H.-J. Chung. Error estimates and adaptive time stepping for various direct time integration methods. *Computers & Structures*, 60(6):923–944, 1996.
- [37] E. Chow and A. Patel. Fine-Grained Parallel Incomplete LU Factorization. *SIAM Journal on Scientific Computing*, 37(2):C169–C193, 2015.
- [38] J. Chung, E.-H. Cho, and K. Choi. A priori error estimator of the generalized- α method for structural dynamics. *International Journal for Numerical Methods in Engineering*, 57(4):537–554, 2003.

- [39] J. Chung and G. Hulbert. A Time Integration Algorithm for Structural Dynamics With Improved Numerical Dissipation: The Generalized- α Method. *Journal of Applied Mechanics*, 60(2):371–375, 1993.
- [40] P. Crosetto, S. Deparis, G. Fourestey, and A. Quarteroni. Parallel Algorithms for Fluid-Structure Interaction Problems in Haemodynamics. *SIAM Journal on Scientific Computing*, 33(4):1598–1622, 2011.
- [41] P. Crosetto, P. Reymond, S. Deparis, D. Kontaxakis, N. Stergiopoulos, and A. Quarteroni. Fluid–structure interaction simulation of aortic blood flow. *Computers & Fluids*, 43(1):46–57, 2011. Symposium on High Accuracy Flow Simulations. Special Issue Dedicated to Prof. Michel Deville Symposium on High Accuracy Flow Simulations.
- [42] E. Cyr, J. Shadid, and T. Wildey. Towards efficient backward-in-time adjoint computations using data compression techniques. *Computer Methods in Applied Mechanics and Engineering*, 288:24–44, 2015.
- [43] T. A. Davis. Algorithm 832: UMFPACK V4.3 — an Unsymmetric-pattern Multifrontal Method. *ACM Transactions on Mathematical Software*, 30(2):196–199, 2004.
- [44] T. A. Davis. *Direct Methods for Sparse Linear Systems*. Fundamentals of Algorithms. SIAM, 2006.
- [45] R. de Borst, M. A. Crisfield, J. J. C. Remmers, and C. V. Verhoosel. *Non-Linear Finite Element Analysis of Solids and Structures*. John Wiley & Sons, Ltd, Chichester, West Sussex, 2nd edition, 2012.
- [46] T. de Mulder. The role of bulk viscosity in stabilized finite element formulations for incompressible flow: A review. *Computer Methods in Applied Mechanics and Engineering*, 163(1–4):1–10, 1998.
- [47] E. A. de Souza Neto, D. Perić, M. Dutko, and D. R. J. Owen. Design of simple low order finite elements for large strain analysis of nearly incompressible solids. *International Journal of Solids and Structures*, 33(20–22):3277–3296, 1996.
- [48] H. de Sterck. A Nonlinear GMRES Optimization Algorithm for Canonical Tensor Decomposition. *SIAM Journal on Scientific Computing*, 34(3):A1351–A1379, 2012.
- [49] C. Degand and C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers & Structures*, 80(3–4):305–316, 2002.
- [50] S. Deparis, M. Discacciati, G. Fourestey, and A. Quarteroni. Fluid–structure algorithms based on Steklov–Poincaré operators. *Computer Methods in Applied Mechanics and Engineering*, 195(41–43):5797–5812, 2006. John H. Argyris Memorial Issue. Part {II}.
- [51] S. Deparis, M. A. Fernández, and L. Formaggia. Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37:601–616, 2003.
- [52] S. Deparis, D. Forti, P. Gervasio, and A. Quarteroni. INTERNODES: an accurate interpolation-based method for coupling the Galerkin solutions of PDEs on subdomains featuring non-conforming interfaces. *Computers & Fluids*, 141:22–41, 2016.
- [53] S. Deparis, D. Forti, G. Grandperrin, and A. Quarteroni. FaCSI: A block parallel preconditioner for fluid–structure interaction in hemodynamics. *Journal of Computational Physics*, 327:700–718, 2016.
- [54] S. Deparis, D. Forti, and A. Quarteroni. A Rescaled Localized Radial Basis Function Interpolation on Non-Cartesian and Nonconforming Grids. *SIAM Journal on Scientific Computing*, 36(6):A2745–A2762, 2014.
- [55] S. Deparis, D. Forti, and A. Quarteroni. A Fluid–Structure Interaction Algorithm Using Radial Basis Function Interpolation Between Non-Conforming Interfaces. In Y. Bazilevs and K. Takizawa, editors, *Advances in Computational Fluid-Structure Interaction and Flow Simulation*, Modeling and Simulation in Science, Engineering and Technology, pages 439–450. Springer, 2016.
- [56] W. Dettmer and D. Perić. A computational framework for fluid–structure interaction: Finite element formulation and applications. *Computer Methods in Applied Mechanics and Engineering*, 195(41–43):5754–5779, 2006.

- [57] W. G. Dettmer and D. Perić. A new staggered scheme for fluid–structure interaction. *International Journal for Numerical Methods in Engineering*, 93(1):1–22, 2013.
- [58] P. Deuffhard and F. Bornemann. *Numerische Mathematik 2: Gewöhnliche Differentialgleichungen*, volume 2. De Gruyter, Boston, 3rd edition, 2008.
- [59] P. Deuffhard and A. Hohmann. *Numerische Mathematik 1: Eine algorithmische Einführung*, volume 1. De Gruyter, Boston, 4th edition, 2008.
- [60] J. Donea, S. Giuliani, and J. Halleaux. An Arbitrary Lagrangian-Eulerian Finite Element Method for Transient Dynamic Fluid-Structure Interaction. *Computer Methods in Applied Mechanics and Engineering*, 33:689–723, 1982.
- [61] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. Wiley, 2003.
- [62] J. Donea, A. Huerta, J. P. Ponthot, and A. Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian Methods. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 1: Fundamentals, chapter 14, pages 1–25. John Wiley & Sons, Ltd, New York, 2004.
- [63] J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980.
- [64] A. Ehrl, A. Popp, V. Gravemeier, and W. A. Wall. A dual mortar approach for mesh tying within a variational multiscale method for incompressible flow. *International Journal for Numerical Methods in Fluids*, 76(1):1–27, 2014.
- [65] A. Eken and M. Sahin. A parallel monolithic algorithm for the numerical simulation of large-scale fluid structure interaction problems. *International Journal for Numerical Methods in Fluids*, 80(12):687–714, 2015.
- [66] R. England. Error estimates for Runge-Kutta type solutions to systems of ordinary differential equations. *The Computer Journal*, 12(2):166–170, 1969.
- [67] J. D. Faires and R. L. Burden. *Numerical Methods*. Brooks/Cole, Pacific Grove, CA, USA, 4th edition, 2012.
- [68] P. Farah, A. Popp, and W. A. Wall. Segment-based vs. element-based integration for mortar methods in computational contact mechanics. *Computational Mechanics*, 55(1):209–228, 2015.
- [69] P. Farah, A.-T. Vuong, W. A. Wall, and A. Popp. Volumetric coupling approaches for multiphysics simulations on non-matching meshes. *International Journal for Numerical Methods in Engineering*, accepted, 2016.
- [70] C. Farhat. CFD-Based Nonlinear Computational Aeroelasticity. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 3: Fluids, chapter 13, pages 459–480. John Wiley & Sons, Ltd, New York, 2004.
- [71] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163(1–4):231–245, 1998.
- [72] C. Farhat, M. Lesoinne, and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157(1–2):95–114, 1998.
- [73] E. Fehlberg. New high-order Runge-Kutta formulas with step size control for systems of first and second order differential equations. *Zeitschrift für Angewandte Mathematik und Mechanik*, 44 (Sonderheft):T17–T19, 1964.
- [74] E. Fehlberg. Klassische Runge-Kutta-Formeln fünfter und siebenter Ordnung mit Schrittweiten-Kontrolle. *Computing*, 4(2):93–106, 1969.

-
- [75] E. Fehlberg. Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems. Technical Report 315, NASA, 1969.
- [76] E. Fehlberg. Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme. *Computing*, 6(1–2):61–71, 1970.
- [77] M. A. Fernández, J.-F. Gerbeau, and C. Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007.
- [78] M. A. Fernández and M. Moubachir. A Newton method using exact jacobians for solving fluid–structure coupling. *Computers & Structures*, 83(2–3):127–142, 2005.
- [79] M. A. Fernández, J. Mullaert, and M. Vidrascu. Explicit Robin-Neumann schemes for the coupling of incompressible fluids with thin-walled structures. *Computer Methods in Applied Mechanics and Engineering*, 267:566–593, 2013.
- [80] 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.
- [81] C. Förster. *Robust methods for fluid-structure interaction with stabilised finite elements*. PhD thesis, Universität Stuttgart, Stuttgart, 2007.
- [82] 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.
- [83] D. Forti. *Parallel Algorithms for the Solution of Large-Scale Fluid-Structure Interaction Problems in Hemodynamics*. PhD thesis, Ecole Polytechnique Federale de Lausanne, Lausanne, CH, 2016.
- [84] M. S. Gadala. Recent trends in ALE formulation and its applications in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 193(39–41):4247–4275, 2004.
- [85] M. S. Gadala, M. R. Movahhedy, and J. Wang. On the mesh motion for ALE modeling of metal forming processes. *Finite Elements in Analysis and Design*, 38(5):435–459, 2002.
- [86] M. S. Gadala and J. Wang. ALE formulation and its application in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 167(1–2):33–55, 1998.
- [87] M. Gee. *Effiziente Lösungsstrategien in der nichtlinearen Schalenmechanik*. PhD thesis, Universität Stuttgart, Institut für Baustatik, Stuttgart, 2005.
- [88] M. W. Gee, C. Förster, and W. A. Wall. A computational strategy for prestressing patient-specific biomechanical problems under finite deformation. *International Journal for Numerical Methods in Biomedical Engineering*, 26(1):52–72, 2010.
- [89] M. W. Gee, U. Küttler, and W. A. Wall. Truly monolithic algebraic multigrid for fluid–structure interaction. *International Journal for Numerical Methods in Engineering*, 85(8):987–1016, 2011.
- [90] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala. ML 5.0 Smoothed Aggregation User’s Guide. Technical Report SAND2006-2649, Sandia National Laboratories, Albuquerque, NM (USA) 87185, 2006.
- [91] J.-F. Gerbeau and M. Vidrascu. A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 37(4):631–647, 2003.
- [92] 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, 2008. Computational Methods in Fluid–Structure Interaction.
- [93] S. Ghosh and N. Kikuchi. An arbitrary Lagrangian-Eulerian finite element method for large deformation analysis of elastic-viscoplastic solids. *Computer Methods in Applied Mechanics and Engineering*, 86(2):127–188, 1991.

- [94] S. Ghosh and S. Raju. R–S adapted arbitrary Lagrangian–Eulerian finite element method for metal-forming problems with strain localization. *International Journal for Numerical Methods in Engineering*, 39(19):3247–3272, 1996.
- [95] M. Gitterle. *A dual mortar formulation for finite deformation frictional contact problems including wear and thermal coupling*. Dissertation, Technische Universität München, München, 2012.
- [96] M. Gitterle, A. Popp, M. W. Gee, and W. A. Wall. Finite deformation frictional mortar contact using a semi-smooth Newton method with consistent linearization. *International Journal for Numerical Methods in Engineering*, 84(5):543–571, 2010.
- [97] S. Giuliani. An algorithm for continuous rezoning of the hydrodynamic grid in Arbitrary Lagrangian–Eulerian computer codes. *Nuclear Engineering and Design*, 72(2):205–212, 1982.
- [98] V. Gravemeier. *The variational multiscale method for laminar and turbulent incompressible flow*. PhD thesis, Universität Stuttgart, Holzgartenstr. 16, 70174 Stuttgart, 2003.
- [99] V. Gravemeier, A. Comerford, L. Yoshihara, M. Ismail, and W. A. Wall. A novel formulation for Neumann inflow boundary conditions in biomechanics. *International Journal for Numerical Methods in Biomedical Engineering*, 28(5):560–573, 2012.
- [100] V. Gravemeier, M. W. Gee, M. Kronbichler, and W. A. Wall. An algebraic variational multiscale–multigrid method for large eddy simulation of turbulent flow. *Computer Methods in Applied Mechanics and Engineering*, 199(13–16):853–864, 2010.
- [101] P. M. Gresho, D. F. Griffiths, and D. J. Silvester. Adaptive time-stepping for incompressible flow. Part I: Scalar Advection-Diffusion. *SIAM Journal on Scientific Computing*, 30(4):2018–2054, 2008.
- [102] P. M. Gresho and R. L. Sani. *Incompressible Flow and the Finite Element Method*. Wiley, 2000.
- [103] M. E. Gurtin. *An Introduction to Continuum Mechanics*. Academic Press, New York, 1981.
- [104] K. Gustafsson. Control Theoretic Techniques for Step Size Selection in Explicit Runge-Kutta Methods. *ACM Transactions on Mathematical Software*, 17(4):533–554, 1991.
- [105] K. Gustafsson. *Control of Error and Convergence in ODE Solvers*. PhD thesis, Lund Institute of Technology, 1992.
- [106] K. Gustafsson. Control-theoretic Techniques for Step Size Selection in Implicit Runge-Kutta Methods. *ACM Transactions on Mathematical Software*, 20(4):496–517, 1994.
- [107] K. Gustafsson, M. Lundh, and G. Söderlind. A PI Step Size Control for the Numerical Solution of Ordinary Differential Equations. *BIT*, 28(2):270–287, 1988.
- [108] W. Hackbusch. *Multi-Grid Methods and Applications*, volume 4 of *Springer Series in Computational Mathematics*. Springer Berlin / Heidelberg, 1985.
- [109] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*, volume 95 of *Applied Mathematical Sciences*. Springer, 1994.
- [110] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2nd edition, 1993.
- [111] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2nd edition, 1996.
- [112] M. Heil. An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(1–2):1–23, 2004.
- [113] M. Heil, A. Hazel, and J. Boyle. Solvers for large-displacement fluid–structure interaction problems: segregated versus monolithic approaches. *Computational Mechanics*, 43(1):91–101, 2008.
- [114] J. Hermansson and P. Hansbo. A variable diffusion method for mesh smoothing. *Communications in Numerical Methods in Engineering*, 19(11):897–908, 2003.

-
- [115] M. A. Heroux. AztecOO User Guide. Technical Report SAND2004-3796, Sandia National Laboratories, Albuquerque, NM (USA) 87185, 2007.
- [116] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, 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 Transactions on Mathematical Software*, 31(3):397–423, 2005.
- [117] C. W. Hirt, A. A. Amsden, and J. L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 1974.
- [118] G. A. Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. Wiley, 2000.
- [119] S. Hübner and B. I. Wohlmuth. A primal–dual active set strategy for non-linear multibody contact problems. *Computer Methods in Applied Mechanics and Engineering*, 194(27–29):3147–3166, 2005.
- [120] A. Huerta and F. Casadei. New ALE applications in non-linear fast-transient solid dynamics. *Engineering Computations*, 11(4):317–345, 1994.
- [121] A. Huerta, A. Rodríguez-Ferran, P. Díez, and J. Sarrate. Adaptive finite element strategies based on error assessment. *International Journal for Numerical Methods in Engineering*, 46(10):1803–1818, 1999.
- [122] J. Huétink, P. T. Vreede, and J. van der Lugt. Progress in mixed Eulerian-Lagrangian finite element simulation of forming processes. *International Journal for Numerical Methods in Engineering*, 30(8):1441–1457, 1990.
- [123] T. J. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: a stable Petrov-Galerkin formulation of the stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59(1):85–99, 1986.
- [124] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover, 2000.
- [125] T. J. R. Hughes, W. K. Liu, and T. K. Zimmermann. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29(3):329–349, 1981.
- [126] G. M. Hulbert and I. Jang. Automatic time step control algorithms for structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 126(1–2):155–178, 1995.
- [127] S. Hyun and L.-E. Lindgren. Smoothing and adaptive remeshing schemes for graded element. *Communications in Numerical Methods in Engineering*, 17(1):1–17, 2001.
- [128] M. Ismail, W. A. Wall, and M. W. Gee. Adjoint-based inverse analysis of windkessel parameters for patient-specific vascular models. *Journal of Computational Physics*, 244:113–130, 2013.
- [129] K. E. Jansen, C. H. Whiting, and G. M. Hulbert. A generalized- α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190(3–4):305–319, 2000.
- [130] C. Johnson. Error Estimates and Adaptive Time-Step Control for a Class of One-Step Methods for Stiff Ordinary Differential Equations. *SIAM Journal on Numerical Analysis*, 25(4):908–926, 1988.
- [131] C. Johnson, R. Rannacher, and M. Boman. Numerics and Hydrodynamic Stability: Toward Error Control in Computational Fluid Dynamics. *SIAM Journal on Numerical Analysis*, 32(4):1058–1079, 1995.
- [132] M. M. Joosten, W. G. Dettmer, and D. Perić. On the temporal stability and accuracy of coupled problems with reference to fluid–structure interaction. *International Journal for Numerical Methods in Fluids*, 64(10–12):1363–1378, 2010.
- [133] D. Kay, P. Gresho, D. F. Griffiths, and D. Silvester. Adaptive time-stepping for incompressible flow. Part II: Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 32(1):111–128, 2010. © 2008 Society for Industrial and Applied Mathematics.

- [134] C. T. Kelley. *Solving Nonlinear Equations with Newton's method*. Fundamentals of Algorithms. SIAM, Philadelphia, PA, USA, 2003.
- [135] H.-G. Kim. A new coupling strategy for fluid–solid interaction problems by using the interface element method. *International Journal for Numerical Methods in Engineering*, 81(4):403–428, 2010.
- [136] P. Kjellgren and J. Hyvärinen. An Arbitrary Lagrangian-Eulerian finite element method. *Computational Mechanics*, 21(1):81–90, 1998.
- [137] T. Klöppel. *A Finite Element Model for the Human Red Blood Cell*. PhD thesis, Technische Universität München, München, 2012.
- [138] T. Klöppel, A. Popp, U. Küttler, and W. A. Wall. Fluid–structure interaction for non-conforming interfaces based on a dual mortar formulation. *Computer Methods in Applied Mechanics and Engineering*, 200(45–46):3111–3126, 2011.
- [139] T. Klöppel and W. A. Wall. A novel two-layer, coupled finite element approach for modeling the nonlinear elastic and viscoelastic behavior of human erythrocytes. *Biomechanics and Modeling in Mechanobiology*, 10(4):445–459, 2011.
- [140] N. Koshiba, J. Ando, X. Chen, and T. Hisada. Multiphysics Simulation of Blood Flow and LDL Transport in a Porohyperelastic Arterial Wall Model. *Journal of Biomechanical Engineering*, 129(3):374–385, 2006.
- [141] R. H. Krause and B. I. Wohlmuth. Nonconforming Domain Decomposition Methods Techniques for Linear Elasticity. *East–West Journal of Numerical Mathematics*, 8(3):177–206, 2000.
- [142] U. Küttler. *Effiziente Lösungsverfahren für Fluid-Struktur-Interaktions-Probleme*. Dissertation, Technische Universität München, München, 2009.
- [143] U. Küttler, M. W. Gee, C. Förster, A. Comerford, and W. A. Wall. Coupling strategies for biomedical fluid–structure interaction problems. *International Journal for Numerical Methods in Biomedical Engineering*, 26(3–4):305–321, 2010.
- [144] U. Küttler and W. A. Wall. Fixed-point fluid–structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43(1):61–72, 2008.
- [145] U. Küttler and W. A. Wall. Vector Extrapolation for Strong Coupling Fluid-Structure Interaction Solvers. *Journal of Applied Mechanics*, 76(2):021205–1–021205–7, 2009.
- [146] B. P. Lamichhane, R. P. Stevenson, and B. I. Wohlmuth. Higher Order Mortar Finite Element Methods in 3D with Dual Lagrange Multiplier Bases. *Numerische Mathematik*, 102(1):93–121, 2005.
- [147] B. P. Lamichhane and B. I. Wohlmuth. Biorthogonal bases with local support and approximation properties. *Mathematics of Computation*, 76(257):233–249, 2007.
- [148] U. Langer and H. Yang. Numerical Simulation of Fluid–Structure Interaction Problems with Hyperelastic Models: A Monolithic Approach. *Mathematics and Computers in Simulation*, 2016.
- [149] U. Langer and H. Yang. Robust and efficient monolithic fluid-structure-interaction solvers. *International Journal for Numerical Methods in Engineering*, 108(4):303–325, 2016.
- [150] M. G. Larson and F. Bengzon. *The Finite Element Method: Theory, Implementation and Applications*, volume 10 of *Texts in Computational Science and Engineering*. Springer, Berlin / Heidelberg, 2013.
- [151] F. Larsson and K. Runesson. A sequential-adaptive strategy in space–time with application to consolidation of porous media. *Computer Methods in Applied Mechanics and Engineering*, 288:146–171, 2015.
- [152] T. A. Laursen. *Computational Contact and Impact Mechanics: Fundamentals of Modeling Interfacial Phenomena in Nonlinear Finite Element Analysis*. Springer Berlin / Heidelberg, 2003.
- [153] P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 190(24–25):3039–3067, 2001.
- [154] X. D. Li and N.-E. Wiberg. Implementation and adaptivity of a space-time finite element method for structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 156(1–4):211–229, 1998.

-
- [155] X. D. Li, L. F. Zeng, and N.-E. Wiberg. A simple local error estimator and an adaptive time-stepping procedure for direct integration method in dynamic analysis. *Communications in Numerical Methods in Engineering*, 9(4):273–292, 1993.
- [156] W. K. Liu, T. Belytschko, and H. Chang. An arbitrary lagrangian-eulerian finite element method for path-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 58(2):227–245, 1986.
- [157] W. K. Liu, H. Chang, J.-S. Chen, and T. Belytschko. Arbitrary lagrangian-eulerian petrov-galerkin finite elements for nonlinear continua. *Computer Methods in Applied Mechanics and Engineering*, 68(3):259–310, 1988.
- [158] R. Löhner and C. Yang. Improved ALE mesh velocities for moving bodies. *Communications in Numerical Methods in Engineering*, 12(10):599–608, 1996.
- [159] A. Maier, M. W. Gee, C. Reeps, J. Pongratz, H.-H. Eckstein, and W. A. Wall. A Comparison of Diameter, Wall Stress, and Rupture Potential Index for Abdominal Aortic Aneurysm Rupture Risk Prediction. *Annals of Biomedical Engineering*, 38(10):3124–3134, 2010.
- [160] J. Mandel, M. Brezina, and P. Vaněk. Energy Optimization of Algebraic Multigrid Bases. *Computing*, 62(3):205–228, 1999.
- [161] K. Mani and D. J. Mavriplis. Error estimation and adaptation for functional outputs in time-dependent flow problems. *Journal of Computational Physics*, 229(2):415–440, 2010.
- [162] U. M. Mayer, A. Popp, A. Gerstenberger, and W. A. Wall. 3D fluid–structure–contact interaction based on a combined XFEM FSI and dual mortar contact approach. *Computational Mechanics*, 46(1):53–67, 2010.
- [163] M. Mayr, T. Klöppel, W. A. Wall, and M. W. Gee. A Temporal Consistent Monolithic Approach to Fluid–Structure Interaction Enabling Single Field Predictors. *SIAM Journal on Scientific Computing*, 37(1):B30–B59, 2015.
- [164] M. McCormick, D. Nordsletten, D. Kay, and N. Smith. Simulating left ventricular fluid–solid mechanics through the cardiac cycle under lvad support. *Journal of Computational Physics*, 244(0):80–96, 2013.
- [165] J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric m -matrix. *Mathematics of Computation*, 31(1):148–162, 1977.
- [166] A. Meister. *Numerik linearer Gleichungssysteme*. Springer, 5 edition, 2015.
- [167] R. H. Merson. An operational method for the study of integration processes. In *Proc. Symp. Data Processing, Weapons Research Establishment*, pages 110/1–110/25, Salisbury, Australia, 1957.
- [168] R. Mittal and G. Iaccarino. Immersed Boundary Methods. *Annual Reviews of Fluid Mechanics*, 37:329–261, 2005.
- [169] P. Moireau, N. Xiao, M. Astorino, C. A. Figueroa, D. Chapelle, C. A. Taylor, and J.-F. Gerbeau. External tissue support and fluid–structure simulation in blood flows. *Biomechanics and Modeling in Mechanobiology*, 11(1):1–18, 2012.
- [170] D. P. Mok. *Partitionierte Lösungsansätze in der Strukturdynamik und der Fluid-Struktur-Interaktion*. PhD thesis, Universität Stuttgart, Stuttgart, 2001.
- [171] D. S. Molony, A. Callanan, E. G. Kavanagh, M. T. Walsh, and T. M. McGloughlin. Fluid-structure interaction of a patient-specific abdominal aortic aneurysm treated with an endovascular stent-graft. *BioMedical Engineering OnLine*, 8(1):1–12, 2009.
- [172] R. L. Muddle, M. Mihajlović, and M. Heil. An efficient preconditioner for monolithically-coupled large-displacement fluid–structure interaction problems with pseudo-solid mesh updates. *Journal of Computational Physics*, 231(21):7315–7334, 2012.
- [173] N. M. Newmark. Computation of Dynamic Structural Response in the Range Approaching Failure. In *Proceedings of the Symposium on Earthquake and Blast Effects on Structures*, Los Angeles, 1952. Earthquake Engineering Research Institute.

- [174] N. M. Newmark. A Method of Computation for Structural Dynamics. *Journal of the Engineering Mechanics Division*, 85(3):67–94, 1959.
- [175] I. Newton. *Philosophiae naturalis principia mathematica*. Jussu Societatis Regiae ac Typis Joseph Streater ..., London, 1687.
- [176] F. Nobile and C. Vergara. An Effective Fluid-Structure Interaction Formulation for Vascular Dynamics by Generalized Robin Conditions. *SIAM Journal on Scientific Computing*, 30(2):731–763, 2008.
- [177] M. Noll. Domain Decomposition/Redistribution and Hybrid Additive/Multiplicative Schwarz Preconditioning for Monolithic Fluid-Structure Interaction Solvers. Semesterarbeit, Technische Universität München, 2015.
- [178] A. Oddy, J. Goldak, M. McDill, and M. Bibby. A distortion metric for isoparametric finite elements. *Transactions of the Canadian Society for Mechanical Engineering*, 12(4):213–217, 1988.
- [179] C. W. Oosterlee and T. Wahsio. Krylov Subspace Acceleration of Nonlinear Multigrid with Application to Recirculating Flows. *SIAM Journal on Scientific Computing*, 21(5):1670–1690, 2000.
- [180] K. Park and P. Underwood. A variable-step central difference method for structural dynamics analysis – part 1. Theoretical aspects. *Computer Methods in Applied Mechanics and Engineering*, 22(2):241–258, 1980.
- [181] C. S. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220–252, 1977.
- [182] S. Piperno. Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid sub-cycling for 2d inviscid aeroelastic simulations. *International Journal for Numerical Methods in Fluids*, 25(10):1207–1226, 1997.
- [183] J.-P. Ponthot and T. Belytschko. Arbitrary Lagrangian-Eulerian formulation for element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 152(1–2):19–46, 1998. Containing papers presented at the Symposium on Advances in Computational Mechanics.
- [184] A. Popp. *Mortar Methods for Computational Contact Mechanics and General Interface Problems*. PhD thesis, Technische Universität München, München, 2012.
- [185] 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.
- [186] A. Popp, M. Gitterle, M. W. Gee, and W. A. Wall. A dual mortar approach for 3D finite deformation contact with consistent linearization. *International Journal for Numerical Methods in Engineering*, 83(11):1428–1465, 2010.
- [187] A. Popp, B. I. Wohlmuth, M. W. Gee, and W. A. Wall. Dual Quadratic Mortar Finite Element Methods for 3D Finite Deformation Contact. *SIAM Journal on Scientific Computing*, 34(4):B421–B446, 2012.
- [188] A. Prokopenko, J. J. Hu, T. A. Wiesner, C. M. Siefert, and R. S. Tuminaro. MueLu User’s Guide 1.0. Technical Report SAND2014-18874, Sandia National Laboratories, Albuquerque, NM (USA) 87185, 2014.
- [189] M. A. Puso. A 3D mortar method for solid mechanics. *International Journal for Numerical Methods in Engineering*, 59(3):315–336, 2004.
- [190] M. A. Puso and T. A. Laursen. Mesh tying on curved interfaces in 3D. *Engineering Computations*, 20(3):305–319, 2003.
- [191] M. A. Puso and T. A. Laursen. A mortar segment-to-segment contact method for large deformation solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 193(6–8):601–629, 2004.
- [192] M. A. Puso and T. A. Laursen. A mortar segment-to-segment frictional contact method for large deformations. *Computer Methods in Applied Mechanics and Engineering*, 193(45–47):4891–4913, 2004.
- [193] A. Quarteroni, R. Sacco, F. Saleri, and P. Gervasio. *Matematica Numerica*. Springer, 4th edition, 2014.
- [194] A. Quarteroni, F. Saleri, and P. Gervasio. *Calcolo Scientifico*. Springer, 5a edition, 2012.

-
- [195] A. Quarteroni and A. M. P. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Clarendon, Oxford, 2005.
- [196] M. Raghavan and D. A. Vorp. Toward a biomechanical tool to evaluate rupture potential of abdominal aortic aneurysm: identification of a finite strain constitutive model and evaluation of its applicability. *Journal of Biomechanics*, 33(4):475–482, 2000.
- [197] E. Ramm and W. A. Wall. Shell structures—a sensitive interrelation between physics and numerics. *International Journal for Numerical Methods in Engineering*, 60(1):381–427, 2004.
- [198] D. R. Reynolds, D. J. Gardner, A. C. Hindmarsh, C. S. Woodward, and J. M. Sexton. User Documentation for ARKode v1.0.2 (SUNDIALS v2.6.2). Technical Report LLNL-SM-668082, Lawrence Livermore National Laboratory, 2015.
- [199] L. F. Richardson. The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210:307–357, 1911.
- [200] L. F. Richardson and J. A. Gaunt. The Deferred Approach to the Limit. Part I. Single Lattice. Part II. Interpenetrating Lattices. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 226:299–361, 1927.
- [201] A. Rodríguez-Ferran, F. Casadei, and A. Huerta. ALE stress update for transient and quasistatic processes. *International Journal for Numerical Methods in Engineering*, 43(2):241–262, 1998.
- [202] A. Rodríguez-Ferran, A. Pérez-Foguet, and A. Huerta. Arbitrary Lagrangian–Eulerian (ALE) formulation for hyperelastoplasticity. *International Journal for Numerical Methods in Engineering*, 53(8):1831–1851, 2002.
- [203] I. Romero and L. M. Lacoma. Analysis of error estimators for the semidiscrete equations of linear solid and structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 195(19–22):2674–2696, 2006.
- [204] I. Romero and L. M. Lacoma. A methodology for the formulation of error estimators for time integration in linear solid and structural dynamics. *International Journal for Numerical Methods in Engineering*, 66(4):635–660, 2006.
- [205] M. R. Ross, C. A. Felippa, K. Park, and M. A. Sprague. Treatment of acoustic fluid–structure interaction by localized Lagrange multipliers: Formulation. *Computer Methods in Applied Mechanics and Engineering*, 197(33–40):3057–3079, 2008.
- [206] M. R. Ross, M. A. Sprague, C. A. Felippa, and K. Park. Treatment of acoustic fluid–structure interaction by localized Lagrange multipliers and comparison to alternative interface-coupling methods. *Computer Methods in Applied Mechanics and Engineering*, 198(9–12):986–1005, 2009.
- [207] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.
- [208] Y. Saad and M. H. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.
- [209] M. G. Sala and M. A. Heroux. Robust Algebraic Preconditioners using IFPACK 3.0. Technical Report SAND2005-0662, Sandia National Laboratories, Albuquerque, NM (USA) 87185, 2005.
- [210] M. G. Sala and R. S. Tuminaro. A New Petrov–Galerkin Smoothed Aggregation Preconditioner for Nonsymmetric Linear Systems. *SIAM Journal on Scientific Computing*, 31(1):143–166, 2008.
- [211] F. Saleri and A. Veneziani. Pressure Correction Algebraic Splitting Methods for the Incompressible Navier–Stokes Equations. *SIAM Journal on Numerical Analysis*, 43(1):174–194, 2005.
- [212] D. Sarafyan. Error estimation for Runge–Kutta methods through pseudo-iterative formulas. Technical Report 14, Louisiana State University, New Orleans, 1966.

- [213] J. Sarrate and A. Huerta. An improved algorithm to smooth graded quadrilateral meshes preserving the prescribed element size. *Communications in Numerical Methods in Engineering*, 17(2):89–99, 2001.
- [214] A. Schleupen and E. Ramm. Local and global error estimations in linear structural dynamics. *Computers & Structures*, 76(6):741–756, 2000.
- [215] P. J. G. Schreurs, F. E. Veldpaus, and W. A. M. Brekelmans. Simulation of forming processes, using the arbitrary eulerian-lagrangian formulation. *Computer Methods in Applied Mechanics and Engineering*, 58(1):19–36, 1986.
- [216] H. R. Schwarz and N. Köckler. *Numerische Mathematik*. Vieweg+Teubner Verlag, 8th edition, 2011.
- [217] L. R. Scott and S. Zhang. Finite element interpolation of nonsmooth functions satisfying boundary conditions. *Mathematics of Computation*, 54(190):483–493, 1990.
- [218] S. Shahmiri. *A Hybrid Fixed-Grid-ALE Approach for Fluid-Structure Interaction*. PhD thesis, Technische Universität München, München, 2014.
- [219] L. F. Shampine. Local error control in codes for ordinary differential equations. *Applied Mathematics and Computation*, 3(3):189–210, 1977.
- [220] B. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 2008.
- [221] G. Söderlind. Automatic Control and Adaptive Time-Stepping. *Numerical Algorithms*, 31(1–4):281–310, 2002.
- [222] G. Söderlind. Digital Filters in Adaptive Time-stepping. *ACM Transactions on Mathematical Software*, 29(1):1–26, 2003.
- [223] G. Söderlind. Time-step selection algorithms: Adaptivity, control, and signal processing. *Applied Numerical Mathematics*, 56(3–4):488–502, 2006.
- [224] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Wellesley Cambridge Press, Wellesley, MA, 2nd edition, 2008.
- [225] A. Swillens, L. Lanoye, J. De Backer, N. Stergiopoulos, P. Verdonck, F. Vermassen, and P. Segers. Effect of an Abdominal Aortic Aneurysm on Wave Reflection in the Aorta. *IEEE Transactions on Biomedical Engineering*, 55(5):1602–1611, 2008.
- [226] 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. Incompressible {CFD}.
- [227] 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.
- [228] A. Toselli and O. B. Widlund. *Domain Decomposition Methods: Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer Berlin / Heidelberg, 2005.
- [229] J. Y. Trépanier, M. Reggio, M. Paraschivoiu, and R. Camarero. Unsteady Euler solutions for arbitrarily moving bodies and boundaries. *American Institute of Aeronautics and Astronautics Journal*, 31(10):1869–1876, 1993.
- [230] P. Triccerri, L. Dedè, S. Deparis, A. Quarteroni, A. M. Robertson, and A. Sequeira. Fluid-structure interaction simulations of cerebral arteries modeled by isotropic and anisotropic constitutive laws. *Computational Mechanics*, 55(2):479–498, 2015.
- [231] C. Truesdell and W. Noll. *The Non-Linear Field Theories of Mechanics*. Springer, 3rd edition, 2004.
- [232] A. M. Turing. Rounding-Off Errors in Matrix Processes. *The Quarterly Journal of Mechanics and Applied Mathematics*, 1(1):287–308, 1947.

-
- [233] P. Underwood and K. Park. A variable-step central difference method for structural dynamics analysis – part 2. Implementation and performance evaluation. *Computer Methods in Applied Mechanics and Engineering*, 23(3):259–279, 1980.
- [234] P. Vaněk, M. Brezina, and J. Mandel. Convergence of algebraic multigrid based on smoothed aggregation. *Numerische Mathematik*, 88:559–579, 2001.
- [235] P. Vaněk, M. Brezina, and R. Tezaur. Two-grid Method for Linear Elasticity on Unstructured Meshes. *SIAM Journal on Scientific Computing*, 21(3):900–923, 1999.
- [236] P. Vaněk, J. Mandel, and M. Brezina. Algebraic Multigrid By Smoothed Aggregation For Second And Fourth Order Elliptic Problems. *Computing*, 56:179–196, 1996.
- [237] A. Veneziani and U. Villa. ALADINS: An ALgebraic splitting time ADaptive solver for the Incompressible Navier–Stokes equations. *Journal of Computational Physics*, 238:359–375, 2013.
- [238] F. Verdugo, N. Parés, and P. Díez. Modal-based goal-oriented error assessment for timeline-dependent quantities in transient dynamics. *International Journal for Numerical Methods in Engineering*, 95(8):685–720, 2013.
- [239] F. Verdugo, N. Parés, and P. Díez. Goal-oriented space-time adaptivity for transient dynamics using a modal description of the adjoint solution. *Computational Mechanics*, 54(2):331–352, 2014.
- [240] F. Verdugo and W. A. Wall. Unified computational framework for the efficient solution of n -field coupled problems with monolithic schemes. *Computer Methods in Applied Mechanics and Engineering*, 310:335–366, 2016.
- [241] J. H. Verner. Explicit Runge–Kutta Methods with Estimates of the Local Truncation Error. *SIAM Journal on Numerical Analysis*, 15(4):772–790, 1978.
- [242] W. A. Wall. *Fluid–Struktur–Interaktion mit stabilisierten Finiten Elementen*. PhD thesis, Universität Stuttgart, Institut für Baustatik, Stuttgart, 1999.
- [243] W. A. Wall and M. W. Gee. Baci: A parallel multiphysics simulation environment. Technical report, Institute for Computational Mechanics, Technische Universität München, 2008.
- [244] 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*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 195–232. Springer Berlin Heidelberg, 2006.
- [245] W. A. Wall, A. Gerstenberger, U. Küttler, and U. M. Mayer. *An XFEM Based Fixed-Grid Approach for 3D Fluid-Structure Interaction*, pages 327–349. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [246] T. Washio and C. W. Oosterlee. Krylov subspace acceleration for nonlinear multigrid schemes. *Electronic Transactions on Numerical Analysis*, 6:271–290, 1997.
- [247] P. Wesseling. *An introduction to multigrid methods*. Wiley, Chichester, West Sussex, 1992.
- [248] N.-E. Wiberg and X. D. Li. A post-processing technique and an a posteriori error estimate for the newmark method in dynamic analysis. *Earthquake Engineering & Structural Dynamics*, 22(6):465–489, 1993.
- [249] T. A. Wiesner. *Flexible Aggregation-based Algebraic Multigrid Methods for Contact and Flow Problems*. PhD thesis, Technische Universität München, 2015.
- [250] T. A. Wiesner, R. S. Tuminaro, W. A. Wall, and M. W. Gee. Multigrid transfers for nonsymmetric systems based on Schur complements and Galerkin projections. *Numerical Linear Algebra with Applications*, 21(3):415–438, 2014.
- [251] E. K. Wilhelm. Time Adaptivity in Fluid–Structure Interaction. Bachelor’s Thesis, Technische Universität München, 2013.
- [252] A. M. Winslow. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics*, 1(2):149–172, 1966.

- [253] B. I. Wohlmuth. A Mortar Finite Element Method Using Dual Spaces for the Lagrange Multiplier. *SIAM Journal on Numerical Analysis*, 38(3):989–1012, 2000.
- [254] B. I. Wohlmuth. *Discretization Methods and Iterative Solvers Based on Domain Decomposition*, volume 17. Springer, Heidelberg, 2001.
- [255] P. Wriggers. *Nichtlineare Finite-Element-Methoden*. Springer Berlin Heidelberg, 2001.
- [256] P. Wriggers. *Computational Contact Mechanics*. Springer Berlin / Heidelberg, 2nd edition, 2006.
- [257] R. Wüchner, A. Kupzok, and K.-U. Bletzinger. A framework for stabilized partitioned analysis of thin membrane–wind interaction. *International Journal for Numerical Methods in Fluids*, 54(6–8):945–963, 2007.
- [258] N. Xiao, J. Alastruey, and C. Alberto Figueroa. A systematic comparison between 1-D and 3-D hemodynamics in compliant arterial models. *International Journal for Numerical Methods in Biomedical Engineering*, 30(2):204–231, 2014.
- [259] D. Zeng and C. R. Ethier. A semi-torsional spring analogy model for updating unstructured meshes in 3D moving domains. *Finite Elements in Analysis and Design*, 41(11–12):1118–1139, 2005.
- [260] L. F. Zeng, N.-E. Wiberg, X. D. Li, and Y. M. Xie. A posteriori local error estimation and adaptive time-stepping for newmark integration in dynamic analysis. *Earthquake Engineering & Structural Dynamics*, 21(7):555–571, 1992.
- [261] O. C. Zienkiewicz, R. L. Taylor, and D. D. Fox. *The Finite Element Method for Solid and Structural Mechanics*. Butterworth-Heinemann, Oxford, 7th edition, 2014.
- [262] O. C. Zienkiewicz, R. L. Taylor, and P. Nithiarasu. *The Finite Element Method for Fluid Dynamics*. Butterworth-Heinemann, Oxford, 7th edition, 2014.
- [263] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Methods: Its Basis and Fundamentals*. Butterworth-Heinemann, Oxford, 7th edition, 2013.
- [264] O. C. Zienkiewicz and Y. M. Xie. A simple error estimator and adaptive time stepping procedure for dynamic analysis. *Earthquake Engineering & Structural Dynamics*, 20(9):871–887, 1991.
- [265] J. A. Zonneveld. Automatic integration of ordinary differential equations. Technical Report R743, Mathematisch Centrum, Amsterdam, 1963.