JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis

Numerical Simulation of Fluid-Structure Interaction Problems on Hybrid Meshes with Algebraic Multigrid Methods

Dissertation

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

Angefertigt am *Institut für Numerische Mathematik*

Betreuung:

Prof. Dr. Walter Zulehner, Johannes Kepler University Linz
Prof. Dr. Ulrich Langer, Johannes Kepler University Linz

Eingereicht von:

Dipl.-Ing. Huidong Yang

Externer Begutachter:

Prof. Dr. Peter Bastian, Heidelberg University

Linz, January, 2010

For Valéria
and my parents

Abstract

Fluid-structure interaction problems arise in many application fields such as flows around elastic structures or blood flow problems in arteries. One method for solving such a problem is based on a reduction to an equation at the interface, involving the so-called Steklov-Poincaré operators.

This interface equation is solved by a Newton iteration for which directional derivatives with respect to the interface perturbation have to be evaluated appropriately. One step of the Newton iteration requires the solution of several decoupled linear sub-problems in the structure and the fluid domains.

These sub-problems are spatially discretized by a finite element method on hybrid meshes containing different types of elements. For the time discretization implicit first order methods are used. The discretized equations are solved by algebraic multigrid methods for which a stabilized coarsening hierarchy is constructed in a proper way.

We developed a draft implementation of a grid-enabled solver for this fluid-structure interaction problem. A recently designed Client/Server model under the grid environment is used. The interface equation is solved on the server grid node, while the fluid and the structure sub-problems are solved independently on client grid nodes.

Zusammenfassung

Die Fluid-Struktur-Interaktion (FSI) kommt in vielen Anwendungen vor. Die Strömung um elastische Strukturen in technischen Anwendungen oder der Blutfluss in Arterien seien hier als Beispiele genannt. Eine Methode zur Lösung derartiger FSI-Probleme basiert auf der Reduktion der beschreibenden partiellen Differentialgleichungen auf eine nichtlineare Interface-Gleichung, die so genannte Steklov-Poincaré-Operatoren enthalten.

Diese nichtlineare Interface-Gleichung wird nun mit einem Newton-Verfahren gelöst. Die im Newton-Verfahren benötigten Ableitungen, die so genannte Gebietsableitungen enthalten, sind geeignet zu berechnen. Ein Newton-Schritt erfordert die Lösung verschiedener, entkoppelter, linearer Teilprobleme in der elastischen Struktur und im Strömungsgebiet.

Zur Ortsdiskretisierung dieser verschiedenen Teilprobleme werden spezielle Finite-Elemente-Methoden auf hybriden Netzen entwickelt und analysiert. Diese hybriden Netze enthalten im Allgemeinen verschiedene Elementtypen. Die Zeitdiskretisierung wird mit impliziten Methode der Konsistenzordnung 1 realisiert. Die entstehenden linearen Gleichungssysteme werden mit Algebraischen Mehrgitterverfahren gelöst, wobei strömungsseitig ein spezielles Algebraisches Mehrgitterverfahren mit einer geeigneten Stabilisierungstechnik für die Grobgittermatrizen entwickelt wurde.

Die entwickelten Algorithmen sind bestens für das Grid-Computing geeignet. In der Arbeit wird ein kürzlich entwickeltes Client/Server-Modell innerhalb einer Grid-Umgebung diskutiert und pilotmäßig umgesetzt. Dabei wird die Interface-Gleichung auf dem Master-Node und die Teilprobleme im Fluidgebiet und im Strukturgebiet unabhängig voneinander auf verschiedenen Slave-Nodes des Grids behandelt.

Acknowledgments

First of all I own my deep gratitude to my advisor Walter Zulehner, for advising my thesis, arousing my interest, and deepening my understanding in this field. Without this, I am not able to finish my thesis in time. I would like to thank Ulrich Langer, the leader of the project and head of the institute I am working in, for providing me with scientific and financial support, offering me many constructive suggestions on my work, and encouraging me to go forward. At the same time, I would like to thank Peter Bastian for refereeing this thesis.

Special thanks go to Ferdinand Kickinger who supplies me with numerical test cases and lots of helpful suggestions on handling hybrid meshes with kind patience all the time. Meanwhile, I am indebted to Markus Wabro for his many helps on the algebraic multigrid solvers.

Many thanks go to David Pusch, Veronika Pillwein, Johannes Kraus, Erwin Karer, Satyendra Tomar, Joachim Schöberl, Clemens Pechstein, Astrid Sinwel, Sabine Zaglmayr, Markus Baumgartner, Tian Li, Gundolf Haase, Maximilian Liebmann for their kind help during this time period.

Sincere thanks go to Valéria Limpók for her love, and my parents for their supporting me through all the time.

Last, but not least, I gratefully acknowledge the Austrian Grid project funded by the BMWF (Federal ministry of Science and Research), the project "Data-sparse Boundary and Finite Element Domain Decomposition Methods in Electromagnetics" supported by the Austrian Science Fund (FWF) under grant P19255-N18, and the Doctoral Program (DK) "Computational Mathematics" (W1214) granted by FWF.

Huidong Yang
Linz, Austria
2009.09.13

Contents

Chapter 1

Introduction

1.1 Overview

1.1.1 Fluid-structure interaction solvers

Fluid-structure interaction (FSI) describes a large range of physical problems from aeroelastic problems, such as airflows around rigid or deformable structures, to haemodynamics, for instance blood flow in large arteries. More and more interests are arising in this class of problems in many technical applications since a long time (see, e.g. [118, 46, 48, 108, 65, 115, 149, 145, 117, 150, 175, 165, 56]). Recently, FSI simulations have been successfully used in life science as well. For instance, blood flow simulations are among the most interesting and challenging applications in this field (see, e.g. [128, 129, 130, 43, 52, 126]).

To solve fluid-structure interaction problems, two main strategies are studied recently. The first one is the so-called monolithic method, see recent examples of such an approach, for instance, in [136, 76, 75, 13, 54, 16, 104]. In this method, the structure and fluid sub-problems, the boundary conditions, and the interface conditions are all included in one large system of equations. Solvers for this system are usually less modular than solvers based on a partitioned approach, which will be discussed next.

In order to reuse existing structure and fluid solvers which are usually developed and available at hand, another popular method for solving the interaction problem, the so-called partitioned (segregated) approach were investigated (see e.g. [104, 114, 52, 43, 15]). This partitioned approach is based on subsequent solutions of the structure and fluid sub-problems and allows the use of specially designed existing codes for the structural and fluid fields.

In this thesis, we will use the second approach. For the structure and fluid sub-problems algebraic multigrid solvers were developed. Furthermore, the partitioned approach is very well suited for grid computing which will be stressed in Subsection 1.1.4.

In aeroelastic simulations (see [49, 48, 124]), where the structure density is much greater than that of the fluid, loosely coupled (explicit) algorithms are successfully used to achieve convergence (see [59, 122, 123]). This approach solves the structure problem and the fluid problem only once (or just a few times) per time step. However, in the blood flow simulations where the structure and the fluid densities are of the same order and if the domain has slender shape (see [38]), a major drawback of the loosely coupled method in terms of accuracy and stability problems can occur due to the so called added-mass effect (see [38]).

The stability of numerical simulations of fluid-structure interactions for the blood flow

relies heavily on the accuracy in solving the nonlinear coupled problems at each time step (see [44, 109, 108, 118, 148, 43, 115]). Under this situation, numerical experiments show that this requires fully (strongly) coupled schemes in order to ensure stability of the methods, see [44, 52, 43].

Simple strategies to solve the resulting nonlinear problems in each time step are fixed-point based methods (see [10, 39]). However, these methods suffer from very slow convergence and in some cases may fail to converge, see [115, 116, 117, 70].

Recently, the use of Newton-type methods is suggested for their fast convergence in [52, 42, 107]. All these methods are based on the evaluation of the Jacobian (see [149]) associated with the fluid-solid coupled state equations. The critical step consists in the evaluation of the Jacobian (see [149]), which expresses the sensitivity of the fluid state to solid motions. This evaluation can be approximated using finite difference approximation of derivatives (see [149]) or other simple operators (see [149, 65, 42, 66]). In both cases, such approximation might seriously deteriorate the convergence rate, see [149]. A Newton method with exact Jacobian has been investigated theoretically and experimentally in [52], based on shape sensitivity calculus (see [144]). This method shows for some model cases the superiority against approximated versions of Newton's method, see [52].

1.1.2 Extension of linear elements for sub-problems on hybrid meshes

This partitioned approach requires robust solvers (see, e.g. [132, 161]) for the structure and fluid sub-problems on sub-domains which are triangulated by Octree based hybrid meshes (see [89, 81, 147, 87, 86, 8]) arising mainly for computational fluid dynamics applications. The finite volume discretization on such hybrid meshes and its application to incompressible Navier-Stokes equations and fluid-structure interaction problems have been investigated in [8, 87].

In this work, we investigate the finite element discretization based on these hybrid meshes and its application to the structure and the fluid sub-problems. In particular, we extend the standard P_1 linear element on pure tetrahedral meshes to an extended linear element (macro-element) on hybrid meshes which consists of four elements types: tetrahedron, hexahedron, prism and pyramid.

It turns out that the finite element space we constructed on the hybrid mesh is a subset of the space spanned by standard hat functions. The error estimates can be shown by properly chosen interpolation operators. This extended linear element can be applied to the structure sub-problem in a straightforward way.

However, for the fluid sub-problem, we are not able to directly apply this extended linear element for the velocity and pressure components due to two sources of instability: the discrete inf-sup condition (see [35]) and the dominating convection (see [23]).

In order to overcome the inf-sup instability, we use this extended P_1 element for the velocity and pressure spaces, enriched with sufficiently many bubble functions for the velocity space. In this work, we have analyzed this method by explicitly constructing the so-called Fortin operator (see [35]) on the hybrid mesh. The additional degrees of freedom are eliminated by static condensation, which, for simplicity, is presented only for the Stokes problem.

For the implementation, we adopted another approach, namely, a combination of the pressure stabilization Petrov-Galerkin method (PSPG) and the streamline upwind Petrov-Galerkin method (SUPG), which overcome the instability caused by those two effects, re-

spectively. This approach has been introduced and applied to the Oseen and Navier-Stokes problems in [36, 133, 82, 23, 103, 152].

1.1.3 Algebraic multigrid solvers for sub-problems

In order to solve the large linear(ized) systems arising from the stabilized finite element discretization, we will apply algebraic multigrid (AMG) methods. For an overview of the general technique and various applications we refer, for example, to [132, 145, 146, 135, 90, 153]. The AMG for solving the structure problem can be found in [156, 94, 69, 153, 95].

For the application of AMG to saddle point problems, the first trial is the segregated approach, i.e. to use a classical method (Uzawa, SIMPLE, etc) for an outer iteration and to apply AMG method to the resulting elliptic problems (see [145, 68]). A second approach is the coupled or all-at-once approach where an AMG method for the whole saddle point system must be developed (see [161, 163, 162]) which can compete with or is even superior to classical approaches. The related technique has been investigated in [161, 163, 162, 167, 131, 22, 6, 143]. In particular, in [161], M. Wabro investigated this coupled AMG technique applied to the numerical solution of the incompressible Navier-Stokes equations.

If one applies the coarsening strategy using black-box AMG solvers to the whole system in a straightforward way, it will lead to a mixture of velocity and pressure components on coarse levels. Therefore, an important feature of the coupled AMG for the saddle point problem is to coarsen the velocity and pressure unknowns separately. This issue has been addressed in [161, 163, 162].

Another key point in the coupled AMG method for the saddle point problem, in particular, for the fluid problem, is to construct (only with pure algebraic information) stabilized systems which satisfy the inf-sup condition on all coarser levels. In [161, 163, 162], M. Wabro has shown how to construct stabilized hierarchy for $P_1\text{iso}P_2$-P_1 element, MINI-element and P_1^{nc}-P_0 element.

In this thesis, we extend the ideas from the previous work of M. Wabro and show how to construct a stabilized P_1-P_1 hierarchy for the AMG solver of the fluid sub-problem on the hybrid mesh.

1.1.4 Simulations on grid computing environment

The partitioned approach for solving the fluid structure interaction problem is well suitable to the *grid computing* (see [58, 51, 57, 41, 125, 112, 160, 28]) since this method is based on efficient, robust and fast solvers for each of the sub-problems (fluid and structure), which is the main cost of this type of algorithms and can be distributed and parallelized to many processors under the grid environment, see [99, 100].

Since the early 1990s, the term *grid computing* has appeared in Ian Foster's and Carl Kesselman's seminal work, "The Grid: Blueprint for a new computing infrastructure", see [3]. Grids offer a way to solve Grand Challenge problems such as protein folding, financial modeling, earthquake simulation, and climate modeling.

Under the grid environment, the large scale numerical simulations can be performed in a more efficient way because of its huge computational and memory storage resources, e.g. its applications in computational fluid dynamics (CFD), see [173, 102, 98], in artery flow dynamics, see [128, 129, 130], in human eye simulations, see [106, 166, 113, 37] and in fluid-structure interaction problem, see [99, 100, 97, 101].

In this work, we developed a grid-enabled solver using the numerical method proposed in Chapter 3 for the fluid-structure interaction problem under the Austrian Grid environment (see [1, 20, 134]), see details in Section 6.4.

1.2 Outline

The thesis is organized as follows.

Chapter 2 contains the preliminaries concerning the functional analysis and some results on Sobolev spaces which are needed in the next chapters. All these results are standard from the textbook, and hence no proof are provided for them.

In Chapter 3, a domain decomposition based Newton method for solving the interface equation, the so-called Steklov-Poincaré equation, is described in detail. This requires evaluation of the derivatives of introduced structure and fluid operators.

Chapter 4 deals with the extension of the P_1 element to hybrid meshes. This extended element will be applied to each FSI sub-problem. For the fluid sub-problem, due to instabilities caused from the violation of the discrete inf-sup condition (see [35]) and the dominating convection (see [23]), special treatment must be taken into account. We describe two methods to overcome these difficulties.

Chapter 5 is devoted to the construction of efficient AMG solvers for the structure and fluid sub-problems arising from the fluid-structure interaction problem. The main task of this chapter is to construct a stabilized P_1-P_1 hierarchy on hybrid meshes by proper scaling on some sub-blocks from the saddle point system.

Chapter 6 presents some numerical results. Their task is to show numerically the behaviour of the discretization error on the hybrid meshes, the robustness of AMG solvers for the sub-problems, and the Newton convergence for the FSI simulation, which illustrate the theoretical results from previous chapters. As an application of grid computing to the fluid-structure interaction simulation, we described a grid-enabled solver for the fluid-structure interaction problem, for which a recently designed Client/Server (CS) model under the grid computing environment is presented. We use the numerical method from Chapter 3 where the whole problem is reduced to solving the interface equation requiring solutions of the fluid and the structure sub-problems independently (i.e. in parallel) solved on the corresponding client grid node.

Chapter 2

Preliminaries

This chapter contains the preliminaries for the thesis. Note, throughout the thesis, we will not distinguish between scalar values (functions) and vector values (functions) in characters. For matrices we use capital letters $(A, ...)$ or component-notation $A = (a_{ij})$.

For a comprehensive introduction to functional analysis and Sobolev spaces we refer to [7, 174]. We only mention some spaces and the related properties which are needed for the proofs in the next chapters.

We assume that set $\Omega \subset \mathbb{R}^d, d \geq 2$ is an open and bounded domain with Lipschitz boundary $\Gamma = \partial\Omega$.

2.1 Operators

For vectors $u = (u_1, ..., u_d)^T$ and $v = (v_1, ..., v_d)^T$, we introduce the following products:

- scalar product: $u \cdot v = \sum_{i=1}^{d} u_i v_i$,
- tensor product: $u \bigotimes v = (u_i v_j)_{i,j=1,...,d}$.

For vector fields $u = (u_1, ..., u_d)^T : \Omega \to \mathbb{R}^d$ and $v = (v_1, ..., v_d)^T : \Omega \to \mathbb{R}^d$, we introduce the following differential operators:

- gradient: $\nabla u = \left(\frac{\partial u_i}{\partial x_j}\right)_{i,j=1,...,d}$,
- divergence: $\mathrm{div} u = \sum_{i=1}^{d} \frac{\partial u_i}{\partial x_i}$,
- convection: $(u \cdot \nabla)v = \left(\sum_{j=1}^{d} u_j \frac{\partial u_i}{\partial x_j}\right)_{i=1,...,d}$,
- vector Laplace: $\Delta u = \mathrm{div}(\nabla u) = \left(\sum_{j=1}^{d} \frac{\partial u_i}{\partial x_j}\right)_{i=1,...,d}$.

2.2 Function spaces

Let $L^p(\Omega)$ denote the space of all measurable functions on Ω whose p^{th} power is integrable, equipped with norm

$$\|u\|_{L^p(\Omega)} := \begin{cases} \left(\int_\Omega |u(x)|^p dx\right)^{\frac{1}{p}}, & \text{if } 1 \le p < \infty, \\ \operatorname*{ess\,sup}_{x\in\Omega} |u(x)|, & \text{if } p = \infty. \end{cases} \tag{2.2.1}$$

If u and v are in the space of locally integrable functions for some open set Ω, we say that v is the α^{th}-weak derivative of u if

$$\int_\Omega u D^\alpha \varphi dx = (-1)^{|\alpha|} \int_\Omega v\varphi dx \tag{2.2.2}$$

for all $\varphi \in C_0^\infty(\Omega)$, i.e. for all infinitely differentiable functions φ with compact support in Ω.

The Sobolev space $W_p^k(\Omega)$ is defined as the vector space of functions whose weak derivatives up to a given order k are in $L^p(\Omega)$, equipped with the Sobolev norm $\|\cdot\|_{W_p^k(\Omega)}$ for $k \in \mathbb{N}_0$

$$\|u\|_{W_p^k(\Omega)} := \begin{cases} \left(\sum_{|\alpha|\le k} \|D^\alpha u\|_{L^p}^p\right)^{\frac{1}{p}}, & \text{if } 1 \le p < \infty, \\ \max_{|\alpha|\le k} \|D^\alpha u\|_{L^\infty(\Omega)}, & \text{if } p = \infty, \end{cases} \tag{2.2.3}$$

which consists of the L^p norms of the function itself as well as its weak derivatives up to a given order k. Here $\alpha = (\alpha_1, \alpha_2, ..., \alpha_d)$ is a multiindex and $|\alpha| = \alpha_1 + \alpha_2 + ... + \alpha_d$.

The Sobolev seminorm $|\cdot|_{W_p^k(\Omega)}$ for $k \in \mathbb{N}_0$ consists of the L^p norms of the highest order derivatives,

$$|u|_{W_p^k(\Omega)} := \begin{cases} \left(\sum_{|\alpha|= k} \|D^\alpha u\|_{L^p}^p\right)^{\frac{1}{p}}, & \text{if } 1 \ge p < \infty, \\ \max_{|\alpha|= k} \|D^\alpha u\|_{L^\infty(\Omega)}, & \text{if } p = \infty. \end{cases} \tag{2.2.4}$$

For a Lipschitz domain Ω, we have the following property:

$$W_p^k(\Omega) := \overline{C^\infty(\bar\Omega)}^{\|\cdot\|_{W_p^k(\Omega)}}, \tag{2.2.5}$$

where $\overline{C^\infty(\bar\Omega)}^{\|\cdot\|_{W_p^k(\Omega)}}$ denotes the closure of $C^\infty(\bar\Omega)$ with respect to the norm $\|\cdot\|_{W_p^k(\Omega)}$ for $k \in \mathbb{N}_0$.

In particular, for $p = 2$, $W_2^k(\Omega)$ is a Hilbert space, also denoted by $H^k(\Omega)$, with the inner product

$$\langle u, v\rangle_{H^k(\Omega)} := \sum_{|\alpha|\le k} \int_\Omega D^\alpha u(x) D^\alpha v(x) dx. \tag{2.2.6}$$

To derive some approximation properties of piecewise polynomial spaces, we need the following result, see [33, 40].

Lemma 2.1 (Bramble-Hilbert Lemma)**.** *For $k \in \mathbb{N}_0$, let $l(\cdot)$ be a linear functional on $W_2^{k+1}(\Omega)$ satisfying*

$$|l(v)| \leq c_f \|v\|_{W_2^{k+1}(\Omega)} \text{ for all } v \in W_2^{k+1}(\Omega), \tag{2.2.7}$$

i.e. $l(\cdot)$ belongs to the dual space $[W_2^{k+1}(\Omega)]^*$ of all linear and bounded (continuous) functions. If additionally, $l(v) = 0$ holds for all $v \in P_k$, where $P_k(\Omega)$ denotes the spaces of all polynomials of degree less than or equal to k defined in Ω, then we have

$$|l(v)| \leq c(c_p) c_f |v|_{W_2^{k+1}(\Omega)} \tag{2.2.8}$$

for all $v \in W_2^{k+1}(\Omega)$, where the constant $c(c_p)$ depends only on the constant c_p of the Poincaré inequality (see also Theorem 2.3)

$$\int_\Omega |v(x)|^2 dx \leq c_p \left(\left[\int_\Omega v(x) dx \right]^2 + \int_\Omega |\nabla v(x)|^2 dx \right). \tag{2.2.9}$$

2.3 Inequalities

The Cauchy's inequality for vectors is given by

$$\left(\sum_{k=1}^n a_k b_k \right)^2 \leq \left(\sum_{k=1}^n a_k^2 \right) \left(\sum_{k=1}^n b_k^2 \right) \tag{2.3.1}$$

where equality holds for $a_k = c b_k$ with $c \in \mathbb{R}$. It can be written in vector form as

$$|a \cdot b| \leq \|a\| \|b\|, \tag{2.3.2}$$

where $\|\cdot\|$ denotes the Euclidean norm.

The Cauchy-Schwarz inequality in $L^2(\Omega)$ reads

$$\left| \int_\Omega u(x) v(x) dx \right| \leq \|u\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \tag{2.3.3}$$

for $u, v \in L^2(\Omega)$.

The arithmetic-geometric-mean inequality (ε-inequality) is as follows

$$xy \leq \frac{1}{2} \left(\frac{x^2}{\varepsilon} + \varepsilon y^2 \right). \tag{2.3.4}$$

The Friedrichs' inequality is stated in the following theorem (see [62, 5])

Theorem 2.2 (Friedrichs' inequality)**.** *Let Ω be a bounded Lipschitz domain and let Γ_D be a connected part of $\partial\Omega$ with positive surface measure. Then there exists a constant $c_F > 0$ depending only on diameter of Ω and Γ_D such that for all $u \in [H^1(\Omega)]^d$ with $u_{|\Gamma_D} = 0$ we have*

$$\|u\|_{L^2(\Omega)} \leq c_F |u|_{H^1(\Omega)}. \tag{2.3.5}$$

From this inequality, we easily obtain

$$\|u\|_{H^1(\Omega)} \leq \sqrt{1 + c_F^2}\, |u|_{H^1(\Omega)} \tag{2.3.6}$$

for $u \in [H^1(\Omega)]^d$ with $u_{|\Gamma_D} = 0$, i.e. on $[H^1_D(\Omega)]^d = \{u \in [H^1(\Omega)]^d : u|_{\Gamma_D} = 0\}$, the seminorm $|\cdot|_{H^1(\Omega)}$ and the norm $\|\cdot\|_{H^1(\Omega)}$ are equivalent.

The Poincaré's inequality is as follows (see [33]).

Theorem 2.3 (Poincaré's inequality)**.** *Let Ω be a bounded Lipschitz domain. Then there exists a constant $c_P > 0$ depending on diameter of Ω such that for all $u \in [H^1(\Omega)]^d$ with $\int_\Omega u dx = 0$, we have*

$$\|u\|_{L^2(\Omega)} \leq c_P |u|_{H^1(\Omega)}. \tag{2.3.7}$$

Chapter 3

A Newton based solver for FSI problems

3.1 Introduction

Two main strategies for solving fluid-structure interaction problems have been studied recently. One is to simultaneously solve the fluid and the structure problems with a unique solver. This method is the so-called monolithic method (see [104]). Recent examples of such an approach can be found, for instance, in [136, 76, 75, 13, 54, 16, 17]. Another strategy is the so-called partitioned (segregated) approach (see e.g. [114, 52, 43]) which is based on subsequent solutions of the fluid and structure sub-problems and allows the use of existing codes for the fluid and structural fields. Note that the monolithic approach typically needs a global solver which is less modular than a solver based on a partitioned approach (see [104]). We will adopt the second approach in order to be able to apply the algebraic multigrid solvers to each sub-problems for which they are better understood. The algebraic multigrid solvers will be discussed in Chapter 5.

One of the partitioned methods is the so called loosely coupled (explicit) algorithm which has been successfully used in aeroelasticity (see [49] and the references therein). This approach solves the structure and fluid sub-problems only once (or just a few times) per time step and does not satisfy exactly the coupling transmission conditions to achieve convergence [15, 59, 122, 123, 43]. However in our case, where the structure and fluid densities are of the same order and the structure domain has a slender shape, the fully coupling method is used for the sake of accuracy and stability (see [38]). In this situation, numerical experiments show that only fully coupled schemes can ensure the stability (see [109, 108, 44, 115]). Thus in the end, we have to solve a coupled highly nonlinear system using efficient methods that preserve the fluid-structure sub-system splitting, see [52].

Simple strategies to solve these non-linear problems may fail to converge, see [44, 65, 115]. Recent advances have been achieved by using Newton's method, which requires the evaluation of the Jacobian of the system, see [52, 43, 171, 172, 170]. We will follow this approach.

3.2 Problem setting

In order to describe the interaction between the fluid and structure domains in time under relatively large displacement, we adopt a purely Lagrangian framework for the structure problem and the arbitrary Lagrangian Eulerian (ALE) formulation (see [55]) for the fluid problem.

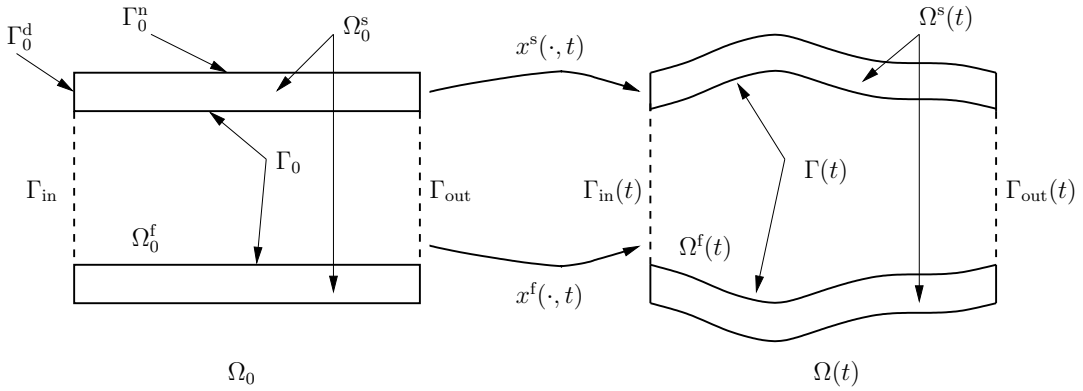


Figure 3.1: An injective mapping x.

3.2.1 Geometrical description and the ALE mapping

Let Ω_0 denote the initial domain at time $t = 0$ consisting of the structure and fluid sub-domains Ω_0^{s} and Ω_0^{f}, respectively. Γ_0^{d} and Γ_0^{n} denote the boundaries of the structure domain with Dirichlet and Neumann boundary conditions, respectively. Γ_{in} and Γ_{out} denote the initial in-flow and out-flow boundaries of the fluid domain, respectively. Γ_0 is the initial interface between the two sub-domains. The domain $\Omega(t)$ at a time t is composed of the deformable structure sub-domain $\Omega^{\mathrm{s}}(t)$ and the fluid sub-domain $\Omega^{\mathrm{f}}(t)$. $\Gamma_{\mathrm{in}}(t)$ and $\Gamma_{\mathrm{out}}(t)$ denote the boundaries of the fluid domain with in-flow and out-flow boundary conditions, respectively. The corresponding interface $\Gamma(t)$ is evolving from the initial interface Γ_0. The evolution of $\Omega(t)$ is obtained by an injective mapping (Figure 3.1):

$$x : \Omega_0 \times R^+ \to R^3. \tag{3.2.1}$$

The position of a point $x_0 \in \Omega_0^{\mathrm{s}}$ at a time t is given by the mapping for the structure domain

$$x_t^{\mathrm{s}} : \Omega_0^{\mathrm{s}} \to \Omega^{\mathrm{s}}(t) \tag{3.2.2}$$

with $x_t^{\mathrm{s}}(x_0) \equiv x^{\mathrm{s}}(x_0, t) = x(x_0, t) = x_0 + d^{\mathrm{s}}(x_0, t)$ for $x_0 \in \Omega_0^{\mathrm{s}}$, where $d^{\mathrm{s}}(x_0, t)$ denotes the displacement of the structure domain at a time t. It can be seen that this mapping x_t^{s} is straightforwardly determined by the displacement of the solid medium with respect to the reference configuration.

Correspondingly, the position of any point $x_0 \in \Omega_0^{\mathrm{f}}$ at a time t is given by the ALE mapping for the fluid domain

$$x_t^{\mathrm{f}} : \Omega_0^{\mathrm{f}} \to \Omega^{\mathrm{f}}(t) \tag{3.2.3}$$

with $x_t^{\mathrm{f}}(x_0) \equiv x^{\mathrm{f}}(x_0,t) = x(x_0,t) = x_0 + d^{\mathrm{f}}(x_0,t)$ for $x_0 \in \Omega_0^{\mathrm{f}}$, where $d^{\mathrm{f}}(x_0,t)$ denotes the displacement of the fluid domain. The displacement is defined as an extension of the structure displacement d^{s} at the interface Γ_0:

$$d^{\mathrm{f}} = \mathrm{Ext}(d^{\mathrm{s}}|_{\Gamma_0}). \tag{3.2.4}$$

A classical choice is to use a harmonic extension in the reference domain:

Problem 3.1. *Find solution $d^{\mathrm{f}} := d^{\mathrm{f}}(x_0,t)$ such that*

$$\begin{cases} -\Delta d^{\mathrm{f}} = 0 & \text{in } \Omega_0^{\mathrm{f}}, \\ d^{\mathrm{f}} = d^{\mathrm{s}} & \text{on } \Gamma_0, \\ d^{\mathrm{f}} = 0 & \text{on } \Gamma_{\mathrm{in}} \cup \Gamma_{\mathrm{out}}. \end{cases} \tag{3.2.5}$$

Furthermore, we introduce the domain velocities

$$w^{\mathrm{s}} := \frac{\partial d^{\mathrm{s}}}{\partial t} = \frac{\partial x^{\mathrm{s}}}{\partial t} \text{ and } \hat{w}^{\mathrm{f}} := \frac{\partial d^{\mathrm{f}}}{\partial t} = \frac{\partial x^{\mathrm{f}}}{\partial t},$$

the deformation gradients

$$F(d^{\mathrm{s}}) := \nabla_{x_0} d^{\mathrm{s}} = \nabla_{x_0} x^{\mathrm{s}} \text{ and } F(d^{\mathrm{f}}) := \nabla_{x_0} d^{\mathrm{f}} = \nabla_{x_0} x^{\mathrm{f}},$$

and the Jacobian determinants

$$J(d^{\mathrm{s}}) := \mathrm{det}F(d^{\mathrm{s}}) \text{ and } J(d^{\mathrm{f}}) := \mathrm{det}F(d^{\mathrm{f}})$$

for the structure and fluid domains, respectively.

3.2.2 Structure modeling

The Lagrange formulation of the pure displacement model of linearized elasticity is defined

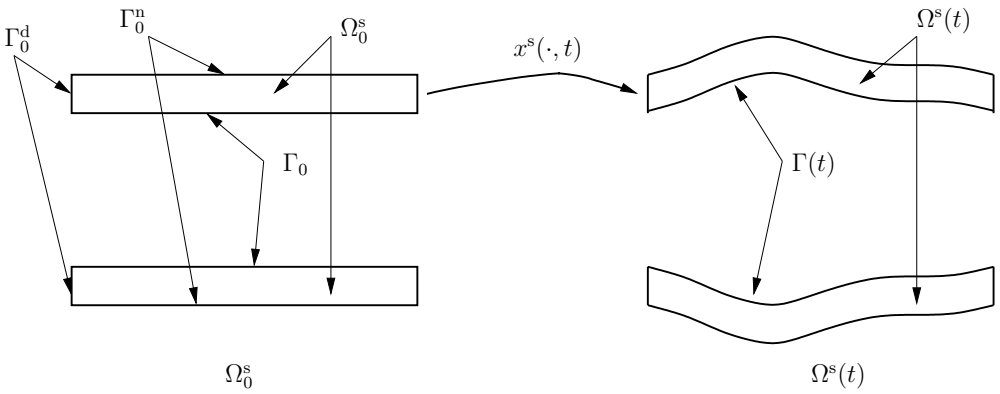


Figure 3.2: Structure deformation

in the reference material configuration Ω_0^{s} (Figure 3.2). The state variable d^{s} satisfies the momentum balance law

$$\rho_{\mathrm{s}} \frac{\partial^2 d^{\mathrm{s}}}{\partial t^2} - \mathrm{div}(\sigma_{\mathrm{s}}(d^{\mathrm{s}})) = f_{\mathrm{s}} \qquad \text{in } \Omega_0^{\mathrm{s}}, \tag{3.2.6}$$

and the boundary conditions

$$\begin{aligned} \sigma_{\mathrm{s}}(d^{\mathrm{s}})n_{\mathrm{s}} &= 0 \qquad \text{on } \Gamma_0^{\mathrm{n}}, \\ d^{\mathrm{s}} &= 0 \qquad \text{on } \Gamma_0^{\mathrm{d}}, \end{aligned} \tag{3.2.7}$$

where ρ_{s} is the density, σ_{s} is the first Piola-Kirchoff stress tensor, f_{s} the external force density, and n_{s} the outward normal of Ω_0^{s}.

We use the linearized elasticity model for the structure problem, in particular, the linear Saint-Venant Kirchoff elastic model:

$$\begin{cases} \sigma_{\mathrm{s}}(d^{\mathrm{s}}) = 2\mu^l \varepsilon(d^{\mathrm{s}}) + \lambda^l \mathrm{div}(d^{\mathrm{s}})I, \\ \varepsilon(d^{\mathrm{s}}) = \dfrac{1}{2}\left(\nabla d^{\mathrm{s}} + (\nabla d^{\mathrm{s}})^T\right) \end{cases} \tag{3.2.8}$$

with Lamé constants λ^l and μ^l. The first line of (3.2.8) is Hooke's law which relates the strain tensor ε and the Cauchy stress tensor σ_{s}. The Lamé constants can be expressed in terms of Young's modulus E and Poisson's ratio ν:

$$\begin{cases} \lambda^l = \dfrac{E\nu}{(1+\nu)(1-2\nu)}, \\ \mu^l = \dfrac{E}{2(1+\nu)}. \end{cases} \tag{3.2.9}$$

Other models could be chosen for the structure depending on particular problems at hand (see [127]).

3.2.3 Fluid modeling

The ALE mapping for the fluid domain is illustrated in Figure 3.3. The system of equations

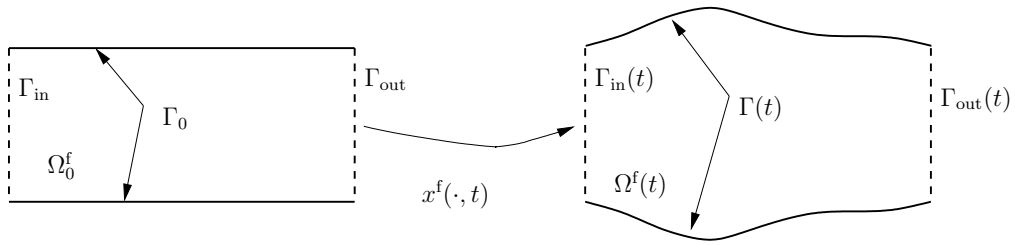


Figure 3.3: Fluid domain deformation

for the incompressible fluid problem under the Eulerian framework is obtained from the balance law of momentum

$$\rho_{\mathrm{f}}\frac{\partial u}{\partial t} + \rho_{\mathrm{f}}(u\cdot\nabla)u - 2\mu\mathrm{div}\varepsilon(u) + \nabla p = 0 \qquad \text{in } \Omega^{\mathrm{f}}(t), \tag{3.2.10}$$

mass conservation

$$\mathrm{div}u = 0 \qquad \text{in } \Omega^{\mathrm{f}}(t), \tag{3.2.11}$$

and properly chosen boundary conditions

$$\begin{aligned} \sigma_{\mathrm{f}}(u,p)n_{\mathrm{f}} &= g_{\mathrm{in}} \qquad \text{on } \Gamma_{\mathrm{in}}(t), \\ \sigma_{\mathrm{f}}(u,p)n_{\mathrm{f}} &= 0 \qquad \text{on } \Gamma_{\mathrm{out}}(t), \end{aligned} \tag{3.2.12}$$

where ρ_{f} is the fluid density, μ the dynamic viscosity, n_{f} the outward normal of $\Omega^{\mathrm{f}}(t)$, and

$$\begin{cases} \sigma_{\mathrm{f}}(u,p) = -pI + 2\mu\varepsilon(u), \\ \quad \varepsilon(u) = \dfrac{1}{2}\left(\nabla u + (\nabla u)^{\mathrm{T}}\right), \end{cases} \tag{3.2.13}$$

for the Cauchy stress tensor σ_{f} and the strain rate tensor ε, respectively.

The ALE time derivative of $u(x,t)$ is introduced in order to overcome the difficulty for evaluating the time derivative of velocity $u(x,t)$ under the Eulerian framework in a moving domain. Let $x \in \Omega^{\mathrm{f}}(t)$, then the ALE time derivative is given by

$$\left.\frac{\partial u}{\partial t}\right|_{x_0} = \frac{\partial}{\partial t}\left(u \circ x_t^{\mathrm{f}}\right) \circ \left(x_t^{\mathrm{f}}\right)^{-1}. \tag{3.2.14}$$

Proposition 3.2. *Let $x \in \Omega^{\mathrm{f}}(t)$. Then*

$$\left.\frac{\partial u}{\partial t}\right|_{x_0}(x,t) = \frac{\partial u}{\partial t}(x,t) + (w^{\mathrm{f}}(x,t)\cdot\nabla)u(x,t), \tag{3.2.15}$$

where $w^{\mathrm{f}}(x,t) = \hat{w}^{\mathrm{f}} \circ \left(x_t^{\mathrm{f}}\right)^{-1}(x)$.

Proof. Use the chain rule. □

Remark 3.3. *If the fluid domain does not change in time, i.e. $w^{\mathrm{f}} = 0$, the ALE derivative coincides with the classical Eulerian derivative $\frac{\partial u}{\partial t}(x,t)$. If the mapping x_t^{f} tracks the fluid particles motion, i.e. $w^{\mathrm{f}} = u$, then the ALE derivative coincides with the material or Lagrangian derivative $\frac{\partial u}{\partial t}(x,t) + u(x,t)\cdot\nabla u(x,t)$. When using the harmonic extension (3.2.5), only the particles motion at the interface Γ_0 is tracked, but the fluid domain displacement inside $\Omega^{\mathrm{f}}(t)$ does not necessarily track particle motions.*

Using (3.2.15), we obtain the ALE formulation for the fluid problem. Velocity $u : \Omega^{\mathrm{f}}(t) \to \mathbb{R}^3$ and pressure $p : \Omega^{\mathrm{f}}(t) \to \mathbb{R}$ satisfy the momentum equation

$$\rho_{\mathrm{f}}\left.\frac{\partial u}{\partial t}\right|_{x_0} + \rho_{\mathrm{f}}\left((u - w^{\mathrm{f}})\cdot\nabla\right)u - 2\mu\mathrm{div}\varepsilon(u) + \nabla p = 0 \qquad \text{in } \Omega^{\mathrm{f}}(t). \tag{3.2.16}$$

3.2.4 Interface equations

When coupling the two sub-problems together, interface conditions are needed. In particular, no-slip conditions at the interface (see Figure 3.4) are explicitly imposed at time t on Γ_0 between the structure and fluid domains:

$$\left.u \circ x_t^{\mathrm{f}}\right|_{\Gamma_0} = \left.\frac{\partial d^{\mathrm{s}}}{\partial t}\right|_{\Gamma_0}. \tag{3.2.17}$$

The second interface condition is the equilibrium of normal stresses:

$$(\sigma_{\mathrm{f}}(u,p)n_{\mathrm{f}}) \circ x_t^{\mathrm{f}} + \sigma_{\mathrm{s}}(d^{\mathrm{s}})n_{\mathrm{s}} = 0. \tag{3.2.18}$$

To summarize, the complete model consists of problem (3.2.5), equations (3.2.6), (3.2.16), (3.2.11), boundary conditions (3.2.7), (3.2.12), and interface conditions (3.2.17), (3.2.18) for the state variables $d^{\mathrm{s}}, u, p, d^{\mathrm{f}}$.

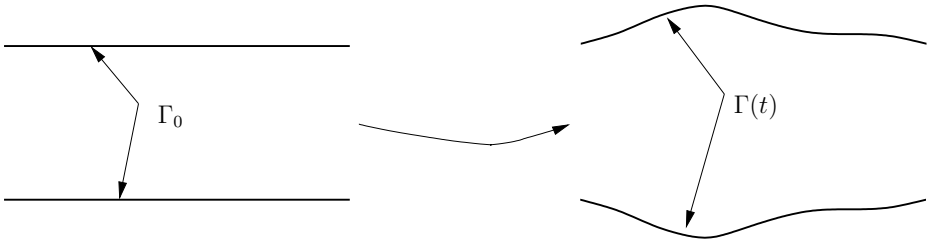


Figure 3.4: The interface changing with time t.

3.2.5 Reformulation of the model

As is proceeded in [43], we would like to express the interface condition in terms of the so-called Steklov-Poincaré operators for which we introduce the interface variable $\lambda(t)$ by

$$d^{\mathrm{s}} = d^{\mathrm{f}} = \lambda$$

for time t at Γ_0. Then the no-slip interface condition is automatically satisfied.

Structure sub-problem

Let $S_{\mathrm{s}}(\lambda)$ denote the Neumann data $\sigma_{\mathrm{s}}(d^{\mathrm{s}})n_{\mathrm{s}}$ of the structure problem with prescribed Dirichlet data λ at the interface Γ_0. This is stated in the following problem.

Problem 3.4. *Calculate*

$$S_{\mathrm{s}}(\lambda) := \sigma_{\mathrm{s}}(d^{\mathrm{s}})n_{\mathrm{s}} \tag{3.2.19}$$

via solving the following structure problem: Find the displacement $d^{\mathrm{s}} := d^{\mathrm{s}}(x_0, t)$ such that

$$\begin{cases} \rho_{\mathrm{s}}\dfrac{\partial^2 d^{\mathrm{s}}}{\partial t^2} - \mathrm{div}(\sigma_{\mathrm{s}}(d^{\mathrm{s}})) = f_{\mathrm{s}} & \text{in } \Omega_0^{\mathrm{s}}, \\ \sigma_{\mathrm{s}}(d^{\mathrm{s}})n_{\mathrm{s}} = 0 & \text{on } \Gamma_0^{\mathrm{n}}, \\ d^{\mathrm{s}} = 0 & \text{on } \Gamma_0^{\mathrm{d}}, \\ d^{\mathrm{s}} = \lambda & \text{on } \Gamma_0. \end{cases} \tag{3.2.20}$$

Fluid sub-problem

Let $S_{\mathrm{f}}(\lambda)$ denote the Neumann data $\sigma_{\mathrm{f}}(u,p)n_{\mathrm{f}} \circ x_t^{\mathrm{f}}$ of the following fluid problem with prescribed Dirichlet data $\frac{\partial \lambda(t)}{\partial t}$ at the interface Γ_0. This is stated in the following problem.

Problem 3.5. *Calculate*

$$S_{\mathrm{f}}(\lambda) := (\sigma_{\mathrm{f}}(u,p)n_{\mathrm{f}}) \circ x_t^{\mathrm{f}} \tag{3.2.21}$$

via solving the following problem.

We first compute the harmonic extension for given λ at the interface Γ_0 for the fluid domain: Find $d^{\mathrm{f}} := d^{\mathrm{f}}(x_0, t)$ such that

$$\begin{cases} -\Delta d^{\mathrm{f}} = 0 & \text{in } \Omega_0^{\mathrm{f}}, \\ d^{\mathrm{f}} = \lambda & \text{on } \Gamma_0, \\ d^{\mathrm{f}} = 0 & \text{on } \Gamma_{\mathrm{in}} \cup \Gamma_{\mathrm{out}}. \end{cases} \tag{3.2.22}$$

The computational fluid domain is then given by

$$\Omega^{\mathrm{f}}(t) = d^{\mathrm{f}} + \Omega_0^{\mathrm{f}}.$$

Then, for the fluid sub-problem, find (u, p) *such that*

$$\begin{cases} \rho_{\mathrm{f}} \dfrac{\partial u}{\partial t}\Big|_{x_0} + \rho_{\mathrm{f}} \Big((u - w^{\mathrm{f}}) \cdot \nabla\Big) u - 2\mu \mathrm{div}\varepsilon(u) + \nabla p = 0 & \text{in } \Omega^{\mathrm{f}}(t), \\ \mathrm{div} u = 0 & \text{in } \Omega^{\mathrm{f}}(t), \\ \sigma_{\mathrm{f}}(u,p) n_{\mathrm{f}} = g_{\mathrm{in}} & \text{on } \Gamma_{\mathrm{in}}(t), \\ \sigma_{\mathrm{f}}(u,p) n_{\mathrm{f}} = 0 & \text{on } \Gamma_{\mathrm{out}}(t), \\ u \circ x_t^{\mathrm{f}} = \dfrac{\partial \lambda}{\partial t} & \text{on } \Gamma_0, \end{cases} \tag{3.2.23}$$

where the ALE time derivative $\frac{\partial u}{\partial t}\big|_{x_0}$ *and the fluid domain velocity* w^{f} *are defined in Proposition 3.2.*

Steklov-Poincaré equation

Then the coupled problem is reduced to the following equation

$$S(\lambda) := S_{\mathrm{s}}(\lambda) + S_{\mathrm{f}}(\lambda) = 0, \tag{3.2.24}$$

which is the so-called Steklov-Poincaré equation. Pay attention this is a time dependent problem which has to be satisfied for all t.

3.3 Weak formulations

For the weak formulation, we need the function spaces

$$\begin{cases} V^{\mathrm{s}} = [H^1(\Omega_0^{\mathrm{s}})]^3, \\ V_0^{\mathrm{s}} = \{v^{\mathrm{s}} \in V^{\mathrm{s}} | v^{\mathrm{s}} = 0 \quad \text{on } \Gamma_0^{\mathrm{d}} \cup \Gamma_0\}, \\ V_g^{\mathrm{s}}(t) = \{v^{\mathrm{s}} \in V^{\mathrm{s}} | v^{\mathrm{s}} = \lambda(t) \quad \text{on } \Gamma_0\} \end{cases} \tag{3.3.1}$$

for the structure. For the fluid, we define

$$\begin{cases} D^{\mathrm{f}} = [H^1(\Omega_0^{\mathrm{f}})]^3, \\ D_0^{\mathrm{f}} = \{d \in D^{\mathrm{f}} | d = 0 \quad \text{on } \Gamma_0\}, \\ D_g^{\mathrm{f}}(t) = \{d \in D^{\mathrm{f}} | d = \lambda(t) \quad \text{on } \Gamma_0\}, \\ V^{\mathrm{f}}(t) = \{v^{\mathrm{f}} | v^{\mathrm{f}} = \hat{v}^{\mathrm{f}} \circ \left(x_t^{\mathrm{f}}\right)^{-1}, \hat{v}^{\mathrm{f}} \in [H^1(\Omega_0^{\mathrm{f}})]^3\}, \\ V_0^{\mathrm{f}}(t) = \{v^{\mathrm{f}} \in V^{\mathrm{f}}(t) | v^{\mathrm{f}} \circ x_t^{\mathrm{f}} = 0 \quad \text{on } \Gamma_0\}, \\ V_g^{\mathrm{f}}(t) = \{v^{\mathrm{f}} \in V^{\mathrm{f}}(t) | v^{\mathrm{f}} \circ x_t^{\mathrm{f}} = \hat{w}^{\mathrm{f}} \quad \text{on } \Gamma_0\}, \\ Q^{\mathrm{f}}(t) = \{q^{\mathrm{f}} | q^{\mathrm{f}} = \hat{q}^{\mathrm{f}} \circ \left(x_t^{\mathrm{f}}\right)^{-1}, \hat{q}^{\mathrm{f}} \in L^2(\Omega_0^{\mathrm{f}})\}, \end{cases} \tag{3.3.2}$$

where $H^1(\Omega_0^{\mathrm{s}})$ and $H^1(\Omega_0^{\mathrm{f}})$ denote the standard Sobolev spaces.

Then we obtain:

3.3.1 The weak form of the structure problem

Structure weak formulation

By standard argument, we obtain the following variational problem for $d^{\mathrm{s}} \in V_g^{\mathrm{s}}(t)$:

$$\int_{\Omega_0^{\mathrm{s}}} \rho_{\mathrm{s}} \frac{\partial^2 d^{\mathrm{s}}}{\partial t^2} \cdot v^{\mathrm{s}} dx_0 + \int_{\Omega_0^{\mathrm{s}}} [\lambda^l \mathrm{div} d^{\mathrm{s}} \mathrm{div} v^{\mathrm{s}} + 2\mu^l \epsilon(d^{\mathrm{s}}) : \epsilon(v^{\mathrm{s}})] dx_0 = 0 \tag{3.3.3}$$

for all $v^{\mathrm{s}} \in V_0^{\mathrm{s}}$.

Time semi-discretized structure weak formulation

We denote the time step size by δt and the time level at n by $t^n = n\delta t$.

For the time discretization of the structure problem, we follow the strategy in [43], where the Newmark method with $\gamma = 2\beta = 1$ was proposed:

$$\int_{\Omega_0^{\mathrm{s}}} \rho_{\mathrm{s}} \frac{\partial^2 d^{\mathrm{s}}}{\partial t^2} \cdot v^{\mathrm{s}} dx_0 \approx \frac{2}{\delta t^2} \int_{\Omega_0^{\mathrm{s}}} \rho_{\mathrm{s}} d^{\mathrm{s},n+1} v^{\mathrm{s}} dx_0 - \frac{2}{\delta t^2} \int_{\Omega_0^{\mathrm{s}}} \rho_{\mathrm{s}} (d^{\mathrm{s},n} + \delta t w^{\mathrm{s},n}) v^{\mathrm{s}} dx_0. \tag{3.3.4}$$

Here $w^{\mathrm{s},n}$ is the structure domain velocity at time t^n. Using the calculated displacement $d^{\mathrm{s},n+1}$ at time t^{n+1}, we can update the structure domain velocity by

$$w^{\mathrm{s},n+1} = \frac{2}{\delta t}(d^{\mathrm{s},n+1} - d^{\mathrm{s},n}) - w^{\mathrm{s},n}. \tag{3.3.5}$$

This leads to the following variational problem, which must be solved in each time step:

Problem 3.6. *Find $d^{\mathrm{s},n+1} = d^{\mathrm{s}}(t^{n+1}) \in V_g^{\mathrm{s}}(t)$ such that for all $v^{\mathrm{s}} \in V_0^{\mathrm{s}}$,*

$$a^{\mathrm{s}}(d^{\mathrm{s},n+1}, v^{\mathrm{s}}) = \langle F^{\mathrm{s},n}, v^{\mathrm{s}} \rangle \tag{3.3.6}$$

with the bilinear and linear forms on V^{s}:

$$\begin{cases} a^{\mathrm{s}}(d^{\mathrm{s},n+1}, v^{\mathrm{s}}) = \dfrac{2}{\delta t^2} \displaystyle\int_{\Omega_0^{\mathrm{s}}} \rho_{\mathrm{s}} d^{\mathrm{s},n+1} v^{\mathrm{s}} dx_0 \\ \qquad\qquad + \displaystyle\int_{\Omega_0^{\mathrm{s}}} [\lambda^l \mathrm{div} d^{\mathrm{s},n+1} \mathrm{div} v^{\mathrm{s}} + 2\mu^l \epsilon(d^{\mathrm{s},n+1}) : \epsilon(v^{\mathrm{s}})] dx_0, \\ \langle F^{\mathrm{s},n}, v^{\mathrm{s}} \rangle = \dfrac{2}{\delta t^2} \displaystyle\int_{\Omega_0^{\mathrm{s}}} \rho_{\mathrm{s}} (d^{\mathrm{s},n} + \delta t w^{\mathrm{s},n}) v^{\mathrm{s}} dx_0. \end{cases} \tag{3.3.7}$$

3.3.2 The weak form of the fluid problem

Fluid weak formulation

Due to the ALE formulation, we need to calculate the variational form with the ALE time derivative

$$\int_{\Omega^{\mathrm{f}}(t)} \rho_{\mathrm{f}} \left. \frac{\partial u}{\partial t} \right|_{x_0} \cdot v^{\mathrm{f}} dx. \tag{3.3.8}$$

From the Reynolds transport theorem and the definition (3.2.15), we obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}u\cdot v^{\mathrm{f}}dx &= \int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}\left(\frac{\partial}{\partial t}\left(u\cdot v^{\mathrm{f}}\right)+\mathrm{div}(u\cdot v^{\mathrm{f}}w^{\mathrm{f}})\right)dx\\ &= \int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}\left(\frac{\partial u}{\partial t}+\left(w^{\mathrm{f}}\cdot\nabla\right)u\right)v^{\mathrm{f}}dx\\ &+ \int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}\left(\frac{\partial v^{\mathrm{f}}}{\partial t}+\left(w^{\mathrm{f}}\cdot\nabla\right)v^{\mathrm{f}}\right)udx\\ &+ \int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}\mathrm{div}w^{\mathrm{f}}u\cdot v^{\mathrm{f}}dx\\ &= \int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}\left.\frac{\partial u}{\partial t}\right|_{x_0}\cdot v^{\mathrm{f}}dx+\int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}\left.\frac{\partial v^{\mathrm{f}}}{\partial t}\right|_{x_0}\cdot udx+\int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}\mathrm{div}w^{\mathrm{f}}u\cdot v^{\mathrm{f}}dx.\end{aligned} \tag{3.3.9}$$

Since we know

$$\left.\frac{\partial v^{\mathrm{f}}}{\partial t}\right|_{x_0}=\frac{\partial}{\partial t}\left(v^{\mathrm{f}}\circ x_t^{\mathrm{f}}\right)\circ\left(x_t^{\mathrm{f}}\right)^{-1}=\frac{\partial}{\partial t}\left(\hat{v}^{\mathrm{f}}\circ\left(x_t^{\mathrm{f}}\right)^{-1}\circ x_t^{\mathrm{f}}\right)\circ\left(x_t^{\mathrm{f}}\right)^{-1}=0, \tag{3.3.10}$$

we obtain

$$\int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}\left.\frac{\partial u}{\partial t}\right|_{x_0}\cdot v^{\mathrm{f}}dx=\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}u\cdot v^{\mathrm{f}}dx-\int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}\mathrm{div}w^{\mathrm{f}}u\cdot v^{\mathrm{f}}dx. \tag{3.3.11}$$

The weak form for the fluid problem is obtained as follows. We firstly need the weak formulation for the harmonic extension of the fluid domain: Find $d^{\mathrm{f}}\in D_g^{\mathrm{f}}(t)$ such that for all $\phi\in D_0^{\mathrm{f}}$,

$$\int_{\Omega_0^{\mathrm{f}}}\nabla d^{\mathrm{f}}:\nabla\phi dx_0=0. \tag{3.3.12}$$

The computational fluid domain $\Omega^{\mathrm{f}}(t)$ is then given by

$$\Omega^{\mathrm{f}}(t)=d^{\mathrm{f}}+\Omega_0^{\mathrm{f}}, \tag{3.3.13}$$

and the ALE weak formulation for the fluid reads: Find $(u(t),p(t))\in V_g^{\mathrm{f}}(t)\times Q^{\mathrm{f}}(t)$ such that for all $(v^{\mathrm{f}},q^{\mathrm{f}})\in V_0^{\mathrm{f}}(t)\times Q^{\mathrm{f}}(t)$,

$$\begin{cases}\dfrac{\mathrm{d}}{\mathrm{d}t}\displaystyle\int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}u\cdot v^{\mathrm{f}}dx+\int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}\left((u-w^{\mathrm{f}})\cdot\nabla\right)u\cdot v^{\mathrm{f}}dx+2\mu\displaystyle\int_{\Omega^{\mathrm{f}}(t)}\varepsilon(u):\varepsilon(v^{\mathrm{f}})dx\\ -\displaystyle\int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}\mathrm{div}w^{\mathrm{f}}u\cdot v^{\mathrm{f}}dx-\int_{\Omega^{\mathrm{f}}(t)}p\ \mathrm{div}v^{\mathrm{f}}dx=\displaystyle\int_{\Gamma_{\mathrm{in}}(t)}g_{\mathrm{in}}\cdot v^{\mathrm{f}}ds,\\ -\displaystyle\int_{\Omega^{\mathrm{f}}(t)}q^{\mathrm{f}}\ \mathrm{div}udx=0.\end{cases} \tag{3.3.14}$$

Time semi-discretized fluid weak formulation

For the fluid problem an implicit Euler scheme is used:

$$\left[\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega^{\mathrm{f}}(t)}\rho_{\mathrm{f}}u\cdot v^{\mathrm{f}}dx\right]_{t^{n+1}}\approx\frac{1}{\delta t}\left(\int_{\Omega^{\mathrm{f}}(t^{n+1})}\rho_{\mathrm{f}}u(t^{n+1})\cdot v^{\mathrm{f},n+1}dx-\int_{\Omega^{\mathrm{f}}(t^{n})}\rho_{\mathrm{f}}u(t^{n})\cdot v^{\mathrm{f},n}dx\right), \tag{3.3.15}$$

where $v^{\mathrm{f}}=\hat{v}^{\mathrm{f}}\circ\left(x_t^{\mathrm{f}}\right)^{-1}$, in particular, $v^{\mathrm{f},k}=\hat{v}^{\mathrm{f}}\circ\left(x_{t^k}^{\mathrm{f}}\right)^{-1}$. The non-linear convective term is treated in a semi-implicit way (see [52]). Then we obtain the semi-implicit fluid weak formulation.

Problem 3.7. *We firstly write down the weak formulation for the harmonic extension of the fluid domain. Find $d^{\mathrm{f},n+1}\in D_g^{\mathrm{f}}(t^{n+1})$ such that for all $\phi\in D_0^{\mathrm{f}}$,*

$$\int_{\Omega_0^{\mathrm{f}}}\nabla d^{\mathrm{f},n+1}:\nabla\phi dx_0=0. \tag{3.3.16}$$

The computational fluid domain $\Omega^{\mathrm{f}}(t^{n+1})$ will be given by

$$\Omega^{\mathrm{f}}(t^{n+1})=d^{\mathrm{f},n+1}+\Omega_0^{\mathrm{f}}. \tag{3.3.17}$$

For the fluid domain velocity we set

$$w^{\mathrm{f},n+1}=\frac{d^{\mathrm{f},n+1}-d^{\mathrm{f},n}}{\delta t}\circ\left(x_{t^{n+1}}^{\mathrm{f}}\right)^{-1}. \tag{3.3.18}$$

Then the fluid sub-problem reads: find $(u^{n+1},p^{n+1})=(u(t^{n+1}),p(t^{n+1}))\in V_g^{\mathrm{f}}(t^{n+1})\times Q^{\mathrm{f}}(t^{n+1})$ such that for all $(v^{\mathrm{f},n+1},q^{\mathrm{f},n+1})\in V_0^{\mathrm{f}}(t^{n+1})\times Q^{\mathrm{f}}(t^{n+1})$,

$$\begin{cases}a^{\mathrm{f}}(u^{n+1},v^{\mathrm{f},n+1})+b(v^{\mathrm{f},n+1},p^{n+1}) &=\langle F^{\mathrm{f},n+1},v^{\mathrm{f},n+1}\rangle,\\ b(u^{n+1},q^{\mathrm{f},n+1}) &=0\end{cases} \tag{3.3.19}$$

with the bilinear and linear forms on $V^{\mathrm{f}}(t)$ and $Q^{\mathrm{f}}(t)$:

$$\begin{cases}a^{\mathrm{f}}(u^{n+1},v^{\mathrm{f},n+1})=&\dfrac{1}{\delta t}\displaystyle\int_{\Omega^{\mathrm{f}}(t^{n+1})}\rho_{\mathrm{f}}u^{n+1}\cdot v^{\mathrm{f},n+1}dx-\int_{\Omega^{\mathrm{f}}(t^{n+1})}\rho_{\mathrm{f}}\left(\mathrm{div}w^{\mathrm{f},n+1}\right)u^{n+1}\cdot v^{\mathrm{f},n+1}dx\\ &+\displaystyle\int_{\Omega^{\mathrm{f}}(t^{n+1})}\rho_{\mathrm{f}}\left(\left(\hat{u}^n-w^{\mathrm{f},n+1}\right)\cdot\nabla\right)u^{n+1}\cdot v^{\mathrm{f},n+1}dx\\ &+2\mu\displaystyle\int_{\Omega^{\mathrm{f}}(t^{n+1})}\varepsilon(u^{n+1}):\varepsilon(v^{\mathrm{f},n+1})dx,\\ b(v^{\mathrm{f},n+1},p^{n+1})=&-\displaystyle\int_{\Omega^{\mathrm{f}}(t^{n+1})}p^{n+1}\ \mathrm{div}v^{\mathrm{f},n+1}dx,\\ \langle F^{\mathrm{f},n+1},v^{\mathrm{f},n+1}\rangle=&\dfrac{1}{\delta t}\displaystyle\int_{\Omega^{\mathrm{f}}(t^{n})}\rho_{\mathrm{f}}u^{n}\cdot v^{\mathrm{f},n}dx+\int_{\Gamma_{\mathrm{in}}(t^{n+1})}g_{\mathrm{in}}\cdot v^{\mathrm{f},n+1}ds,\end{cases} \tag{3.3.20}$$

where $\hat{u}^n=u^n\circ x_{t^n}^{\mathrm{f}}\circ\left(x_{t^{n+1}}^{\mathrm{f}}\right)^{-1}$. This composition is due to the fact that we have to transport the old values u^n to the current domain $\Omega^{\mathrm{f}}(t^{n+1})$ such that the integral of this quantity over the new domain will make sense.

3.3.3 The variational form of the interface equation

In the weak form, the previously introduced Steklov-Poincaré operators become operators from the Sobolev space $[H^{1/2}(\Gamma_0)]^3$ (which is the space of traces on Γ_0 of $[H^1(\Omega)]^3$-functions) to its dual $[H^{-1/2}(\Gamma_0)]^3$:

$$S_{\rm s} : [H^{1/2}(\Gamma_0)]^3 \to [H^{-1/2}(\Gamma_0)]^3, \quad S_{\rm f} : [H^{1/2}(\Gamma_0)]^3 \to [H^{-1/2}(\Gamma_0)]^3. \tag{3.3.21}$$

Then we end up with the following problem:

Problem 3.8. *Find* $\lambda \in [H^{1/2}(\Gamma_0)]^3$ *such that*

$$\langle S_{\rm f}(\lambda), \mu\rangle_{\Gamma_0} + \langle S_{\rm s}(\lambda), \mu\rangle_{\Gamma_0} = 0 \tag{3.3.22}$$

for all $\mu \in [H^{1/2}(\Gamma_0)]^3$.

3.4 Newton's method for the interface equation

3.4.1 Newton's method

The problem (3.3.22) is already discretized in time. When the solution is available at time t^n, we are looking for the solution at the next time level $t^{n+1} = t^n + \delta t$. If no ambiguity appears, all quantities will be referred to $t = t^{n+1}$ and, for simplicity, we will drop the time variables. So the problem now reads

$$S_{\rm s}(\lambda) + S_{\rm f}(\lambda) = 0. \tag{3.4.1}$$

Newton's method applied to the interface equation is given by

$$\lambda^{k+1} = \lambda^k + P_k^{-1}\left(-S_{\rm s}(\lambda^k) - S_{\rm f}(\lambda^k)\right) \tag{3.4.2}$$

with

$$P_k = S_{\rm s}'(\lambda^k) + S_{\rm f}'(\lambda^k), \tag{3.4.3}$$

where the superscript k indicates the Newton iteration steps, see[43].

In each step of the iterative method a problem of the form

$$P_k \delta\lambda^k = -\left(S_{\rm s}(\lambda^k) + S_{\rm f}(\lambda^k)\right) \tag{3.4.4}$$

has to be solved. For this we will use (after discretization in space) a preconditioned GMRES method with preconditioner $S_{\rm s}'(\lambda^k)$ (see [43]).

Summarizing, the method is described in Algorithm 3.4.1.

Note that Step 1 can be parallelized due to the independence of the sub-problems for given interface boundary conditions. Step 2 requires solving linearized structure and fluid sub-problems several times during the GMRES iteration. The algebraic multigrid method (AMG) will be used for solving these structure and fluid sub-problems, and for solving their linearized sub-problems as well, see Chapter 5.

In computation, Algorithm 3.4.1 requires the evaluation of $S_{\rm s}(\lambda)$ and $S_{\rm f}(\lambda)$ (in Step 1), and the evaluation of $S_{\rm s}'(\lambda)\delta\lambda$ and $S_{\rm f}'(\lambda)\delta\lambda$ (in Step 2 for GMRES).

Algorithm 3.4.1 Newton iterations

For $k \geq 0$,

1: update the residual $S_{\mathrm{s}}(\lambda^k) + S_{\mathrm{f}}(\lambda^k)$ by solving the structure and fluid sub-problems,

2: solve the linear problem $\left(S'_{\mathrm{s}}(\lambda^k) + S'_{\mathrm{f}}(\lambda^k)\right)\delta\lambda^k = -\left(S_{\mathrm{s}}(\lambda^k) + S_{\mathrm{f}}(\lambda^k)\right)$ via GMRES method,

3: update the displacement $\lambda^{k+1} = \lambda^k + \delta\lambda^k$, if not accurate enough, go to step 1.

3.4.2 Evaluation of $S(\lambda)$

The normal stresses are computed from the residuals of the weak forms for the structure and fluid sub-problems, respectively.

Proposition 3.9. *For the structure problem, we have*

$$\langle S_{\mathrm{s}}(\lambda), \mu\rangle_{\Gamma_0} = a^{\mathrm{s}}(d^{\mathrm{s},n+1}, v^{\mathrm{s}}) - \langle F^{\mathrm{s},n}, v^{\mathrm{s}}\rangle \tag{3.4.5}$$

for all $v^{\mathrm{s}} \in V^{\mathrm{s}}$ with $\mu = v^{\mathrm{s}}|_{\Gamma_0}$, where $d^{\mathrm{s},n+1} = d^{\mathrm{s},n+1}(\lambda)$ solves the structure problem (3.3.6) for prescribed λ at the interface Γ_0.

In an analogous way, for the fluid problem, we have

$$\langle S_{\mathrm{f}}(\lambda), \mu\rangle_{\Gamma_0} = a^{\mathrm{f}}(u^{n+1}, v^{\mathrm{f}}) + b(v^{\mathrm{f}}, p^{n+1}) - \langle F^{\mathrm{f},n+1}, v^{\mathrm{f}}\rangle \tag{3.4.6}$$

for all $v^{\mathrm{f}} \in V^{\mathrm{f}}(t)$ with $\mu = v^{\mathrm{f}} \circ x_t^{\mathrm{f}}|_{\Gamma_0}$, where $u^{n+1} = u^{n+1}(\lambda)$, $p^{n+1} = p^{n+1}(\lambda)$ solve the fluid problem (3.3.19) for prescribed λ at the interface Γ_0. The computational domain $\Omega^{\mathrm{f}}(t^{n+1})$ is calculated from $d^{\mathrm{f},n+1} = d^{\mathrm{f},n+1}(\lambda)$ given by problem (3.3.16) for prescribed λ at the interface Γ_0.

Proof. Multiply the momentum balance equation for the structure problem by $v^{\mathrm{s}} \in V^{\mathrm{s}}$ and do integration by parts. Multiply the momentum balance equation for the fluid problem by $v^{\mathrm{f}} \in V^{\mathrm{f}}(t)$ and do integration by parts. □

Motivated by the right hand side in (3.4.5) and (3.4.6), we introduced the following operators.

Definition 3.10. *Let $U^{\mathrm{s}} = V^{\mathrm{s}}$ and $U^{\mathrm{f}} = V^{\mathrm{f}}(t) \times Q^{\mathrm{f}}(t) \times D^{\mathrm{f}}$ be spaces of the structure state $d^{\mathrm{s},n+1}$ and fluid states $(u^{n+1}, p^{n+1}, d^{\mathrm{f},n+1})$, respectively. For the structure, we define*

$$\mathcal{S} : U^{\mathrm{s}} \longrightarrow (V^{\mathrm{s}})' \tag{3.4.7}$$

at time level $t = t^{n+1}$ by

$$\langle \mathcal{S}(d^{\mathrm{s},n+1}), v^{\mathrm{s}}\rangle := a^{\mathrm{s}}(d^{\mathrm{s},n+1}, v^{\mathrm{s}}) - \langle F^{\mathrm{s},n}, v^{\mathrm{s}}\rangle. \tag{3.4.8}$$

In an analogous way, for the fluid, we define

$$\mathcal{F} : U^{\mathrm{f}} \longrightarrow (V^{\mathrm{f}})' \tag{3.4.9}$$

at time level $t = t^{n+1}$ by

$$\langle \mathcal{F}(u^{n+1}, p^{n+1}, d^{\mathrm{f},n+1}), v^{\mathrm{f}}\rangle := a^{\mathrm{f}}(u^{n+1}, v^{\mathrm{f}}) + b(v^{\mathrm{f}}, p^{n+1}) - \langle F^{\mathrm{f},n+1}, v^{\mathrm{f}}\rangle. \tag{3.4.10}$$

With these notations for $\mathcal{S}$, $\mathcal{F}$, and by dropping the superscripts for simplicity, Proposition 3.9 reduces to

$$\langle S_{\mathrm{s}}(\lambda), \mu\rangle_{\Gamma_0} = \langle \mathcal{S}(d^{\mathrm{s}}), v^{\mathrm{s}}\rangle \tag{3.4.11}$$

and

$$\langle S_{\mathrm{f}}(\lambda), \mu\rangle_{\Gamma_0} = \langle \mathcal{F}(u, p, d^{\mathrm{f}}), v^{\mathrm{f}}\rangle. \tag{3.4.12}$$

Note that all variables d^{s}, u, p and d^{f} are depending on λ.

3.5 Evaluation of $S^{'}(\lambda)\delta\lambda$

We will evaluate $S_{\mathrm{s}}^{'}(\lambda)\delta\lambda$ and $S_{\mathrm{f}}^{'}(\lambda)\delta\lambda$, respectively.

3.5.1 Evaluation of $S_{\mathbf{s}}^{'}(\lambda)\delta\lambda$

From the identity

$$\langle S_{\mathrm{s}}(\lambda), \mu\rangle_{\Gamma_0} = \langle \mathcal{S}\left(d^{\mathrm{s}}\left(\lambda\right)\right), v^{\mathrm{s}}\rangle, \tag{3.5.1}$$

we get for the directional derivative in direction $\delta\lambda$ by the chain rule

$$\langle S_{\mathrm{s}}^{'}(\lambda)\delta\lambda, \mu\rangle_{\Gamma_0} = \langle D_{d^{\mathrm{s}}}\mathcal{S}\left(d^{\mathrm{s}}\right)\delta d^{\mathrm{s}}, v^{\mathrm{s}}\rangle, \tag{3.5.2}$$

where $D_{d^{\mathrm{s}}}\mathcal{S}$ is the derivative of $\mathcal{S}$ with respect to d^{s}, and δd^{s} is the directional derivative of $d^{\mathrm{s}}(\lambda)$ in direction of $\delta\lambda$.

It is easy to see that

$$\langle D_{d^{\mathrm{s}}}\mathcal{S}(d^{\mathrm{s}})\delta d^{\mathrm{s}}, v^{\mathrm{s}}\rangle = a^{\mathrm{s}}(\delta d^{\mathrm{s}}, v^{\mathrm{s}}). \tag{3.5.3}$$

It remains to describe how to compute δd^{s}:

It can be shown that $\delta d^{\mathrm{s}} \in V_{\delta\lambda}^{\mathrm{s}} := \{v \in V^{\mathrm{s}} | v = \delta\lambda \text{ on } \Gamma_0\}$ solves the Problem 3.6 with modified right hand side

$$a^{\mathrm{s}}(\delta d^{\mathrm{s}}, v^{\mathrm{s}}) = 0 \tag{3.5.4}$$

for all $v^{\mathrm{s}} \in V_0^{\mathrm{s}}$.

This problem corresponds to the following problem in strong form:

$$\begin{cases} \dfrac{2}{\delta t^2}\rho_{\mathrm{s}}\delta d^{\mathrm{s}} - \mathrm{div}(\sigma_{\mathrm{s}}(\delta d^{\mathrm{s}})) = 0 & \text{in } \Omega_0^{\mathrm{s}}, \\ \sigma_{\mathrm{s}}(\delta d^{\mathrm{s}})n_{\mathrm{s}} = 0 & \text{on } \Gamma_0^{\mathrm{n}}, \\ \delta d^{\mathrm{s}} = 0 & \text{on } \Gamma_0^{\mathrm{d}}, \\ \delta d^{\mathrm{s}} = \delta\lambda & \text{on } \Gamma_0. \end{cases} \tag{3.5.5}$$

3.5.2 Evaluation of $S_{\mathbf{f}}^{'}(\lambda)\delta\lambda$

From the identity

$$\langle S_{\mathrm{f}}(\lambda), \mu\rangle_{\Gamma_0} = \langle \mathcal{F}(u(\lambda), p(\lambda), d^{\mathrm{f}}(\lambda)), v^{\mathrm{f}}\rangle, \tag{3.5.6}$$

we get for the directional derivative in direction $\delta\lambda$, by the chain rule

$$\langle S_{\mathrm{f}}^{'}(\lambda)\delta\lambda, \mu\rangle_{\Gamma_0} = \langle D_{(u,p,d^{\mathrm{f}})}\mathcal{F}(u, p, d^{\mathrm{f}})(\delta u, \delta p, \delta d^{\mathrm{f}}), v^{\mathrm{f}}\rangle, \tag{3.5.7}$$

where $D_{(u,p,d^{\mathrm{f}})}\mathcal{F}$ is the derivative of $\mathcal{F}$ with respect to u, p, and d^{f}, and δu, δp and δd^{f} are directional derivatives of $u(\lambda)$, $p(\lambda)$ and $d^{\mathrm{f}}(\lambda)$ in direction $\delta\lambda$.

We can split the directional derivative into two terms:

$$\begin{aligned}\langle D_{(u,p,d^{\mathrm{f}})}\mathcal{F}(u,p,d^{\mathrm{f}})(\delta u,\delta p,\delta d^{\mathrm{f}}),v^{\mathrm{f}}\rangle &= \langle D_{(u,p,0)}\mathcal{F}(u,p,d^{\mathrm{f}})(\delta u,\delta p,0),v^{\mathrm{f}}\rangle \\ &\quad + \langle D_{(0,0,d^{\mathrm{f}})}\mathcal{F}(u,p,d^{\mathrm{f}})(0,0,\delta d^{\mathrm{f}}),v^{\mathrm{f}}\rangle,\end{aligned} \tag{3.5.8}$$

where $D_{(u,p,0)}\mathcal{F}$ is the derivative of $\mathcal{F}$ with respect to u and p, $D_{(0,0,d^{\mathrm{f}})}\mathcal{F}$ is the derivative of $\mathcal{F}$ with respective to d^{f}.

For the first term from the right hand side of (3.5.8), it is easy to see that

$$\langle D_{(u,p,0)}\mathcal{F}(u,p,d^{\mathrm{f}})(\delta u,\delta p,0),v^{\mathrm{f}}\rangle = a^{\mathrm{f}}(\delta u,v^{\mathrm{f}}) + b(v^{\mathrm{f}},\delta p). \tag{3.5.9}$$

Following the lines of arguments in [52, 63, 64, 144, 105] (see Appendix 3.6 for details), we obtain

$$\begin{aligned}\langle D_{(0,0,d^{\mathrm{f}})}\mathcal{F}(u,p,d^{\mathrm{f}})(0,0,\delta d^{\mathrm{f}}),v^{\mathrm{f}}\rangle = &-\int_\Omega \rho_{\mathrm{f}}\left(\mathrm{div}\left(\left(I\mathrm{div}\delta\hat{d}^{\mathrm{f}}-\nabla\delta\hat{d}^{\mathrm{f}}\right)w\right)\right)u\cdot v^{\mathrm{f}}dx \\ &-\int_\Omega p\left(I\mathrm{div}\delta\hat{d}^{\mathrm{f}}-\left(\nabla\delta\hat{d}^{\mathrm{f}}\right)^T\right):\nabla v^{\mathrm{f}}dx \\ &-\frac{1}{\delta t}\int_\Omega \rho_{\mathrm{f}}\left(\delta\hat{d}^{\mathrm{f}}\cdot\nabla\right)u\cdot v^{\mathrm{f}}dx \\ &+\int_\Omega \rho_{\mathrm{f}}\left(\hat{u}^n-w\right)\cdot\nabla u\left(I\mathrm{div}\delta\hat{d}^{\mathrm{f}}-\nabla\delta\hat{d}^{\mathrm{f}}\right)\cdot v^{\mathrm{f}}dx \\ &-\int_\Omega \mu\left(\nabla u\nabla\delta\hat{d}^{\mathrm{f}}+\left(\nabla u\nabla\delta\hat{d}^{\mathrm{f}}\right)^T\right):\nabla v^{\mathrm{f}}dx \\ &+2\mu\int_\Omega \varepsilon(u)\left(I\mathrm{div}\delta\hat{d}^{\mathrm{f}}-\left(\nabla\delta\hat{d}^{\mathrm{f}}\right)^T\right):\nabla v^{\mathrm{f}}dx\end{aligned} \tag{3.5.10}$$

with

$$\delta\hat{d}^{\mathrm{f}} = \delta d^{\mathrm{f}}\circ\left(id+d^{\mathrm{f}}\right)^{-1}. \tag{3.5.11}$$

It remains to describe how to compute δd^{f}, δu and δp:

The evaluation of δd^{f} is given by the harmonic extension (3.3.12):

$$\delta d^{\mathrm{f}} = \mathrm{Ext}(\delta\lambda). \tag{3.5.12}$$

If we differentiate the mixed problem (3.3.19) with respect to λ, it can be shown that $\delta u \in V^{\mathrm{f}}_{\delta\hat{d}}(t) := \{v\in V^{\mathrm{f}}(t) : \delta u|_\Gamma = \frac{\delta\hat{d}^{\mathrm{f}}}{\delta t}\}$ and $\delta p \in Q^{\mathrm{f}}(t)$ solve the modified fluid mixed Problem 3.7:

$$\begin{cases} a^{\mathrm{f}}(\delta u,v^{\mathrm{f}})+b(v^{\mathrm{f}},\delta p) &= \langle F^{\mathrm{f}},v^{\mathrm{f}}\rangle, \\ b(\delta u,q) &= \langle G^{\mathrm{f}},q\rangle \end{cases} \tag{3.5.13}$$

with

$$\langle F^{\mathrm{f}},v^{\mathrm{f}}\rangle = -\langle D_{(0,0,d^{\mathrm{f}})}\mathcal{F}(u,p,d^{\mathrm{f}})(0,0,\delta d^{\mathrm{f}}),v^{\mathrm{f}}\rangle \tag{3.5.14}$$

and

$$\langle G^{\mathrm{f}},q\rangle = -\int_\Omega\left(\mathrm{div}\left(\left(I\mathrm{div}\delta\hat{d}^{\mathrm{f}}-\nabla\delta\hat{d}^{\mathrm{f}}\right)u\right)\right)qdx \tag{3.5.15}$$

on the current domain Ω at the time step t^{n+1}. The Dirichlet data at the interface Γ is given by

$$\delta u|_{\Gamma} = \frac{\delta \hat{d}^{\mathrm{f}}}{\delta t}. \tag{3.5.16}$$

This corresponds to the following problem in strong form:

$$\begin{cases} \rho_{\mathrm{f}} \dfrac{\delta u}{\delta t} - 2\mu\varepsilon(\delta u) - \rho_{\mathrm{f}} \mathrm{div} w \delta u + \rho_{\mathrm{f}} \left((\hat{u}^n - w) \cdot \nabla \right) \delta u + \nabla \delta p = \\ \rho_{\mathrm{f}} \left(\mathrm{div} \left(\left(I \mathrm{div} \delta \hat{d}^{\mathrm{f}} - \nabla \delta \hat{d}^{\mathrm{f}} \right) w \right) \right) u - \mathrm{div} \left(p \left(I \mathrm{div} \delta \hat{d}^{\mathrm{f}} - \left(\nabla \delta \hat{d}^{\mathrm{f}} \right)^T \right) \right) \\ + \dfrac{1}{\delta t} \rho_{\mathrm{f}} \left(\delta \hat{d}^{\mathrm{f}} \cdot \nabla \right) u - \rho_{\mathrm{f}} \left(\hat{u}^n - w \right) \cdot \nabla u \left(I \mathrm{div} \delta \hat{d}^{\mathrm{f}} - \nabla \delta \hat{d}^{\mathrm{f}} \right) \\ - \mu \mathrm{div} \left(\nabla u \nabla \delta \hat{d}^{\mathrm{f}} + \left(\nabla u \nabla \delta \hat{d}^{\mathrm{f}} \right)^T \right) + 2\mu \mathrm{div} \left(\varepsilon(u) \left(I \mathrm{div} \delta \hat{d}^{\mathrm{f}} - \left(\nabla \delta \hat{d}^{\mathrm{f}} \right)^T \right) \right), & \text{in } \Omega, \\ \mathrm{div} \delta u = -\mathrm{div} \left(\left(I \mathrm{div} \delta \hat{d}^{\mathrm{f}} - \nabla \delta \hat{d}^{\mathrm{f}} \right) u \right), & \text{in } \Omega, \\ \sigma_{\mathrm{f}}(\delta u, \delta p) n = p \left(I \mathrm{div} \delta \hat{d}^{\mathrm{f}} - \left(\nabla \delta \hat{d}^{\mathrm{f}} \right)^T \right) n \\ + \mu \left(\nabla u \nabla \delta \hat{d}^{\mathrm{f}} + \left(\nabla u \nabla \delta \hat{d}^{\mathrm{f}} \right)^T \right) n - 2\mu\varepsilon(u) \left(I \mathrm{div} \delta \hat{d}^{\mathrm{f}} - \left(\nabla \delta \hat{d}^{\mathrm{f}} \right)^T \right) n, & \text{on } \Gamma_{\mathrm{in}} \cup \Gamma_{\mathrm{out}}, \\ \delta u = \dfrac{\delta \hat{d}^{\mathrm{f}}}{\delta t}, & \text{on } \Gamma. \end{cases} \tag{3.5.17}$$

3.6 Appendix on treatment of the convection term

Most of the terms appearing in the directional derivatives (3.5.10) have been derived in [52]. The only difference is that there the convection part is treated in conservative form,

$$\int_{\Omega^{\mathrm{f}}} \rho_{\mathrm{f}} \mathrm{div} \left(u \otimes (\hat{u}^n - w) \right) \cdot v^{\mathrm{f}} dx, \tag{3.6.1}$$

while we treat it in convective form, which leads to the replacement of (3.6.1) by

$$\int_{\Omega^{\mathrm{f}}} \rho_{\mathrm{f}} \left((\hat{u}^n - w) \cdot \nabla \right) u \cdot v dx - \int_{\Omega^{\mathrm{f}}} \rho_{\mathrm{f}} \left(\mathrm{div} w \right) u \cdot v dx. \tag{3.6.2}$$

This leads to different directional derivatives concerning this part. In this appendix, we will detail how to obtain the directional derivatives with respect to the variable λ for the weak form (3.6.2).

In order to derive the directional derivative, we need to introduce a perturbed domain Ω_α of the current domain Ω from given $\lambda + \alpha\delta\lambda$ at the interface Γ_0 as shown in Figure 3.5. The perturbed domain Ω_α is given by a transformation $T_\alpha : \Omega \to \Omega_\alpha$,

$$T_\alpha = (id + d^{\mathrm{f}} + \alpha \delta d^{\mathrm{f}}) \circ (id + d^{\mathrm{f}})^{-1}. \tag{3.6.3}$$

The "velocity field" $V : \Omega \times R \to R^3$ is introduced by

$$V(x, \alpha) = \frac{\partial T_\alpha(x)}{\partial \alpha} \circ (T_\alpha(x))^{-1} \tag{3.6.4}$$

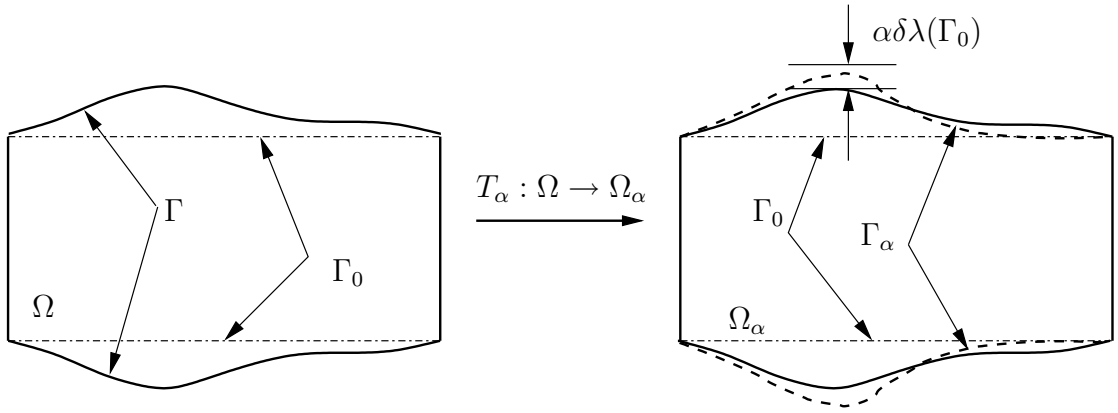


Figure 3.5: The perturbed domain Ω_α ($--$ line) of current domain Ω is due to the perturbation of the displacement $\alpha\delta\lambda$ at the interface Γ_0 ($- \cdot -$ line).

for $x \in \Omega$, see [144].

Using (3.6.3) in (3.6.4), we obtain

$$V(x,\alpha) = \delta d^{\mathrm{f}} \circ \left(id + d^{\mathrm{f}} + \alpha\delta d^{\mathrm{f}} \right)^{-1} \tag{3.6.5}$$

for $x \in \Omega$. With (3.5.11), we obtain

$$V(x,0) = \delta\hat{d}^{\mathrm{f}}. \tag{3.6.6}$$

The following properties are well known:

Proposition 3.11. *Let $F_\alpha := \nabla T_\alpha(x)$ be the Jacobian matrix of T_α at $x \in \Omega$ and $J_\alpha(x) := \det F_\alpha$ be the Jacobian determinant, then*

$$\left[\frac{\mathrm{d}J_\alpha}{\mathrm{d}\alpha}\right]_{\alpha=0} = [\mathrm{div}V]_{\alpha=0} = \mathrm{div}\delta\hat{d}^{\mathrm{f}}, \tag{3.6.7}$$

$$\left[\frac{\mathrm{d}F_\alpha^{-1}}{\mathrm{d}\alpha}\right]_{\alpha=0} = [-\nabla V]_{\alpha=0} = -\nabla\delta\hat{d}^{\mathrm{f}}, \tag{3.6.8}$$

$$\left[\frac{\mathrm{d}F_\alpha^{-T}}{\mathrm{d}\alpha}\right]_{\alpha=0} = \left[-\left(\nabla V\right)^T\right]_{\alpha=0} = -\left(\nabla\delta\hat{d}^{\mathrm{f}}\right)^T. \tag{3.6.9}$$

We have the following identities due to the change of variables.

$$\int_{\Omega_\alpha} \left(\left(\hat{u}_\alpha^n - w_\alpha\right)\cdot\nabla\right) u_\alpha \cdot v_\alpha dx = \int_\Omega \left(\hat{u}^n - w\right)\cdot\nabla u F_\alpha^{-1}\cdot v J_\alpha dx_0, \tag{3.6.10}$$

$$\int_{\Omega_\alpha} \left(\mathrm{div}w_\alpha\right) u_\alpha \cdot v_\alpha dx = \int_\Omega \left(\mathrm{div}\left(J_\alpha F_\alpha^{-1} w\right)\right) u\cdot v dx_0, \tag{3.6.11}$$

with

$$\begin{aligned} u_\alpha &= u\circ T_\alpha^{-1}, \qquad & v_\alpha &= v\circ T_\alpha^{-1},\\ w_\alpha &= w\circ T_\alpha^{-1}, \qquad & \hat{u}_\alpha^n &= \hat{u}^n\circ T_\alpha^{-1}. \end{aligned} \tag{3.6.12}$$

Now, we are able to introduce the directional derivative with respect to the change of geometry:

$$D_{(0,0,d^{\mathrm{f}})}\mathcal{F}(u,p,d^{\mathrm{f}})(0,0,\delta d^{\mathrm{f}}) = \left[\frac{\mathrm{d}\mathcal{F}(u_\alpha, p_\alpha, d^{\mathrm{f}}+\alpha\delta d^{\mathrm{f}})}{\mathrm{d}\alpha}\right]_{\alpha=0}, \tag{3.6.13}$$

where $p_\alpha = p\circ T_\alpha^{-1}$.

As we specified at the beginning, we will only concentrate on the derivative for (3.6.2). For convenience of reading, we will present the result for the two terms in (3.6.2). The derivatives for all the other terms can be found in [52].

Proposition 3.12. *The directional derivatives with respect to the change of geometry of the two terms in (3.6.2) are given in the following:*

$$\begin{aligned}\left[\frac{\mathrm{d}}{\mathrm{d}\alpha}\int_{\Omega_\alpha}\rho_{\mathrm{f}}\left(\left(\hat{u}_\alpha^n - w_\alpha\right)\cdot\nabla\right)u_\alpha\cdot v_\alpha dx\right]_{\alpha=0} &= -\frac{1}{\delta t}\int_\Omega \rho_{\mathrm{f}}\left(\delta\hat{d}^{\mathrm{f}}\cdot\nabla\right)u\cdot v dx_0\\ &\quad+\int_\Omega\rho_{\mathrm{f}}\left(\hat{u}^n - w\right)\cdot\nabla u\left(I\mathrm{div}\delta\hat{d}^{\mathrm{f}} - \nabla\delta\hat{d}^{\mathrm{f}}\right)\cdot v dx_0,\end{aligned} \tag{3.6.14}$$

$$\begin{aligned}\left[\frac{\mathrm{d}}{\mathrm{d}\alpha}\int_{\Omega_\alpha}\rho_{\mathrm{f}}\left(\mathrm{div}w_\alpha\right)u_\alpha\cdot v_\alpha dx\right]_{\alpha=0} &= \frac{1}{\delta t}\int_\Omega\rho_{\mathrm{f}}\left(\mathrm{div}\delta\hat{d}^{\mathrm{f}}\right)u\cdot v dx_0\\ &\quad+\int_\Omega\rho_{\mathrm{f}}\left(\mathrm{div}\left(\left(I\mathrm{div}\delta\hat{d}^{\mathrm{f}} - \nabla\delta\hat{d}^{\mathrm{f}}\right)w\right)\right)u\cdot v dx_0.\end{aligned} \tag{3.6.15}$$

Proof. For the proof of (3.6.14), using transformation (3.6.10), we have

$$\begin{aligned}\left[\frac{\mathrm{d}}{\mathrm{d}\alpha}\int_\Omega\rho_{\mathrm{f}}\left(\hat{u}^n - w\right)\cdot\nabla u F_\alpha^{-1}\cdot v J_\alpha dx_0\right]_{\alpha=0} &= \left[\int_\Omega\rho_{\mathrm{f}}\frac{\mathrm{d}\left(\hat{u}^n - w\right)}{\mathrm{d}\alpha}\cdot\nabla u F_\alpha^{-1}\cdot v J_\alpha dx_0\right]_{\alpha=0}\\ &\quad+\left[\int_\Omega\rho_{\mathrm{f}}\left(\hat{u}^n - w\right)\cdot\nabla\frac{\mathrm{d}u}{\mathrm{d}\alpha}F_\alpha^{-1}\cdot v J_\alpha dx_0\right]_{\alpha=0}\\ &\quad+\left[\int_\Omega\rho_{\mathrm{f}}\left(\hat{u}^n - w\right)\cdot\nabla u\frac{\mathrm{d}F_\alpha^{-1}}{\mathrm{d}\alpha}\cdot v J_\alpha dx_0\right]_{\alpha=0}\\ &\quad+\left[\int_\Omega\rho_{\mathrm{f}}\left(\hat{u}^n - w\right)\cdot\nabla u F_\alpha^{-1}\cdot\frac{\mathrm{d}v}{\mathrm{d}\alpha}J_\alpha dx_0\right]_{\alpha=0}\\ &\quad+\left[\int_\Omega\rho_{\mathrm{f}}\left(\hat{u}^n - w\right)\cdot\nabla u F_\alpha^{-1}\cdot v\frac{\mathrm{d}J_\alpha}{\mathrm{d}\alpha}dx_0\right]_{\alpha=0}.\end{aligned} \tag{3.6.16}$$

From definition of u_α, we also know that

$$\frac{\mathrm{d}u}{\mathrm{d}\alpha} = \lim_{\alpha=0}\frac{u_\alpha\circ T_\alpha - u}{\alpha} = 0, \tag{3.6.17}$$

and the similar result for v, $\frac{\mathrm{d}v}{\mathrm{d}\alpha} = 0$. Since

$$\begin{aligned}\left[\frac{\mathrm{d}\left(\hat{u}^n - w\right)}{\mathrm{d}\alpha}\right]_{\alpha=0} &= \left[\frac{\mathrm{d}\left(-w\right)}{\mathrm{d}\alpha}\right]_{\alpha=0}\\ &= -\left[\frac{\left(d^{\mathrm{f}}+\alpha\delta d^{\mathrm{f}} - d^{\mathrm{f},n}\right) - \left(d^{\mathrm{f}} - d^{\mathrm{f},n}\right)}{\delta t\alpha}\right]_{\alpha=0}\circ\left(id + d^{\mathrm{f}}\right)^{-1}\\ &= -\frac{\delta\hat{d}^{\mathrm{f}}}{\delta t},\end{aligned} \tag{3.6.18}$$

by using (3.6.7) and (3.6.9), we obtain the result (3.6.14).

For the proof of (3.6.15), using transformation (3.6.11), we have

$$\begin{aligned}\left[\frac{\mathrm{d}}{\mathrm{d}\alpha}\int_\Omega \rho_{\mathrm{f}}\left(\operatorname{div}\left(J_\alpha F_\alpha^{-1} w\right)\right) u\cdot v dx_0\right]_{\alpha=0} &= \left[\int_\Omega \rho_{\mathrm{f}}\left(\operatorname{div}\left(\frac{\mathrm{d}J_\alpha}{\mathrm{d}\alpha}F_\alpha^{-1}w\right)\right) u\cdot v dx_0\right]_{\alpha=0}\\ &+ \left[\int_\Omega \rho_{\mathrm{f}}\left(\operatorname{div}\left(J_\alpha \frac{\mathrm{d}F_\alpha^{-1}}{\mathrm{d}\alpha}w\right)\right) u\cdot v dx_0\right]_{\alpha=0}\\ &+ \left[\int_\Omega \rho_{\mathrm{f}}\left(\operatorname{div}\left(J_\alpha F_\alpha^{-1}\frac{\mathrm{d}w}{\mathrm{d}\alpha}\right)\right) u\cdot v dx_0\right]_{\alpha=0}\\ &+ \left[\int_\Omega \rho_{\mathrm{f}}\left(\operatorname{div}\left(J_\alpha F_\alpha^{-1}w\right)\right)\frac{\mathrm{d}u}{\mathrm{d}\alpha}\cdot v dx_0\right]_{\alpha=0}\\ &+ \left[\int_\Omega \rho_{\mathrm{f}}\left(\operatorname{div}\left(J_\alpha F_\alpha^{-1}w\right)\right) u\cdot \frac{\mathrm{d}v}{\mathrm{d}\alpha} dx_0\right]_{\alpha=0}.\end{aligned} \tag{3.6.19}$$

From $\frac{\mathrm{d}u}{\mathrm{d}\alpha}=0$ and $\frac{\mathrm{d}v}{\mathrm{d}\alpha}=0$, we have

$$\left[\frac{\mathrm{d}w}{\mathrm{d}\alpha}\right]_{\alpha=0} = \left[\frac{\left(d^{\mathrm{f}}+\alpha\delta d^{\mathrm{f}}-d^{\mathrm{f},n}\right)-\left(d^{\mathrm{f}}-d^{\mathrm{f},n}\right)}{\delta t\alpha}\right]_{\alpha=0}\circ\left(id+d^{\mathrm{f}}\right)^{-1} = \frac{\delta\hat{d}^{\mathrm{f}}}{\delta t}, \tag{3.6.20}$$

and using (3.6.7) and (3.6.8), we obtain the result (3.6.15). □

If we collect this result and the derivatives for all the other terms from [52], we obtain (3.5.10), which is the second part of the directional derivatives for the fluid problem.

3.7 Summary

In this chapter we formulated Newton's method for solving the Steklov-Poincaré equation based on the previous work [43, 52]. This method requires solving the sub-problems (3.3.6), (3.3.16) and (3.3.19) for the variational residuals and solving the sub-problems (3.5.4), (3.5.12) and (3.5.13) for the directional derivatives.

In order to solve the interface equation (3.4.1) in finite dimensions, in the next two chapters which are the two main contributions of this thesis, we will focus on how to discretize the sub-problems in space, namely, by the finite element methods in Chapter 4, and how to solve the discretized finite element equations using algebraic multigrid methods (AMG) in Chapter 5. Since we are using hybrid meshes containing different element types, the stability and error estimates of the discretization based on such meshes will be discussed in detail. Some ingredients of the AMG solver have to be adjusted such that the algebraic equations arising from the stabilized discretization of the sub-problems will be solved efficiently.

Chapter 4

Finite elements for the sub-problems on hybrid meshes

4.1 Introduction

So far, we have not touched the issue concerning spatial discretization for the structure and fluid sub-problems. The discretization of both sub-problems will be done on hybrid meshes containing several different elements types: tetrahedron, hexahedron, prism and pyramid. The solvability and uniqueness of discrete solutions for each sub-problem have to be guaranteed.

In the next, we will firstly extend the standard P_1 element on pure tetrahedral meshes to hybrid meshes. In order to extend this element to hybrid meshes, we split the elements (hexahedron, prism and pyramid) into pure tetrahedra by introducing artificial points in the volume and quadrilateral face centers of the elements, i.e. we introduce additional nodal degrees of freedom in these elements. These artificially introduced degrees will be eliminated later on by the so-called *mean value approximation*. From this we obtain modified basis functions which are weighted linear combinations of the standard hat basis functions. Thus the finite element space constructed from these basis functions is a subset of the space spanned by standard hat functions. Error estimates can be shown by properly chosen interpolation operators.

This extended P_1 element can be directly applied to the structure sub-problem and the harmonic extension problem for the ALE mapping of the fluid sub-problem.

However, for the fluid problem, we are not allowed to directly apply this extended P_1 elements for the velocity and pressure spaces due to two reasons of instability: the violation of the discrete inf-sup condition and the dominating convection, see [35, 36, 133, 82, 23, 103, 152].

The inf-sup instability can be overcome by enriching the velocity space with sufficiently many bubble functions. For analyzing this method, we explicitly construct a Fortin operator (see [35]) on the hybrid mesh. The additional degrees of freedom are eliminated by static condensation. So the discretized linear system will not be enlarged. For simplicity static condensation is presented only for the Stokes problem.

For the implementation, we use another approach which combines the pressure stabilization Petrov-Galerkin method (PSPG) and the streamline upwind Petrov-Galerkin method (SUPG). These two methods overcome the instability caused by those two effects, respec-

tively. The analysis and their applications to the Oseen and Navier-Stokes problems can be found in [36, 133, 82, 23, 103, 152].

4.2 An extension of the P_1 element on hybrid meshes

In this section, we will extend the P_1 element on pure tetrahedral meshes to hybrid meshes.

4.2.1 The P_1 element on auxiliary tetrahedral meshes

Let $\mathcal{M}_h$ be the original subdivision of the domain $\Omega \subset \mathbb{R}^3$ into tetrahedra, hexahedra, prisms and pyramids, which is assumed to be admissible, i.e. any two elements from $\mathcal{M}_h$ either have no intersection, or have a common face, or have a common edge, or have a common vertex.

Let $\mathcal{T}_h$ be the admissible subdivision into tetrahedra, obtained in the following way: we add points at the centers of quadrilateral faces and subdivide each of them into four triangles, then we add a point at the center of the element, and finally we connect this center point with all the original vertices and the face center points. By this splitting, we obtain 14 tetrahedra for the prism, 24 for the hexahedron, and 8 for the pyramid. See Figure 4.1 for an illustration.

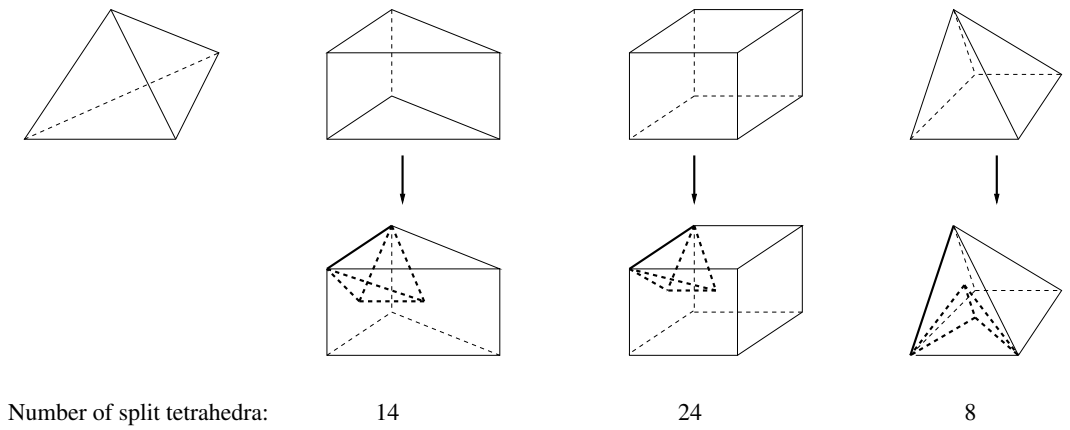


Figure 4.1: Split hybrid elements into tetrahedral elements.

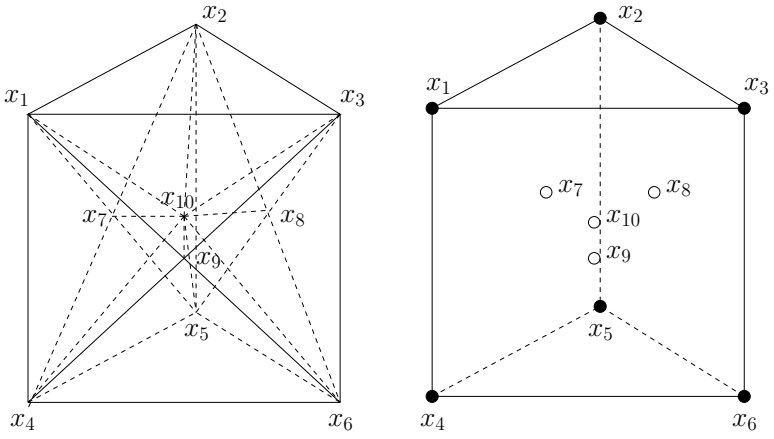
Next we introduce the P_1 element on $\mathcal{T}_h$, by

$$V_h^T = \{v \in C(\overline{\Omega}) : v|_T \in P_1, \forall T \in \mathcal{T}_h\}.$$

4.2.2 Mean value approximation and modified basis functions

Until now, we only considered the finite element spaces on the auxiliary tetrahedral mesh containing the additional nodal degrees of freedom. We are going to eliminate those artificial degrees of freedom by (arithmetic) mean value approximation.

We describe our construction on a prismatic element, say M only. The results extend analogously to other element types. A prism with original vertices x_i, $i \in \{1, ..., 6\}$ can be split into 14 tetrahedra by introducing three face centers x_i, $i = \{7, 8, 9\}$ and a volume center x_{10}, see Figure 4.2.



(a) Splitting of prism (b) Degrees of freedom

Figure 4.2: A prism split into 14 tetraheda by connecting the original vertices x_i, $i = \{1, ..., 6\}$, introduced face centers x_i, $i = \{7, 8, 9\}$ and volume center x_{10}. The filled circles indicate the original degrees of freedom, and the empty circles indicate the added degrees of freedom.

Let $\tilde{\varphi}_i$ be a standard hat function on the tetrahedral subdivision of M:

$$\tilde{\varphi}_i(x_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}. \tag{4.2.1}$$

On $M \in \mathcal{M}_h$, we could represent the discrete finite element solution w_h from V_h^T by a linear combination of these hat functions

$$w_h|_M = \sum_{i=1}^{10} w_i \tilde{\varphi}_i, \tag{4.2.2}$$

where the coefficients $w_i \in \mathbb{R}$ satisfy

$$w_i = w_h(x_i). \tag{4.2.3}$$

The values at the newly introduced artificial vertices x_i, $i \in \{7, ...10\}$, are replaced by taking the mean values at their directly connected original vertices, i.e.,

$$\begin{cases} w(x_7) = w(\dfrac{x_1 + x_2 + x_4 + x_5}{4}) \approx \dfrac{1}{4}\left(w(x_1) + w(x_2) + w(x_4) + w(x_5)\right), \\ w(x_8) = w(\dfrac{x_2 + x_3 + x_5 + x_6}{4}) \approx \dfrac{1}{4}\left(w(x_2) + w(x_3) + w(x_5) + w(x_6)\right), \\ w(x_9) = w(\dfrac{x_1 + x_3 + x_4 + x_6}{4}) \approx \dfrac{1}{4}\left(w(x_1) + w(x_3) + w(x_4) + w(x_6)\right), \\ w(x_{10}) = w(\dfrac{1}{6}\displaystyle\sum_{i=1}^{6} x_i) \approx \dfrac{1}{6}\sum_{i=1}^{6} w(x_i). \end{cases} \tag{4.2.4}$$

This leads to the following approximate representation:

$$w_h|_M = \sum_{i=1}^{6} w_i \varphi_i \tag{4.2.5}$$

for w_h, where φ_i, $i \in 1, ..., 6$, can be represented in the form

$$\varphi_i = \tilde{\varphi}_i + \sum_{j \in N(i)} \frac{1}{4}\tilde{\varphi}_j + \frac{1}{6}\tilde{\varphi}_{10}, \tag{4.2.6}$$

where $N(i)$ is the set of indices $j \in \{7, 8, 9\}$ such that x_j and x_i are on the same face. One can check that the value of the basis function φ_i at particular vertices are given by:

$$\varphi_i(x_j) = \begin{cases} \delta_{i,j} & \text{if } j \in \{1, ..., 6\}, \\ 1/4 & \text{if } j \in \{7, 8, 9\}, \\ 1/6 & \text{if } j = 10. \end{cases}$$

See Figure 4.3 for an illustration of these modified basis functions on a quadrilateral element in 2D, where for each $\varphi_i, i \in \{1, ..., 4\}$, we have $\varphi_i(x_i) = 1$, $\varphi_i(x_j) = 0$ for $j \in \{1, ..., 4\}$ and $j \neq i$, and $\varphi_i(x_5) = 1/4$.

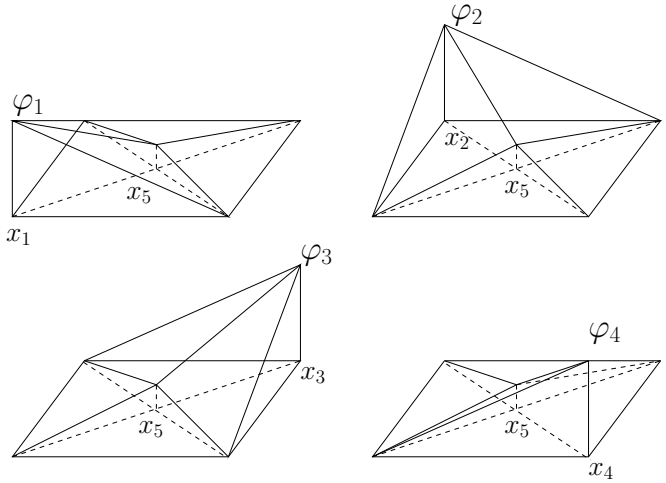


Figure 4.3: Illustration of the modified basis function φ_i on a hybrid element M.

A general formula for the modified basis function on an arbitrary polyhedron can be derived as follows: Suppose a polyhedron has n_V vertices, n_F polygonal faces (excluding triangle faces), and n_{F_k} vertices on the polygonal face F_k. Let x_j, $1 \leq j \leq n_V$ denote the original vertices, x_j, $n_V + 1 \leq j \leq n_V + n_F$ the centers of polygonal faces, and $x_{n_V+n_F+1}$ the center of the element.

Then the modified basis function is given by

$$\varphi_i = \sum_{j=1}^{n_V+n_F+1} \omega_j \tilde{\varphi}_j \tag{4.2.7}$$

with $i \in \{1, ..., n_V\}$ and the weight

$$\omega_j = \begin{cases} 1, & \text{if } j \in \{1, ..., n_V\} \text{ and } j = i, \\ 0, & \text{if } j \in \{1, ..., n_V\} \text{ and } j \neq i, \\ \frac{1}{n_{F_j}}, & \text{if } j \in \{n_V + 1, ..., n_V + n_F\} \text{ and } x_i \text{ and } x_j \text{ are on the same face}, \\ 0, & \text{if } j \in \{n_V + 1, ..., n_V + n_F\} \text{ and } x_i \text{ and } x_j \text{ are on different faces}, \\ \frac{1}{n_V}, & \text{if } j = n_V + n_F + 1. \end{cases} \tag{4.2.8}$$

Note that the basis function φ_i satisfies the following property:

$$\sum_{i=1}^{n_V} \varphi_i = 1, \tag{4.2.9}$$

because

$$\begin{aligned} \sum_{i=1}^{n_V} \varphi_i &= \sum_{i=1}^{n_V} \sum_{j=1}^{n_V+n_F+1} \omega_j \hat{\varphi}_j = \sum_{j=1}^{n_V} \hat{\varphi}_j + \sum_{j=n_V+1}^{n_V+n_F} n_{F_j} \frac{1}{n_{F_j}} \hat{\varphi}_j + \hat{\varphi}_{n_V+n_F+1} \\ &= \sum_{j=1}^{n_V+n_F+1} \hat{\varphi}_j = 1. \end{aligned} \tag{4.2.10}$$

4.2.3 An extended P_1 element on hybrid meshes

The extension of the P_1 element on the subdivision $\mathcal{M}_h$ of the domain $\Omega \subset \mathbb{R}^3$ is now introduced by

$$V_h^H = \{v \in C(\bar{\Omega}) : v|_M \in \Phi_M, \forall M \in \mathcal{M}_h\}, \tag{4.2.11}$$

where

$$\Phi_M = \text{span}\{\varphi_i : M \in \mathcal{M}_h\}. \tag{4.2.12}$$

It can be shown that the global degrees of freedom of V_h^H is the number of original vertices of the hybrid mesh, and the global basis functions can be constructed from the local basis functions.

4.2.4 Interpolation operators

We need to adapt some standard interpolation operators on pure tetrahedral meshes to hybrid meshes.

Regularity assumptions on hybrid meshes

We firstly introduce some useful notations.

x_i, $i \in I = \{1, ..., n_V\}$: the vertices of an element $M \in \mathcal{M}_h$,
$\mathcal{P}_i$: the patch of all elements from $\mathcal{M}_h$ containing x_i as a vertex,
$\mathcal{P} = \bigcup \mathcal{P}_i$: the union of all the patches $\mathcal{P}_i$ covering an element $M \in \mathcal{M}_h$ (see Figure 4.4),

h_T: the diameter of the element $T \in \mathcal{T}_h$,
h_M: the diameter of the element $M \in \mathcal{M}_h$,
$h = \sup_{T\in\mathcal{T}_h}\{h_T\}$: the supremum of h_T over all $T \in \mathcal{T}_h$,
ρ_T: the diameter of the inscribed ball in $T \in \mathcal{T}_h$.

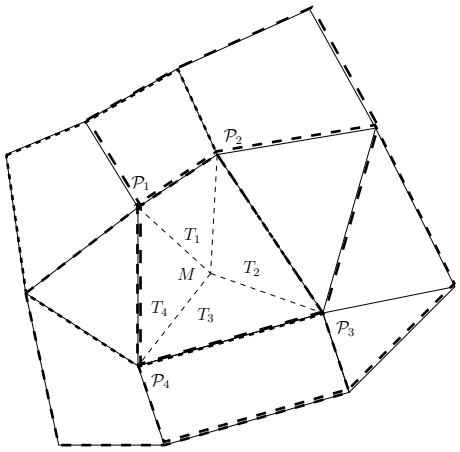


Figure 4.4: The patch $\mathcal{P} = \bigcup \mathcal{P}_i$, $i \in \{1, ..., 4\}$ covers the element M. The element M is split into several triangular (tetrahedral) elements T_i, $i \in \{1, ..., 4\}$.

Assumption 4.1. *We assume that $\mathcal{T}_h$ is regular, i.e. there exists a constant σ such that*

$$h_T/\rho_T \leq \sigma \tag{4.2.13}$$

for all $T \in \mathcal{T}_h$.

Clément interpolation operator on hybrid meshes

The local projection $\pi_i : L^2(\mathcal{P}_i) \to \mathbb{R}$ is given by

$$\pi_i v = \frac{1}{|\mathcal{P}_i|}\int_{\mathcal{P}_i} v dx, \tag{4.2.14}$$

where $|\mathcal{P}_i|$ is the volume of the patch $\mathcal{P}_i$. Using the value prescribed by (4.2.14) as nodal value at x_i, the Clément interpolation operator I_M is defined by

$$I_M v = \sum_{i=1}^{n_V}(\pi_i v)\varphi_i = \sum_{i=1}^{n_V}\frac{\varphi_i}{|\mathcal{P}_i|}\int_{\mathcal{P}_i} v dx. \tag{4.2.15}$$

Patchwise error estimates

For the regular subdivision $\mathcal{T}_h$, it can be seen that the number of tetrahedra in each $\mathcal{P}_i$ is bounded by a constant independent of h and i. On the other hand, the number of patches $\mathcal{P}_i$ containing a given element M is also bounded by a constant independent of i and h.

Thus, the patch $\mathcal{P}_i$ can only assume a finite number of different configurations. To each configuration, there exists a reference patch $\hat{\mathcal{P}}_i$ contained in the unit ball $\{\xi, \|\xi\| \leq 1\}$ and a continuous and invertible mapping $\Phi_{\mathcal{P}_i}$ from $\hat{\mathcal{P}}_i$ onto $\mathcal{P}_i$ such that

$$\Phi_{\mathcal{P}_i}|_{\hat{T}} \text{ is an affine mapping from } \hat{T} \text{ onto } T \tag{4.2.16}$$

for all tetrahedra $\hat{T} \subset \hat{\mathcal{P}}_i$ and the corresponding tetrahedra $T \in \mathcal{P}_i$, see [67]. See Figure 4.5 for an illustration.

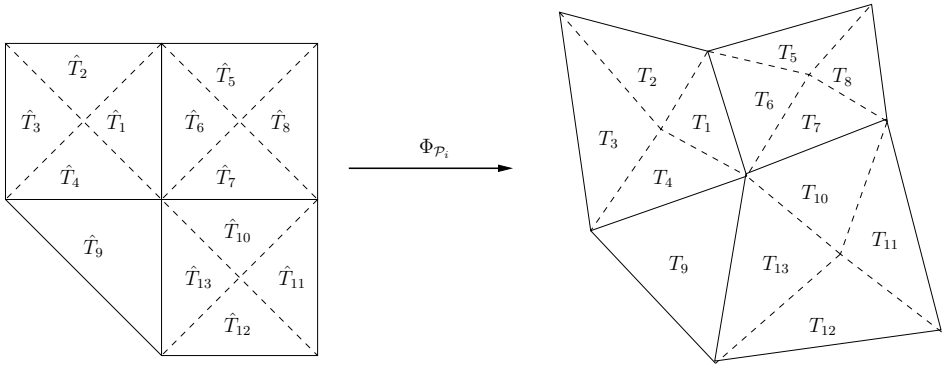


Figure 4.5: Transformation $\Phi_{\mathcal{P}_i}$ associates the reference patch $\hat{\mathcal{P}}_i$ and the physical patch $\mathcal{P}_i$.

We then have

Lemma 4.2. *If $v \in H^1(\mathcal{P}_i)$, then*

$$\|v - \pi_i v\|_{L^2(\mathcal{P}_i)} \leq 2\|v\|_{L^2(\mathcal{P}_i)}, \tag{4.2.17}$$

$$\|v - \pi_i v\|_{L^2(\mathcal{P}_i)} \leq ch_{\mathcal{P}_i}|v|_{H^1(\mathcal{P}_i)}, \tag{4.2.18}$$

$$|v - \pi_i v|_{H^1(\mathcal{P}_i)} = |v|_{H^1(\mathcal{P}_i)}, \tag{4.2.19}$$

where c is a mesh independent parameter and $h_{\mathcal{P}_i} = \sup_{T \in \mathcal{P}_i}\{h_T\}$.

Proof. We start proving (4.2.17). Using the Cauchy-Schwarz inequality (2.3.3), we have

$$\|\pi_i v\|^2_{L^2(\mathcal{P}_i)} = \frac{1}{|\mathcal{P}_i|^2}\int_{\mathcal{P}_i}\left(\int_{\mathcal{P}_i} v dx\right)^2 dx \leq \frac{1}{|\mathcal{P}_i|^2}\int_{\mathcal{P}_i}\|v\|^2_{L^2(\mathcal{P}_i)}|\mathcal{P}_i|dx = \|v\|^2_{L^2(\mathcal{P}_i)}. \tag{4.2.20}$$

Hence,

$$\|v - \pi_i v\|_{L^2(\mathcal{P}_i)} \leq 2\|v\|_{L^2(\mathcal{P}_i)}.$$

It is easy to show that (4.2.18) follows by applying the Bramble-Hilbert Lemma 2.1. It is obvious that (4.2.19) holds.

□

Elementwise error estimates

Correspondingly, we have the following element-wise error estimates.

Lemma 4.3. *If $v \in H^1(\mathcal{P})$, then*

$$\|v - I_M v\|_{L^2(M)} \leq c\|v\|_{L^2(\mathcal{P})}, \tag{4.2.21}$$

$$\|v - I_M v\|_{L^2(M)} \leq ch_{\mathcal{P}}|v|_{H^1(\mathcal{P})}, \tag{4.2.22}$$

$$|v - I_M v|_{H^1(M)} \leq c|v|_{H^1(\mathcal{P})}, \tag{4.2.23}$$

where c is a mesh independent parameter and $h_{\mathcal{P}} = \sup_{T\in\mathcal{P}}\{h_T\}$.

Proof. By the Cauchy inequality (2.3.1), using the property that φ_j forms a partition of unity, and $0 \leq \varphi_i \leq 1$ on M, we have

$$\begin{aligned}\|v - I_M v\|^2_{L^2(M)} &= \int_M \left(\sum_{i=1}^{n_V} \varphi_i (v - \pi_i v)\right)^2 dx \\ &\leq \int_M \sum_{i=1}^{n_V} \varphi_i^2 \sum_{i=1}^{n_V} (v - \pi_i v)^2 dx, \\ &\leq \sum_{i=1}^{n_V} \|v - \pi_i v\|^2_{L^2(\mathcal{P}_i)}.\end{aligned} \tag{4.2.24}$$

Now applying the estimate (4.2.17) to the last line of (4.2.24), we obtain (4.2.21). Applying the estimate (4.2.18) to the last line of (4.2.24), we obtain (4.2.22).

For the proof of (4.2.23), using the partition of unity and the triangle inequality, we have

$$\begin{aligned}|v - I_M v|^2_{H^1(M)} &= \int_M \left|\nabla\left(\sum_{i=1}^{n_V} \varphi_i (v - \pi_i v)\right)\right|^2 dx \\ &\leq \sum_{i=1}^{n_V} \int_{\mathcal{P}_i} |\nabla (\varphi_i (v - \pi_i v))|^2 dx.\end{aligned}$$

Now by using the product rule, the estimates (4.2.18) and (4.2.19), we obtain

$$|v - I_M v|_{H^1(M)} \leq c|v|_{H^1(\mathcal{P})}. \tag{4.2.25}$$

□

Global error estimates

We define the global Clément operator I_C by

$$I_C|_M = I_M \tag{4.2.26}$$

for all $M \in \mathcal{M}_h$. Because of the element-wise error estimates, it is easy to see that we have the following global error estimates.

Lemma 4.4. *If $v \in H^1(\Omega)$, then*

$$\|v - I_C v\|_{L^2(\Omega)} \leq c\|v\|_{L^2(\Omega)}, \tag{4.2.27}$$

$$\|v - I_C v\|_{L^2(\Omega)} \leq ch|v|_{H^1(\Omega)}, \tag{4.2.28}$$

$$|v - I_C v|_{H^1(\Omega)} \leq c|v|_{H^1(\Omega)}, \tag{4.2.29}$$

where c is a mesh independent parameter.

Lagrange interpolation operator on hybrid meshes

For $v \in H^2(\Omega)$, it follows from Sobolev's embedding theorem that

$$\max_{x\in\bar{\Omega}} |v(x)| \leq C_E \|v\|_{H^2(\Omega)}, \tag{4.2.30}$$

where $C_E = C_E(\Omega) > 0$ is a constant. With this regularity, we can define the Lagrange interpolation operator $I_L : H^2(\Omega) \to V_h^H$ by

$$I_L v|_M = \sum_{i=1}^{n_V} v(x_i)\varphi_i \tag{4.2.31}$$

for all $M \in \mathcal{M}_h$. Then we obtain

Lemma 4.5. *For $v \in H^2(M)$, there exists a mesh independent constant $c > 0$ such that*

$$\|v - I_L v\|_{L^2(M)} \leq c h_M |v|_{H^1(M)}, \tag{4.2.32}$$

$$|v - I_L v|_{H^1(M)} \leq c h_M |v|_{H^2(M)} \tag{4.2.33}$$

for each $M \in \mathcal{M}_h$.

Proof. Note that $I_L v = v$ for $v \in P_1$ on each $M \in \mathcal{M}_h$. The rest of the proof is quite standard. We omit it here. □

Correspondingly, we have the following global error estimates:

Lemma 4.6. *For $v \in H^2(\Omega)$, there exists a mesh independent constant $c > 0$ such that*

$$\|v - I_L v\|_{L^2(\Omega)} \leq c h |v|_{H^1(\Omega)}, \tag{4.2.34}$$

$$|v - I_L v|_{H^1(\Omega)} \leq c h |v|_{H^2(\Omega)}. \tag{4.2.35}$$

Now we will apply this extended P_1 finite element on hybrid meshes to the structure sub-problem and the harmonic extension from the FSI simulation.

4.3 Extended P_1 elements for the structure problem

We choose a hybrid mesh of the structure reference domain $\Omega = \Omega_0^{\mathrm{s}}$ in (4.2.11), denoted by $\mathcal{M}_{\mathrm{s}h}^0$ and obtain the finite element space for the displacement:

$$V_h^{\mathrm{s}} = \left[V_h^H\right]^3. \tag{4.3.1}$$

The fully discretized structure problem is given in the following.

Problem 4.7. *The discrete solution $d_h^{\mathrm{s},n+1}$ solves the discrete variational problem: Find $d_h^{\mathrm{f},n+1} \in V_h^{\mathrm{s}} \cap V_g^{\mathrm{s}}(t^{n+1})$ such that for all $v_h^{\mathrm{s}} \in V_h^{\mathrm{s}} \cap V_0^{\mathrm{s}}$,*

$$a^{\mathrm{s}}(d_h^{\mathrm{s},n+1}, v_h^{\mathrm{s}}) = \langle F_h^{\mathrm{s},n}, v_h^{\mathrm{s}} \rangle, \tag{4.3.2}$$

where the bilinear and linear forms are defined as in Problem 3.6. From the finite element solution $d_h^{\mathrm{s},n+1}$, we obtain the discrete structure domain velocity

$$w_h^{\mathrm{s},n+1} = \frac{2}{\delta t}(d_h^{\mathrm{s},n+1} - d_h^{\mathrm{s},n}) - w_h^{\mathrm{s},n}. \tag{4.3.3}$$

4.4 Extended P_1 elements for the harmonic extension

When we apply the extended P_1 element to the fluid sub-problem in the FSI problem, we firstly need to apply it to the harmonic extension. We choose a hybrid mesh of the fluid reference domain $\Omega = \Omega_0^{\mathrm{f}}$ in (4.2.11), denoted by $\mathcal{M}_{fh}^0$ and obtain the finite element space for the harmonic extension:

$$D_h^{\mathrm{f}} = \left[V_h^H\right]^3. \tag{4.4.1}$$

The discrete variational problem of the harmonic extension using P_1 element reads in the following.

Problem 4.8. *The discrete solution $d_h^{\mathrm{f},n+1}$ solves the discrete variational problem: Find $d_h^{\mathrm{f},n+1} \in D_h^{\mathrm{f}} \cap D_g^{\mathrm{f}}(t^{n+1})$ such that for all $\phi_h \in D_h^{\mathrm{f}} \cap D_0^{\mathrm{f}}$,*

$$\int_{\Omega_0^{\mathrm{f}}} \nabla d_h^{\mathrm{f},n+1} : \nabla \phi_h dx = 0. \tag{4.4.2}$$

The computational fluid domain $\Omega^{\mathrm{f}}(t^{n+1})$ will be given from the calculated fluid domain deformation by

$$\Omega^{\mathrm{f}}(t^{n+1}) = d_h^{\mathrm{f},n+1} + \Omega_0^{\mathrm{f}} \tag{4.4.3}$$

and the fluid domain velocity will be updated by

$$w_h^{\mathrm{f},n+1} = \frac{d_h^{\mathrm{f},n+1} - d_h^{\mathrm{f},n}}{\delta t} \circ \left(x_{t^{n+1}}^{\mathrm{f}}\right)^{-1}. \tag{4.4.4}$$

4.5 Stabilized finite elements on hybrid meshes for the fluid problem

For the fluid sub-problem, as we mentioned before, the standard Galerkin finite element method, i.e. applying the extended linear elements directly to the velocity and pressure spaces, may suffer from two instabilities: one is from the violation of the discrete inf-sup condition for the equal order velocity and pressure approximations, and the other is from the dominating convection, see [36, 133, 82, 23, 103, 152].

The method to overcome the inf-sup instability is analyzed in Section 4.5.1-Section 4.5.6. For illustration, we will only consider the Stokes problem under the Eulerian framework.

The combined PSGP/SUPG method for the implementation is described in Section 4.5.7.

4.5.1 Mixed finite element method for the Stokes problem

The Stokes system is given in the following problem.

Problem 4.9. *Consider the system of Stokes equations in a bounded, connected, polyhedral domain $\Omega \subset \mathbb{R}^3$ and $f \in [L^2(\Omega)]^3$: Find the velocity field u and the pressure p such that*

$$\begin{aligned} -2\mu \mathrm{div}\varepsilon(u) + \nabla p &= f \qquad \text{in } \Omega, \\ \mathrm{div} u &= 0 \qquad \text{in } \Omega, \\ \sigma_{\mathrm{f}}(u,p) n_{\mathrm{f}} &= g_{\mathrm{N}} \quad \text{on } \Gamma_{\mathrm{N}}, \\ u &= u_{\mathrm{D}} \quad \text{on } \Gamma_{\mathrm{D}}, \end{aligned} \tag{4.5.1}$$

where μ is the viscosity, Γ_{D} the boundary of domain with Dirichlet data, Γ_{N} the boundary of domain with Neumann data, n the unit outer normal vector.

For simplicity, homogeneous Dirichlet boundary conditions ($\Gamma = \Gamma_D$ and $u_{\mathrm{D}} = 0$) will be considered. The mixed variational formulation of the Stokes problem leads to:

Problem 4.10. *Find $u \in V = [H_0^1(\Omega)]^3$ and $p \in Q = L_0^2 = \{q \in L^2(\Omega) : \int_\Omega p dx = 0\}$ such that*

$$\begin{aligned} a(u,v) + b(v,p) &= \langle F, v\rangle, \\ b(u,q) &= 0 \end{aligned} \tag{4.5.2}$$

for all $v \in V$ and $q \in Q$, where $a(u,v) = \mu \int_\Omega \varepsilon(u) : \varepsilon(v) dx$, $b(v,q) = -\int_\Omega q \text{ div } v dx$, and $\langle F, v\rangle = \int_\Omega f v dx$.

For Problem 4.10, the following theorem is of central importance (see [35]).

Theorem 4.11 (Brezzi-Babuška)**.** *Let V and Q be real Hilbert spaces, $F \in V^*, G \in Q^*, a(\cdot,\cdot) : V \times V \to \mathbb{R}$ and $b(\cdot,\cdot) : V \times Q \to \mathbb{R}$ be bilinear forms. Assume that there exist constants $\alpha_1, \alpha_2, \beta_1, \beta_2 > 0$ with*

1. *$|a(u,v)| \le \alpha_2 \|u\|_V \|v\|_V$ for all $u, v \in V$,*
2. *$|b(v,q)| \le \beta_2 \|v\|_V \|q\|_Q$ for all $v \in V, q \in Q$,*
3. *$a(v,v) \ge \alpha_1 \|v\|_V^2$ for all $v \in W = \mathrm{Ker}(B) = \{v \in V : b(v,q) = 0$ for all $q \in Q\}$,*
4. *$\inf\limits_{0 \ne q \in Q} \sup\limits_{0 \ne v \in V} \dfrac{b(v,q)}{\|v\|_V \|q\|_Q} \ge \beta_1 > 0.$*

Then the variational problem above has a unique solution and we have

$$\|u\|_V \le \frac{1}{\alpha_1} \|F\|_V^*, \quad \|p\|_Q \le \frac{1}{\beta_1}(1 + \frac{\alpha_2}{\alpha_1}) \|F\|_V^*. \tag{4.5.3}$$

For Problem 4.10, the boundedness of the bilinear forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are derived from the Cauchy-Schwarz inequality with respect to the corresponding norm. The coercivity of $a(\cdot,\cdot)$ is the result of the first Korn inequality. The inf-sup condition follows from Nečas' inequality. Therefore, by Theorem 4.11, Problem 4.10 has a unique solution.

We are seeking an approximate solution of the mixed variational problem stated above by choosing appropriate finite-dimensional subspaces $V_h \subset V, Q_h \subset Q$.

Problem 4.12. *Using Galerkin's principle, the discrete solution $(u_h, p_h) \in V_h \times Q_h$ solves the discrete variational problem*

$$\begin{aligned} a(u_h, v_h) + b(v_h, p_h) &= \langle F, v_h\rangle, \\ b(u_h, q_h) &= 0, \end{aligned} \tag{4.5.4}$$

for all $v_h \in V_h$ and $q_h \in Q_h$.

We need the following lemma (see [25]) for showing the existence, uniqueness, and error estimates of a solution of Problem 4.12.

Lemma 4.13 (Generalized Lemma of Cea)**.** *Assume the notations and assumptions of the theorem of Brezzi-Babuška. Let $V_h \subset V, Q_h \subset Q$ be finite-dimensional subspaces. Assume that there exist constants $\tilde{\alpha}_1, \tilde{\beta}_1$ with*

3'. $a(v_h, v_h) \geq \tilde{\alpha}_1 \|v_h\|_V^2$ *for all* $v_h \in W_h = \mathrm{Ker}(B_h) = \{v_h \in V_h : b(v_h, q_h) = 0$ *for all* $q_h \in Q_h\}$,

4'. $\inf\limits_{0 \neq q_h \in Q_h} \sup\limits_{0 \neq v_h \in V_h} \dfrac{b(v_h, q_h)}{\|v_h\|_V \|q_h\|_Q} \geq \tilde{\beta}_1 > 0$.

Then the discrete variational problem has a unique solution $(u_h, p_h) \in V_h \times Q_h$ and we have the error estimates

$$\|u - u_h\|_V \leq (1 + \frac{\alpha_2}{\tilde{\alpha}_1})(1 + \frac{\beta_2}{\tilde{\beta}_1}) \inf_{v_h \in V_h} \|u - v_h\|_V + \frac{\beta_2}{\tilde{\alpha}_1} \inf_{q_h \in Q_h} \|p - q_h\|_Q \tag{4.5.5}$$

for the velocity and

$$\|p - p_h\|_Q \leq (1 + \frac{\alpha_2}{\tilde{\alpha}_1})(1 + \frac{\beta_2}{\tilde{\beta}_1}) \frac{\alpha_2}{\tilde{\beta}_1} \inf_{v_h \in V_h} \|u - v_h\|_V + [1 + \frac{\beta_2}{\tilde{\beta}_1}(1 + \frac{\alpha_2}{\tilde{\alpha}_1})] \inf_{q_h \in Q_h} \|p - q_h\|_Q \tag{4.5.6}$$

for the pressure, respectively.

Since the bilinear form $a(\cdot, \cdot)$ is coercive on $V = [H_0^1(\Omega)]^3$, it is also coercive on $\mathrm{Ker}B_h \subset V$. However, the discrete inf-sup condition does not generally follow from the continuous inf-sup condition. We will study the discrete inf-sup condition on hybrid meshes explicitly in Section 4.5.4.

4.5.2 MINI-element spaces on auxiliary tetrahedral meshes

Now, we introduce the P_1-P_1 element on the tetrahedral mesh $\mathcal{T}_h$, by

$$V_h^L = [V_h^T]^3 \cap [C_0(\overline{\Omega})]^3$$

and

$$Q_h^L = V_h^T \cap L_0^2(\Omega)$$

for approximating the velocity and pressure, respectively. However, this linear finite element is not stable, i.e. the inf-sup condition on the discrete level

$$\inf_{0 \neq q_h \in Q_h^L} \sup_{0 \neq v_h \in V_h^L} \frac{b(v_h, q_h)}{\|v_h\|_V \|q_h\|_Q} \geq \beta_1 > 0$$

is not fulfilled. In order to stabilize the element, the space V_h^L is enlarged by introducing a bubble function

$$b_T(x) = \begin{cases} \lambda_1 \lambda_2 \lambda_3 \lambda_4, & \text{if } x \in T, \\ 0, & \text{if } x \notin T, \end{cases} \tag{4.5.7}$$

determined by the barycentric coordinates $\lambda_i, i \in \{1, ..., 4\}$ on each tetrahedron $T \in \mathcal{T}_h$ with vertices x_i, $i \in \{1, ..., 4\}$. This bubble function vanishes at the boundary of T, i.e. $b_T(x) = 0$

for all $x \in \partial T$, and is extended by 0 to the whole domain Ω. For this bubble function, we also have the following results:

$$\int_T b_T dx = \frac{|T|}{840}, \quad \int_T |\nabla b_T|^2 dx = \frac{s_T^2}{136080|T|}, \tag{4.5.8}$$

where $|T|$ is tetrahedral volume, and $s_T^2 = \sum_{i=1}^4 s_i^2$ the sum of squares of the area of all faces. In both cases, it is easy to see that

$$\sigma(T) := \frac{|\int_T b_T dx|^2}{|T| \int_T |\nabla b_T|^2 dx} = \mathcal{O}(h_T^2), \tag{4.5.9}$$

where h_T is the diameter of the element $T \in \mathcal{T}_h$.

The extension space $\tilde{V}_h$ is defined by $\tilde{V}_h = V_h^L \oplus B_h$, where $B_h = [\text{span}\{b_T, T \in \mathcal{T}_h\}]^3$. Hence the enlarged space $\tilde{V}_h$ is expressed as follows

$$\tilde{V}_h = \{v \in C_0(\bar{\Omega})^3 : v|_T = p_T + b_T \beta_T, p_T \in [P_1]^3, \beta_T \in \mathbb{R}^3 \quad \forall T \in \mathcal{T}_h\}. \tag{4.5.10}$$

This MINI-element approximation fulfills the inf-sup condition on the discrete level, and therefore it is stable, see [35, 67, 137, 138, 157, 159].

4.5.3 Extended MINI-element spaces on hybrid meshes

The extension of the P_1-P_1 element on the subdivision $\mathcal{M}_h$ of the domain $\Omega \subset \mathbb{R}^3$ is now introduced by

$$V_h = [V_h^H]^3 \cap [C_0(\overline{\Omega})]^3 \tag{4.5.11}$$

and

$$Q_h = V_h^H \cap L_0^2(\Omega) \tag{4.5.12}$$

for the velocity and pressure, respectively.

We do not expect stability of this element. However, using the same bubble functions (4.5.7) as we defined on the tetrahedra of the split hybrid elements, one obtains the following extension of V_h on hybrid meshes

$$\bar{V}_h = \{v \in C_0(\bar{\Omega})^3 : v|_M = p_M + \sum_{\bigcup T=M} b_T \beta_T, p_M \in [\Phi_h]^3, \beta_T \in \mathbb{R}^3, \quad \forall M \in \mathcal{M}_h\}. \tag{4.5.13}$$

The stability of this extension of the MINI-element on hybrid meshes will be shown next.

4.5.4 Fortin operator

For showing the inf-sup condition on the discrete level, we use the following lemma, see [35].

Lemma 4.14. *Assume there exists a linear operator $\Pi_h : V \to \bar{V}_h$ such that*
1. $b(\Pi_h v, q_h) = b(v, q_h)$ for all $q_h \in Q_h$ and all $v \in V$,
2. $\|\Pi_h v\|_V \leq c\|v\|_V$ for all $v \in V$ with a constant c independent of h.
Then the finite element spaces $\bar{V}_h$ and Q_h satisfy the inf-sup condition.

Π_h is called a Fortin operator. For the hybrid mesh, we obtain

$$\begin{aligned} b(\Pi_h v - v, q_h) &= \int_\Omega q_h \mathrm{div}(\Pi_h v - v) dx = -\int_\Omega (\Pi_h v - v) \cdot \nabla q_h dx \\ &= -\sum_{M \in \mathcal{M}_h} \sum_{T \in M} \int_T (\Pi_h v - v) \cdot \nabla q_h dx \\ &= -\sum_{M \in \mathcal{M}_h} \sum_{T \in M} \nabla q_h \cdot \int_T (\Pi_h v - v) dx. \end{aligned} \tag{4.5.14}$$

Thus a sufficient condition satisfying $b(\Pi_h v, q_h) = b(v, q_h)$ is

$$\int_T \Pi_h v dx = \int_T v dx \tag{4.5.15}$$

for all $T \in \mathcal{T}_h$.

We use the following ansatz for Π_h:

$$\Pi_h v|_M = I_M v + \sum_{T \in M} b_T \beta_T \tag{4.5.16}$$

for all $M \in \mathcal{M}_h$, where I_M is defined in (4.2.15) and applied to each component of v. By inserting (4.5.16) in (4.5.15), we obtain

$$\beta_T = \frac{\int_T (v - I_M v)\, dx}{\int_T b_T dx}. \tag{4.5.17}$$

This leads to the definition of Π_h from V to $\bar{V}_h$:

$$\Pi_h v = I_C v + I_B (v - I_C v), \tag{4.5.18}$$

where I_C is defined in (4.2.26) and applied to each component of v, and I_B is a linear operator given by

$$I_B v|_M = \sum_{T \in M} b_T \frac{\int_T v dx}{\int_T b_T dx} \tag{4.5.19}$$

for all $M \in \mathcal{M}_h$.

We then have

Lemma 4.15. *Under the Assumption 4.1 on hybrid meshes, the operator Π_h is bounded with respect to the H^1 norm, i.e. $\|\Pi_h v\|_{H^1(\Omega)} \leq c\|v\|_{H^1(\Omega)}$ for all $v \in H^1(\Omega)$ with mesh independent constant $c > 0$.*

Proof. Inequalities (4.2.27) and (4.2.29) imply

$$\|v - I_C v\|_{H^1(\Omega)} \leq c\|v\|_{H^1(\Omega)}. \tag{4.5.20}$$

Thus, by triangle inequality, we obtain

$$\|I_C v\|_{H^1(\Omega)} \leq (1 + c)\|v\|_{H^1(\Omega)}. \tag{4.5.21}$$

We have

$$\begin{aligned}\|I_B v\|_{L^2(M)} &\leq \sum_{T\in M} \|b_T\|_{L^2(T)} \left|\frac{\int_T v dx}{\int_T b_T dx}\right| = \sum_{T\in M} \sqrt{\int_T b_T^2 dx} \left|\frac{\int_T v dx}{\int_T b_T dx}\right| \\ &\leq \sum_{T\in M} \sqrt{\int_T b_T dx} \left|\frac{\int_T v dx}{\int_T b_T dx}\right| \leq \sqrt{\frac{|T|}{\int_T b_T dx}} \sqrt{\int_T v^2 dx}.\end{aligned} \tag{4.5.22}$$

Now by using the first idenitities in (4.5.8), we obtain

$$\|I_B v\|_{L^2(M)} \leq c\|v\|_{L^2(M)} \tag{4.5.23}$$

with a mesh independent constant c.

On the other hand, by using (4.5.9), we have

$$\begin{aligned}|I_B v|_{H^1(M)} &\leq \sum_{T\in M} \left| b_T \frac{\int_T v dx}{\int_T b_T dx}\right|_{H^1(T)} \leq \sum_{T\in M} |b_T|_{H^1(T)} \left|\frac{\int_T v dx}{\int_T b_T dx}\right| \\ &= \sum_{T\in M} \sqrt{\frac{\int_T |\nabla b_T|^2 dx |T|}{|\int_T b_T dx|^2 |T|}} \left|\int_T v dx\right| \leq \sum_{T\in M} c\frac{h_T^{-1}}{\sqrt{|T|}} \left|\int_T v dx\right| \\ &\leq \sum_{T\in M} c h_T^{-1} |v|_{L^2(T)} \leq c h_T^{-1} |v|_{L^2(M)}.\end{aligned} \tag{4.5.24}$$

Now collecting (4.5.22) and (4.5.24), and using (4.2.28) in (4.5.24), we obtain

$$\|I_B (v - I_C v)\|_{H^1(\Omega)} \leq c\|v\|_{H^1(\Omega)} \tag{4.5.25}$$

with a mesh independent constant c.

From (4.5.21) and (4.5.25), the result follows. □

The constructed operator Π_h satisfies all conditions of Lemma 4.14. Hence Π_h is a Fortin operator and the discrete inf-sup condition on hybrid meshes fulfills. This is summarized in the following theorem.

Theorem 4.16. *For a hybrid mesh with regular triangular subdivisions, the spaces $\bar{V}_h$ and Q_h fulfill the discrete inf-sup condition with a constant $\bar{\beta}$ independent of h*

$$\inf_{0\neq q_h\in Q_h} \sup_{0\neq v_h\in \bar{V}_h} \frac{b(v_h, q_h)}{\|v_h\|_V \|q_h\|_Q} \geq \bar{\beta} > 0. \tag{4.5.26}$$

4.5.5 An a priori error estimate

Since Theorem 4.16 holds for the spaces $\bar{V}_h(\Omega)$ and $Q_h(\Omega)$, and the coercivity of $a(\cdot,\cdot)$ bilinear form also holds in the space $\bar{V}_h(\Omega) \subset [H_0^1(\Omega)]^3$, one can apply the generalized Cea's Lemma 4.13: the discretization error can be bounded by the approximation error. On the other hand the later can be bounded by the interpolation error:

$$\inf_{v_h\in\bar{V}_h} \|u - u_h\|_{H^1(\Omega)} \leq \|u - I_L u\|_{H^1(\Omega)}, \tag{4.5.27}$$

$$\inf_{p_h \in Q_h} \|p - p_h\|_{L^2(\Omega)} \leq \|p - I_C p\|_{L^2(\Omega)}, \tag{4.5.28}$$

where I_L is defined in (4.2.31) and applied to each component of u.

Now using estimates (4.2.34), (4.2.35) and (4.2.28), we arrive at an a priori error estimate for hybrid meshes on the spaces $\bar{V}_h$ and Q_h.

Theorem 4.17. *For this extension of the MINI-element approximation of the Stokes system on hybrid meshes, we have the following error estimate for the velocity and pressure*

$$\|u - u_h\|_{H^1(\Omega)} + \|p - p_h\|_{L^2(\Omega)} \leq ch \left(\|u\|_{H^2(\Omega)} + |p|_{H^1(\Omega)} \right) \tag{4.5.29}$$

for all $u_h \in \bar{V}_h$ and $p_h \in Q_h$ provided $u \in [H^2(\Omega)]^3$ and $p \in H^1(\Omega)$.

4.5.6 Static condensation

In this subsection, we will eliminate the bubble degrees of freedom on each hybrid element (*static condensation*). We have the ansatz

$$u_h = u_h^L + u_h^B, \tag{4.5.30}$$

where $u_h^L \in V_h$ and $u_h^B = \sum_{M \in \mathcal{M}_h} \sum_{T \in M} b_T \beta_T \in B_h$. The coefficient β_T is defined as

$$\beta_T = \sum_{j=1}^{3} \beta_{T,j} e_j \tag{4.5.31}$$

where $\beta_{T,j}$ is a real number and e_j the j-th unity vector in $\mathbb{R}^3$. We firstly have an orthogonality property associated with Problem 4.12.

Lemma 4.18. *We have*

$$a(u_h^L, w_h) = 0 \tag{4.5.32}$$

for all $u_h^L \in V_h, w_h \in B_h$. Furthermore, we have

$$a(\beta_T b_T, b_T e_i) = \frac{\mu}{2} \beta_{T,i} \delta_T + \frac{\mu}{2} \int_T \sum_j \frac{\partial b_T}{\partial x_i} \frac{\partial b_T}{\partial x_j} \beta_{T,j} dx. \tag{4.5.33}$$

Proof. Using the ansatz $u_h^L|_{M(\supset T)} = \sum_{i=1}^{n} u_j \varphi_j$, where $u_j \in \mathbb{R}^3$ and n is the number of vertices of M, and noticing that each φ_j is a linear combination of the $\tilde{\varphi}_k$, we only need to consider the $\tilde{\varphi}_k$ which have support on T. By integration by parts over each $T \in \mathcal{T}_h$, $b_T \in H_0^1(T)$, it is easy to see that

$$a(u_h^L, b_T e_i) = 0. \tag{4.5.34}$$

We have

$$\begin{aligned} a(\beta_T b_T, b_T e_i) =& \mu \int_T \varepsilon \left(b_T \sum_j \beta_{T,j} e_j \right) : \nabla (b_T e_i) \, dx \\ =& \frac{\mu}{2} \sum_j \int_T \nabla (b_T \beta_{T,j} e_j) : \nabla (b_T e_i) \, dx + \frac{\mu}{2} \int_T \sum_j \beta_{T,j} \left(\nabla (b_T e_j) \right)^T : \nabla (b_T e_i) \\ =& \frac{\mu}{2} \beta_{T,i} \delta_T + \frac{\mu}{2} \int_T \sum_j \frac{\partial b_T}{\partial x_i} \frac{\partial b_T}{\partial x_j} \beta_{T,j} dx, \end{aligned} \tag{4.5.35}$$

where $\delta_T = \int_T |\nabla b_T|^2 dx$.

□

Next, we will eliminate the additional degrees of freedom β_T on each $T \in \mathcal{T}_h$.

Lemma 4.19. *The coefficient β_T can be determined in terms of the bubble function b_T, the discrete solution p_h and the given right hand side f:*

$$\beta_T = A_T^{-1} g_T, \tag{4.5.36}$$

where

$$g_T = \left(\int_T b_T f dx - \int_T \nabla p_h b_T dx \right) \tag{4.5.37}$$

and

$$A_T = (a_{ij}) = (a_{ji}) = \begin{cases} \frac{\mu}{2} \left(\delta_T + \int_T \left(\frac{\partial b_T}{\partial x_i} \right)^2 dx \right) & \text{if } i = j, \\ \frac{\mu}{2} \int_T \frac{\partial b_T}{\partial x_i} \frac{\partial b_T}{\partial x_j} dx & \text{if } i \neq j. \end{cases} \tag{4.5.38}$$

Proof. Testing the momentum equation in Problem 4.12 with $b_T e_i$ for each $T \in M$ and using Lemma 4.18, we have

$$\begin{aligned} \langle F, b_T e_i \rangle &= a(u_h, b_T e_i) + b(b_T e_i, p_h) = a(u_h^B, b_T e_i) + a(u_h^L, b_T e_i) + b(b_T e_i, p_h) \\ &= a(\beta_T b_T, b_T e_i) + b(b_T e_i, p_h). \end{aligned}$$

By elementary manipulation, it ends up with a linear system of equations on each $T \in \mathcal{T}_h$:

$$\frac{\mu}{2} \begin{pmatrix} \delta_T + \int_T \frac{\partial b_T}{\partial x_1} \frac{\partial b_T}{\partial x_1} dx & \int_T \frac{\partial b_T}{\partial x_1} \frac{\partial b_T}{\partial x_2} & \int_T \frac{\partial b_T}{\partial x_1} \frac{\partial b_T}{\partial x_3} \\ \int_T \frac{\partial b_T}{\partial x_2} \frac{\partial b_T}{\partial x_1} dx & \delta_T + \int_T \frac{\partial b_T}{\partial x_2} \frac{\partial b_T}{\partial x_2} & \int_T \frac{\partial b_T}{\partial x_2} \frac{\partial b_T}{\partial x_3} \\ \int_T \frac{\partial b_T}{\partial x_3} \frac{\partial b_T}{\partial x_1} dx & \int_T \frac{\partial b_T}{\partial x_3} \frac{\partial b_T}{\partial x_2} & \delta_T + \int_T \frac{\partial b_T}{\partial x_3} \frac{\partial b_T}{\partial x_3} \end{pmatrix} \begin{pmatrix} \beta_{T,1} \\ \beta_{T,2} \\ \beta_{T,3} \end{pmatrix} = \int_T b_T \left(f - \nabla p_h \right) dx, \tag{4.5.39}$$

i.e.

$$A_T \beta_T = g_T. \tag{4.5.40}$$

By solving it, we obtain the claimed result. □

Now using the mass equation in Problem 4.12, we obtain

Lemma 4.20. *The mass equation in Problem 4.12 can be reduced to*

$$b(u_h^L, q_h) - \bar{c}_h(p_h, q_h) = \langle \bar{G}_h, q_h \rangle \tag{4.5.41}$$

for all $q_h \in Q_h$, where $\bar{c}_h(p_h, q_h)$ and the right hand side $\langle \bar{G}_h, q_h \rangle$ are given by

$$\bar{c}_h(p_h, q_h) = \sum_{M \in \mathcal{M}_h} \sum_{T \in M} \gamma(T) \int_T A_T^{-1} \nabla p_h \cdot \nabla q_h dx \tag{4.5.42}$$

and

$$\langle \bar{G}_h, q_h \rangle = - \sum_{M \in \mathcal{M}_h} \sum_{T \in M} \gamma(T) \int_T A_T^{-1} \bar{f}(T) \cdot \nabla q_h dx, \tag{4.5.43}$$

respectively, and $\gamma(T)$ is a mesh dependent parameter given by

$$\gamma(T) = \frac{\left(\int_T b_T dx\right)^2}{|T|} \tag{4.5.44}$$

with $\bar{f}(T)$

$$\bar{f}(T) = \frac{\int_T b_T f dx}{\int_T b_T dx}. \tag{4.5.45}$$

Proof. Using the ansatz $u_h = u_h^L + u_h^B$ for the mass equation in Problem 4.12 and by integration by parts over each $T \in \mathcal{T}_h$, one obtains

$$0 = -\int_\Omega q_h \text{ div } u_h^L dx + \sum_{M\in\mathcal{M}_h}\sum_{T\in M}\int_T b_T \beta_T \cdot \nabla q_h dx. \tag{4.5.46}$$

Using β_T calculated in Lemma 4.19, we arrive at

$$\begin{aligned}
\int_\Omega q_h \text{ div } u_h^L dx &= \sum_{M\in\mathcal{M}_h}\sum_{T\in M}\left(\int_T b_T A_T^{-1}\int_T b_T f dx\cdot\nabla q_h dx - \int_T b_T A_T^{-1}\int_T b_T \nabla p_h dx\cdot\nabla q_h dx\right)\\
&= \sum_{M\in\mathcal{M}_h}\sum_{T\in M}|T|\left(\nabla q_h\cdot\frac{\left(\int_T b_T dx\right)^2 A_T^{-1}\int_T b_T f dx}{\int_T b_T dx |T|} - \nabla q_h\cdot A_T^{-1}\nabla p_h \frac{\left(\int_T b_T dx\right)^2}{|T|}\right)\\
&= \sum_{M\in\mathcal{M}_h}\sum_{T\in M}\gamma(T)|T|\left(\nabla q_h\cdot A_T^{-1}\bar{f}(T) - \nabla q_h\cdot A_T^{-1}\nabla p_h\right).
\end{aligned} \tag{4.5.47}$$

Since ∇q_h and ∇p_h are piecewise constant on each $T \in \mathcal{T}_h$, we have

$$\nabla q_h \cdot A_T^{-1}\bar{f}(T) = \frac{\int_T \nabla q_h \cdot A_T^{-1}\bar{f}(T) dx}{|T|} \tag{4.5.48}$$

and

$$\nabla q_h \cdot A_T^{-1}\nabla p_h = \frac{\int_T \nabla q_h \cdot A_T^{-1}\nabla p_h dx}{|T|}. \tag{4.5.49}$$

By inserting (4.5.48) and (4.5.49) into the last line of (4.5.47), the claimed result follows. □

Testing the momentum equation in Problem 4.12 with $v_h^L \in V_h$ and using Lemma 4.18 we obtain

$$a(u_h^L, v_h^L) + b(v_h^L, p_h) = \langle F, v_h^L\rangle \tag{4.5.50}$$

for all $v_h^L \in V_h$. Collecting equations (4.5.41) and (4.5.50), the discrete mixed formulation after static condensation is summarized in the following.

Problem 4.21. *Find $u_h^L \in V_h$ and $p_h \in Q_h$ such that*

$$a(u_h^L, v_h^L) + b(v_h^L, p_h) = \langle F, v_h^L\rangle, \tag{4.5.51}$$

$$b(u_h^L, q_h) - \bar{c}_h(p_h, q_h) = \langle \bar{G}_h, q_h\rangle \tag{4.5.52}$$

for all $v_h^L \in V_h$ and $q_h \in Q_h$.

Remark 4.22. *It is easy to see that $\gamma(T)A_T^{-1}$ in (4.5.42) and (4.5.43) is in order of h_T^2. Thus we would suggest using the so-called residual-based stabilization technique introduced in connection with the Stokes flow by adding element-wise terms to the left hand side of the discrete weak form of the momentum equation*

$$\sum_{M\in\mathcal{M}_h}\sum_{T\in M}\eta(T)\left(-\nabla\cdot\varepsilon(u)+\nabla p,-\nabla\cdot\varepsilon(v)+\nabla q\right)_T \tag{4.5.53}$$

and the right hand side for keeping the consistency

$$\sum_{M\in\mathcal{M}_h}\sum_{T\in M}\eta(T)\left(f,-\nabla\cdot\varepsilon(v)+\nabla q\right), \tag{4.5.54}$$

where $\eta(T)=\mathcal{O}\left(h_T^2\right)$, see [80, 60]. For continuous piecewise linear functions, they will be reduced to

$$\sum_{M\in\mathcal{M}_h}\sum_{T\in M}\eta(T)\left(\nabla p,\nabla q\right)_T \tag{4.5.55}$$

and

$$\sum_{M\in\mathcal{M}_h}\sum_{T\in M}\eta(T)\left(\bar{f},\nabla q\right). \tag{4.5.56}$$

4.5.7 SUPG/PSPG stabilization for the ALE fluid problem

This method combines the pressure stabilization Petrov-Galerkin method (PSPG) and the streamline upwind Petrov-Galerkin method (SUPG) (see [151, 36, 133, 82, 23, 103, 152]), which overcome the instability from the violation of the discrete inf-sup condition and the dominating convection, respectively. Additionally if the diffusion term is too small, i.e. $0<\mu\ll 1$, an element-wise stabilization of divergence constraint denoted as *grad-div stabilization* is important (see [151]).

We choose a hybrid mesh of the fluid domain $\Omega=\Omega^{\mathrm{f}}(t^{n+1})$ in (4.2.11), denoted by $\mathcal{M}_{\mathrm{f}h}$ and obtain the finite element spaces

$$V_h^{\mathrm{f}}(t^{n+1})=\left[V_h^H\right]^3 \tag{4.5.57}$$

and

$$Q_h^{\mathrm{f}}(t^{n+1})=V_h^H \tag{4.5.58}$$

for the velocity and pressure, respectively. Note that the subdivision $\mathcal{M}_{\mathrm{f}h}$ is obtained from the ALE mapping, i.e. $\mathcal{M}_{\mathrm{f}h}=\mathcal{M}_{\mathrm{f}h}^0\circ\left(x_{t^{n+1}}^{\mathrm{f}}\right)^{-1}$.

The approach adds the following terms to the momentum equation in (3.2.16):

$$\begin{aligned}\sum_{M\in\mathcal{M}_{\mathrm{f}h}}\sum_{T\in M}\int_T&\left(\rho_{\mathrm{f}}\left.\frac{\partial u_h}{\partial t}\right|_{x_0}+\rho_{\mathrm{f}}\left(\left(u_h-w_h^{\mathrm{f}}\right)\cdot\nabla\right)u_h-2\mu\mathrm{div}\varepsilon(u_h)+\nabla p_h\right)\\&\cdot\left(\delta_T^{\mathrm{p}}\nabla q_h^{\mathrm{f}}+\delta_T^{\mathrm{u}}\rho_{\mathrm{f}}\left(\left(u_h-w_h^{\mathrm{f}}\right)\cdot\nabla\right)v_h^{\mathrm{f}}\right)dx\end{aligned} \tag{4.5.59}$$

and

$$\sum_{M\in\mathcal{M}_h}\sum_{T\in M}\int_T\gamma_T\nabla\cdot u_h\nabla\cdot v_h^{\mathrm{f}}dx, \tag{4.5.60}$$

where, according to [23], δ_T^{u}, δ_T^{p}, and γ_T are given by

$$\delta_T^{\mathrm{u}} = \delta_T^{\mathrm{p}} = \delta_T \sim \left(\frac{r^4 \mu}{h_T^2} + \frac{r \|\omega_h\|_{L^\infty(T)}}{h_T} + \|c\|_{L^\infty(T)} \right)^{-1}, \quad \gamma_T \sim \frac{h_T^2}{r^2 \delta_T} \tag{4.5.61}$$

with the polynomial degree r for the finite element space.

In the P_1-P_1 case, for the fluid sub-problem under the ALE framework, we have

$$r = 1, \quad \omega_h = \rho_{\mathrm{f}}(u_h - w_h^{\mathrm{f}}), \quad c = \rho_{\mathrm{f}} \left(\frac{1}{\delta t} - \mathrm{div} w_h^{\mathrm{f}} \right). \tag{4.5.62}$$

The discrete stabilized ALE weak formulation leads to the following problem.

Problem 4.23. *Find* $(u_h^{n+1}, p_h^{n+1}) = (u_h(t^{n+1}), p_h(t^{n+1})) \in \big(V_h^{\mathrm{f}}(t^{n+1}) \cap V_g^{\mathrm{f}}(t^{n+1})\big) \times Q_h^{\mathrm{f}}(t^{n+1})$ *such that for all* $(v_h^{\mathrm{f}}, q_h^{\mathrm{f}}) \in \big(V_h^{\mathrm{f}}(t^{n+1}) \cap V_0^{\mathrm{f}}(t^{n+1})\big) \times Q_h^{\mathrm{f}}(t^{n+1})$,

$$\begin{cases} \tilde{a}(u_h^{n+1}, v_h^{\mathrm{f}}) + \tilde{b}_1(v_h^{\mathrm{f}}, p_h^{n+1}) = \langle \tilde{F}_h^{\mathrm{f},n+1}, v_h^{\mathrm{f}} \rangle, \\ \tilde{b}_2(u_h^{n+1}, q_h^{\mathrm{f}}) - \tilde{c}(p_h^{n+1}, q_h^{\mathrm{f}}) = \langle \tilde{G}_h^{\mathrm{f},n+1}, q_h^{\mathrm{f}} \rangle, \end{cases} \tag{4.5.63}$$

with

$$\begin{cases} \tilde{a}(u_h^{n+1}, v_h^{\mathrm{f}}) = a^{\mathrm{f}}(u_h^{n+1}, v_h^{\mathrm{f}}) + a_s^{\mathrm{f}}(u_h^{n+1}, v_h^{\mathrm{f}}), \\ \tilde{b}_1(v_h^{\mathrm{f}}, p_h^{n+1}) = b(v_h^{\mathrm{f}}, p_h^{n+1}) + b_{1,s}(v_h^{\mathrm{f}}, p_h^{n+1}), \\ \tilde{b}_2(u_h^{n+1}, q_h^{\mathrm{f}}) = b(u_h^{n+1}, q_h^{\mathrm{f}}) + b_{2,s}(u_h^{n+1}, q_h^{\mathrm{f}}), \\ \tilde{c}(p_h^{n+1}, q_h^{\mathrm{f}}) = c_s(p_h^{n+1}, q_h^{\mathrm{f}}), \\ \langle \tilde{F}_h^{\mathrm{f},n+1}, v_h^{\mathrm{f}} \rangle = \langle F^{\mathrm{f},n+1}, v_h^{\mathrm{f}} \rangle + \langle F_h^{\mathrm{f},n+1}, v_h^{\mathrm{f}} \rangle_s \\ \langle \tilde{G}_h^{\mathrm{f},n+1}, q_h^{\mathrm{f}} \rangle = \langle G_h^{\mathrm{f},n+1}, q_h^{\mathrm{f}} \rangle_s, \end{cases} \tag{4.5.64}$$

where $a^{\mathrm{f}}(\cdot,\cdot)$, $b(\cdot,\cdot)$, *and* $\langle F^{\mathrm{f},n+1}, \cdot \rangle$ *are given by (3.3.20) in Problem 3.7.*

The stabilization terms with subscription s *are defined by*

$$\begin{aligned} a_s^{\mathrm{f}}(u_h^{n+1}, v_h^{\mathrm{f}}) = & \sum_{M \in \mathcal{M}_{\mathrm{f}h}} \sum_{T \in M} \int_T \gamma_T \nabla \cdot u_h^{n+1} \nabla \cdot v_h^{\mathrm{f}} dx \\ & + \sum_{M \in \mathcal{M}_{\mathrm{f}h}} \sum_{T \in M} \delta_T \int_T \rho_{\mathrm{f}} (\theta_h^{n+1} \cdot \nabla) u_h^{n+1} \cdot \rho_{\mathrm{f}} (\theta_h^{n+1} \cdot \nabla) v_h^{\mathrm{f}} dx \\ & + \frac{1}{\delta t} \sum_{M \in \mathcal{M}_{\mathrm{f}h}} \sum_{T \in M} \delta_T \int_T \rho_{\mathrm{f}} u_h^{n+1} \cdot \rho_{\mathrm{f}} (\theta_h^{n+1} \cdot \nabla) v_h^{\mathrm{f}} dx \\ & - \sum_{M \in \mathcal{M}_{\mathrm{f}h}} \sum_{T \in M} \delta_T \int_T \mathrm{div} w_h^{\mathrm{f},n+1} \rho_{\mathrm{f}} u_h^{n+1} \cdot \rho_{\mathrm{f}} (\theta_h^{n+1} \cdot \nabla) v_h^{\mathrm{f}} dx, \end{aligned} \tag{4.5.65}$$

$$b_{1,s}(v_h^{\mathrm{f}}, p_h^{n+1}) = \sum_{M \in \mathcal{M}_{\mathrm{f}h}} \sum_{T \in M} \delta_T \int_T \nabla p_h^{n+1} \cdot \rho_{\mathrm{f}} (\theta_h^{n+1} \cdot \nabla) v_h^{\mathrm{f}} dx, \tag{4.5.66}$$

$$\begin{aligned} b_{2,s}(u_h^{n+1}, q_h^{\mathrm{f}}) = & - \frac{1}{\delta t} \sum_{M \in \mathcal{M}_{\mathrm{f}h}} \sum_{T \in M} \delta_T \int_T \rho_{\mathrm{f}} u_h^{n+1} \cdot \nabla q_h^{\mathrm{f}} dx \\ & + \sum_{M \in \mathcal{M}_{\mathrm{f}h}} \sum_{T \in M} \delta_T \int_T \mathrm{div} w_h^{\mathrm{f},n+1} \rho_{\mathrm{f}} u_h^{n+1} \cdot \nabla q_h^{\mathrm{f}} dx \\ & - \sum_{M \in \mathcal{M}_{\mathrm{f}h}} \sum_{T \in M} \delta_T \int_T \rho_{\mathrm{f}} (\theta_h^{n+1} \cdot \nabla) u_h^{n+1} \cdot \nabla q_h^{\mathrm{f}} dx, \end{aligned} \tag{4.5.67}$$

$$c_s(p_h^{n+1}, q_h^{\mathrm{f}}) = \sum_{M\in\mathcal{M}_{\mathrm{f}h}} \sum_{T\in M} \delta_T \int_T \nabla p_h^{n+1} \cdot \nabla q_h^{\mathrm{f}} dx, \tag{4.5.68}$$

and the corresponding right hand sides are defined by

$$\langle F_h^{\mathrm{f},n+1}, v_h^{\mathrm{f}}\rangle_s = \frac{1}{\delta t} \sum_{M\in\mathcal{M}_{\mathrm{f}h}^n} \sum_{T\in M} \delta_T \int_T \rho_{\mathrm{f}} u_h^n \cdot \rho_{\mathrm{f}}(\theta_h^n \cdot \nabla)\bar{v}_h^{\mathrm{f}} dx, \tag{4.5.69}$$

$$\langle G_h^{\mathrm{f},n+1}, q_h^{\mathrm{f}}\rangle_s = -\frac{1}{\delta t} \sum_{M\in\mathcal{M}_{\mathrm{f}h}^n} \sum_{T\in M} \delta_T \int_T \rho_{\mathrm{f}} u_h^n \cdot \nabla \bar{q}_h^{\mathrm{f}} dx, \tag{4.5.70}$$

where $\theta_h^{n+1} = u_h^n \circ x_{t^n}^{\mathrm{f}} \circ \left(x_{t^{n+1}}^{\mathrm{f}}\right)^{-1} - w_h^{\mathrm{f},n+1}$, $\bar{v}_h^{\mathrm{f}} = v_h^{\mathrm{f}} \circ x_{t^{n+1}}^{\mathrm{f}} \circ \left(x_{t^n}^{\mathrm{f}}\right)^{-1}$, $\bar{q}_h^{\mathrm{f}} = q_h^{\mathrm{f}} \circ x_{t^{n+1}}^{\mathrm{f}} \circ \left(x_{t^n}^{\mathrm{f}}\right)^{-1}$, *and* $\mathcal{M}_{\mathrm{f}h}^n$ *is a subdivision of the domain* $\Omega(t^n)$ *at the previous time step* t^n.

Problem 4.23 will lead to an indefinite and unsymmetric linear system.

4.6 System assembling of FSI sub-problems

In this section, we will give details for the finite element implementation, namely, the element-wise assembling process. For representing a vector-valued function c_h of the finite element solution, we use the so-called *point-wise ordering*, i.e.

$$c_h = \sum_{j=1}^{n} c_j \varphi_j, \tag{4.6.1}$$

where φ_j is a scalar basis function of V_h^H, c_j is a coefficient from $\mathbb{R}^3$, say $c_j = (c_j^1, c_j^2, c_j^3)^T$, and n is the total number of vertices.

4.6.1 Structure system assembling

Using *point-wise ordering*, the discrete solution d_h^{s} of the displacement at time t^{n+1} will be represented by the anstaz

$$d_h^{\mathrm{s}} = \sum_{j=1}^{n} d_j^{\mathrm{s}} \varphi_j, \tag{4.6.2}$$

where $d_j^{\mathrm{s}} = (d_j^1, d_j^2, d_j^3)^T \in \mathbb{R}^3$ and φ_j is a basis function for each component of the space V_h^{s}.

We choose $v_h^{\mathrm{s}} = e_\beta \varphi_i$ in (4.3.2) in Problem 4.7 for $\beta = 1, 2, 3$:

$$a(\sum_{j=1}^{n}\sum_{\alpha=1}^{3} e_\alpha d_j^\alpha \varphi_j, e_\beta \varphi_i) = \langle F^{\mathrm{s}}, e_\beta \varphi_i\rangle \tag{4.6.3}$$

for $i = 1, ..., n$.

From this we obtain the following linear system of equations:

$$A^{\mathrm{s}} \underline{d}^{\mathrm{s}} = \underline{f}^{\mathrm{s}} \tag{4.6.4}$$

with A^{s} containing 3×3 block matrices $A_{i,j}^{\beta,\alpha}$ for $\beta, \alpha \in \{1, 2, 3\}$ as its entries, $\underline{d}^{\mathrm{s}} = (d_i^{\mathrm{s}})$, and $\underline{f}^{\mathrm{s}} \in (\mathbb{R}^3)^n$ containing $\underline{f}_i^\beta$ for $\beta \in \{1, 2, 3\}$ as its entries.

Using the definitions of $a(\cdot,\cdot)$ and $\langle F^{\mathrm{s},n},\cdot\rangle$ in Problem 3.6, we obtain the entries

$$\begin{aligned}
A_{i,j}^{\beta,\alpha} &= \sum_{M\in\mathcal{M}_{\mathrm{sh}}^0}\sum_{T\in M}\int_T \frac{2}{\delta t^2}\rho_{\mathrm{s}} e_\alpha\varphi_j\cdot e_\beta\varphi_i dx + \sum_{M\in\mathcal{M}_{\mathrm{sh}}^0}\sum_{T\in M}\int_T \lambda^l \mathrm{div}(e_\alpha\varphi_j)\mathrm{div}\left(e_\beta\varphi_i\right)dx \\
&\quad + \sum_{M\in\mathcal{M}_{\mathrm{sh}}^0}\sum_{T\in M}\int_T 2\mu^l\varepsilon(e_\alpha\varphi_j):\varepsilon(e_\beta\varphi_i)dx \\
&= \sum_{M\in\mathcal{M}_{\mathrm{sh}}^0}\sum_{T\in M}\int_T \lambda^l\frac{\partial\varphi_j}{\partial x_\alpha}\frac{\partial\varphi_i}{\partial x_\beta}dx + \begin{cases} \displaystyle\sum_{M\in\mathcal{M}_{\mathrm{sh}}^0}\sum_{T\in M}\int_T \frac{2}{\delta t^2}\rho_{\mathrm{s}}\varphi_j\varphi_i dx & (\text{if } \alpha=\beta) \\ 0 & (\text{if } \alpha\neq\beta) \end{cases} \\
&\quad + \begin{cases} \displaystyle\sum_{M\in\mathcal{M}_{\mathrm{sh}}^0}\sum_{T\in M}\int_T \mu^l\left(\sum_{l=1}^3\frac{\partial\varphi_j}{\partial x_l}\frac{\partial\varphi_i}{\partial x_l}+\frac{\partial\varphi_j}{\partial x_\alpha}\frac{\partial\varphi_i}{\partial x_\alpha}\right)dx & (\text{if } \alpha=\beta) \\ \displaystyle\sum_{M\in\mathcal{M}_{\mathrm{sh}}^0}\sum_{T\in M}\int_T \mu^l\frac{\partial\varphi_j}{\partial x_\beta}\frac{\partial\varphi_i}{\partial x_\alpha}dx & (\text{if } \alpha\neq\beta) \end{cases}
\end{aligned} \tag{4.6.5}$$

and

$$\underline{f}_i^\beta = \sum_{M\in\mathcal{M}_{\mathrm{sh}}^0}\sum_{T\in M}\int_T \frac{2}{\delta t^2}\rho_{\mathrm{s}}(d_h^{\mathrm{s},n}+\delta t w_h^{\mathrm{s},n})\cdot e_\beta\varphi_i dx, \tag{4.6.6}$$

where $d_h^{\mathrm{s},n}$ and $w_h^{\mathrm{s},n}$ are finite element solutions from the previous time step.

Note that the final system has to include the Dirichlet boundary conditions which will lead to some changes for the matrix A^{s} and the right hand side $\underline{f}^{\mathrm{s}}$. Here we will not detail how to numerically incorporate Dirichlet boundary conditions into the system.

4.6.2 Harmonic extension system assembling

Using *point-wise ordering*, the discrete solution d_h^{f} of the displacement for the initial fluid domain at time t^{n+1} will be represented by the anstaz:

$$d_h^{\mathrm{f}} = \sum_{j=1}^n d_j^{\mathrm{f}}\varphi_j, \tag{4.6.7}$$

where $d_j^{\mathrm{f}} = (d_j^1, d_j^2, d_j^3)^T \in \mathbb{R}^3$ and φ_j is a basis function for each component of the space D_h^{f}.

We choose $\phi_h = e_\beta\varphi_i$ in (4.4.2) in Problem 4.8 for $\beta = 1,2,3$:

$$\int_{\Omega_0^{\mathrm{f}}}\nabla\left(\sum_{j=1}^n\sum_{\alpha=1}^3 e_\alpha d_j^\alpha\varphi_j\right):\nabla\left(e_\beta\varphi_i\right)dx = 0 \tag{4.6.8}$$

for $i = 1,...,n$.

From this, we obtain a linear system of equations.

$$D^{\mathrm{f}}\underline{d}^{\mathrm{f}} = \underline{f} \tag{4.6.9}$$

with D^{f} containing block diagonal matrices as its entries

$$D_{i,j}^{\mathrm{f}} = \begin{pmatrix} D_{i,j}^{1,1} & 0 & 0 \\ 0 & D_{i,j}^{2,2} & 0 \\ 0 & 0 & D_{i,j}^{3,3} \end{pmatrix}, \tag{4.6.10}$$

$$\underline{d}^{\mathrm{f}} = (d_i^{\mathrm{f}}), \tag{4.6.11}$$

and $\underline{f} \in (\mathbb{R}^3)^n$ including the Dirichlet boundary conditions.

Using definition of $a(\cdot,\cdot)$ in (4.3.2) of Problem 4.8, the entries are given by

$$D_{i,j}^{\beta,\beta} = a(e_\beta\varphi_j, e_\beta\varphi_i) = \int_{\Omega_0^{\mathrm{f}}} \nabla(e_\beta\varphi_j) : \nabla(e_\beta\varphi_i) dx_0 = \sum_{M\in\mathcal{M}_{\mathrm{f}h}^0} \sum_{T\in M} \int_T \sum_{k=1}^{3} \frac{\partial\varphi_i}{\partial x_k}\frac{\partial\varphi_j}{\partial x_k}. \tag{4.6.12}$$

4.6.3 Fluid system assembling

Using *point-wise ordering* for the discrete velocity solution u_h, we have

$$u_h = \sum_{j=1}^{n} u_j\varphi_j, \tag{4.6.13}$$

where $u_j = (u_j^1, u_j^2, u_j^3)^T \in \mathbb{R}^3$ and φ_j is a basis function for each component of space $V_h^{\mathrm{f}}(t)$.

For the discrete pressure solution p_h, we have

$$p_h = \sum_{j=1}^{n} p_j\varphi_j, \tag{4.6.14}$$

where $p_j \in \mathbb{R}$ and φ_j is a basis function for $Q_h^{\mathrm{f}}(t)$.

We choose $v_h^{\mathrm{f}} = e_\beta\varphi_i$ in the first line in (4.5.63) in Problem 4.23 for $\beta = 1,2,3$ and $q_h^{\mathrm{f}} = \varphi_i$ in the second line in (4.5.63) in Problem 4.23:

$$\begin{cases} \tilde{a}(\sum_{j=1}^{n}\sum_{\alpha=1}^{3} u_j^\alpha e_\alpha\varphi_j, e_\beta\varphi_i) + \tilde{b}_1(e_\beta\varphi_i, \sum_{j=1}^{n} p_j\varphi_j) = \langle \tilde{F}_h^{\mathrm{f},n}, e_\beta\varphi_i\rangle, \\ \quad \tilde{b}_2(\sum_{j=1}^{n}\sum_{\alpha=1}^{3} u_j^\alpha e_\alpha\varphi_j, \varphi_i) - \tilde{c}(\sum_{j=1}^{n} p_j\varphi_j, \varphi_i) = \langle \tilde{G}_h^{\mathrm{f},n}, \varphi_i\rangle \end{cases} \tag{4.6.15}$$

for $i = 1, ..., n$.

From this a linear system of equations arises.

$$\begin{pmatrix} A^{\mathrm{f}} & (B_1^{\mathrm{f}})^T \\ B_2^{\mathrm{f}} & -C^{\mathrm{f}} \end{pmatrix} \begin{pmatrix} \underline{u} \\ \underline{p} \end{pmatrix} = \begin{pmatrix} \underline{f}^{\mathrm{f}} \\ \underline{g}^{\mathrm{f}} \end{pmatrix}, \tag{4.6.16}$$

where A^{f} contains 3×3 block matrices $A_{i,j}^{\beta,\alpha}$ for $\beta,\alpha \in 1,2,3$ as its entries, B_1^{f} and B_2^{f} contain $(B_1^{\mathrm{f}})_{i,j}^{\alpha}$ and $(B_2^{\mathrm{f}})_{i,j}^{\alpha}$ for $\alpha \in 1,2,3$ as their entries, $\underline{u} = (u_i)$, $\underline{p} = (p_i)$, $\underline{f}^{\mathrm{f}} \in (\mathbb{R}^3)^n$ containing $\underline{f}_i^\beta$ for $\beta \in 1,2,3$ as its entries and $\underline{g}^{\mathrm{f}} \in \mathbb{R}^n$.

Using definitions of the bilinear forms and linear forms in Problem 4.23, we obtain the following entries for the system:

$$\begin{aligned}
&A_{i,j}^{\beta,\alpha} = a(e_\alpha\varphi_j, e_\beta\varphi_i) + a_s^{\mathrm{f}}(e_\alpha\varphi_j, e_\beta\varphi_i),\\
&(B_1^{\mathrm{f}})_{i,j}^{\beta} = b(e_\beta\varphi_j, \varphi_i) + b_{1,s}(e_\beta\varphi_j, \varphi_i),\\
&(B_2^{\mathrm{f}})_{i,j}^{\beta} = b(e_\beta\varphi_j, \varphi_i) + b_{2,s}(e_\beta\varphi_j, \varphi_i),\\
&C_{i,j}^{\mathrm{f}} = c_s(\varphi_j, \varphi_i),\\
&\underline{f}_i^{\beta} = \langle F^{\mathrm{f},n+1}, e_\beta\varphi_i\rangle + \langle F^{\mathrm{f},n+1}, e_\beta\varphi_i\rangle_s,\\
&\underline{g}_i = \langle G^{\mathrm{f},n+1}, \varphi_i\rangle_s,
\end{aligned} \tag{4.6.17}$$

where

$$\begin{aligned}
a(e_\alpha\varphi_j, e_\beta\varphi_i) = &\begin{cases} 0 & (\text{if } \alpha \neq \beta)\\ \sum\limits_{M\in\mathcal{M}_{\mathrm{f}h}}\sum\limits_{T\in M}\int_T \left(\dfrac{\rho_{\mathrm{f}}}{\delta t} - \mathrm{div} w_h^{\mathrm{f},n+1}\rho_{\mathrm{f}}\right)\varphi_j\varphi_i dx & (\text{if } \alpha = \beta)\end{cases}\\
&+\begin{cases} 0 & (\text{if } \alpha \neq \beta)\\ \sum\limits_{M\in\mathcal{M}_{\mathrm{f}h}}\sum\limits_{T\in M}\int_T \rho_{\mathrm{f}}(\theta_h^{n+1}\cdot\nabla)\varphi_j\varphi_i dx & (\text{if } \alpha = \beta)\end{cases}\\
&+\begin{cases} \sum\limits_{M\in\mathcal{M}_{\mathrm{f}h}}\sum\limits_{T\in M}\int_T \mu\dfrac{\partial\varphi_j}{\partial x_\beta}\dfrac{\partial\varphi_i}{\partial x_\alpha}dx & (\text{if } \alpha \neq \beta)\\ \sum\limits_{M\in\mathcal{M}_{\mathrm{f}h}}\sum\limits_{T\in M}\int_T \mu(\sum\limits_{l=1}^{3}\dfrac{\partial\varphi_j}{\partial x_l}\dfrac{\partial\varphi_i}{\partial x_l} + \dfrac{\partial\varphi_j}{\partial x_\alpha}\dfrac{\partial\varphi_i}{\partial x_\alpha})dx & (\text{if } \alpha = \beta)\end{cases},
\end{aligned} \tag{4.6.18}$$

$$b(e_\beta\varphi_i, \varphi_j) = = -\sum_{M\in\mathcal{M}_{\mathrm{f}h}}\sum_{T\in M}\int_T \varphi_j\frac{\partial\varphi_i}{\partial x_\beta}dx, \tag{4.6.19}$$

$$\langle F^{\mathrm{f},n+1}, e_\beta\varphi_i\rangle = \sum_{M\in\mathcal{M}_{\mathrm{f}h}}\sum_{T\in M}\frac{1}{\delta t}\delta_T\int_T h_T^2\rho_{\mathrm{f}}\hat{u}_{h,\beta}^n\rho_{\mathrm{f}}(\theta_h^{n+1}\cdot\nabla)\varphi_i dx, \tag{4.6.20}$$

and the corresponding stabilization terms

$$\begin{aligned}
a_s^{\mathrm{f}}(e_\alpha\varphi_j, e_\beta\varphi_i) = &\sum_{M\in\mathcal{M}_{\mathrm{f}h}}\sum_{T\in M}\gamma_T\int_T\frac{\partial\varphi_j}{\partial x_\alpha}\frac{\partial\varphi_i}{\partial x_\beta}dx\\
&+\begin{cases} 0 & (\text{if } \alpha \neq \beta)\\ \sum\limits_{M\in\mathcal{M}_{\mathrm{f}h}}\sum\limits_{T\in M}\delta_T\int_T \rho_{\mathrm{f}}(\theta_h^{n+1}\cdot\nabla)\varphi_j\rho_{\mathrm{f}}(\theta_h^{n+1}\cdot\nabla)\varphi_i dx & (\text{if } \alpha = \beta)\end{cases}\\
&+\begin{cases} 0 & (\text{if } \alpha \neq \beta)\\ \sum\limits_{M\in\mathcal{M}_{\mathrm{f}h}}\sum\limits_{T\in M}\int_T \left(\dfrac{\rho_{\mathrm{f}}^2\delta_T}{\delta t} - \rho_{\mathrm{f}}^2\delta_T\mathrm{div} w_h^{\mathrm{f},n+1}\right)\varphi_j(\theta_h^{n+1}\cdot\nabla)\varphi_i dx & (\text{if } \alpha = \beta)\end{cases},
\end{aligned} \tag{4.6.21}$$

$$b_{1,s}(e_\beta\varphi_i, \varphi_j) = \sum_{M\in\mathcal{M}_{\mathrm{f}h}}\sum_{T\in M}\int_T \delta_T\frac{\partial\varphi_j}{\partial x_\beta}\rho_{\mathrm{f}}(\theta_h^{n+1}\cdot\nabla)\varphi_i dx, \tag{4.6.22}$$

$$
\begin{aligned}
b_{2,s}(e_\beta\varphi_i,\varphi_j) = &\sum_{M\in\mathcal{M}_{\mathrm{f}h}}\sum_{T\in M}\int_T\left(-\frac{\delta_T\rho_{\mathrm{f}}}{\delta t}+\delta_T\rho_{\mathrm{f}}\mathrm{div}w_h^{\mathrm{f},n+1}\right)\varphi_i\cdot\frac{\partial\varphi_j}{\partial x_\beta}dx\\
&-\sum_{M\in\mathcal{M}_{\mathrm{f}h}}\sum_{T\in M}\int_T\delta_T\rho_{\mathrm{f}}(\theta_h^{n+1}\cdot\nabla)\varphi_i\cdot\frac{\partial\varphi_j}{\partial x_\beta}dx,
\end{aligned}
\tag{4.6.23}
$$

$$
c_s(\varphi_j,\varphi_i)=\sum_{M\in\mathcal{M}_{\mathrm{f}h}}\sum_{T\in M}\int_T\delta_T\sum_{l=1}^{3}\frac{\partial\bar{\varphi}_j}{\partial x_l}\frac{\partial\bar{\varphi}_i}{\partial x_l}dx \tag{4.6.24}
$$

$$
\langle F^{\mathrm{f},n+1},e_\beta\varphi_i\rangle_s=\sum_{M\in\mathcal{M}_{\mathrm{f}h}^n}\sum_{T\in M}\int_T\frac{1}{\delta t}\delta_T\rho_{\mathrm{f}}u_{h,\beta}^n\rho_{\mathrm{f}}(\theta_h^n\cdot\nabla)\bar{\varphi}_i dx, \tag{4.6.25}
$$

$$
\langle G^{\mathrm{f},n+1},\varphi_i\rangle_s=-\sum_{M\in\mathcal{M}_{\mathrm{f}h}^n}\sum_{T\in M}\int_T\frac{1}{\delta t}\delta_T\rho_{\mathrm{f}}\sum_{l=1}^{3}u_{h,l}^n\frac{\partial\bar{\varphi}_i}{\partial x_l}dT, \tag{4.6.26}
$$

where $u_{h,\beta}^n$ is the β-th component of the velocity solution u_h^n, and $\bar{\varphi}_i=\varphi_i\circ x_{t^{n+1}}^{\mathrm{f}}\circ\left(x_{t^n}^{\mathrm{f}}\right)^{-1}$.

4.7 Summary

This chapter deals with finite element discretization for the structure and fluid sub-problems on hybrid meshes. We constructed the extended P_1 element on hybrid meshes. This extended P_1 element can be directly applied to the structure sub-problem and the harmonic extension problem. For the fluid sub-problem, in order to overcome the inf-sup instability caused by equal order spaces for both velocity and pressure, we enlarge the velocity space with sufficiently many bubble functions. This method is analyzed by explicitly constructing the Fortin operator on the hybrid mesh. For the implementation, the combined PSPG/SUPG method is used.

After the finite element discretization, algebraic equations arise. One main contribution of the thesis is to solve these equations using algebraic multigrid methods. This will be discussed in Chapter 5.

Chapter 5

Algebraic multigrid methods for saddle point problems

5.1 Introduction

From the proper finite element discretization for the FSI sub-problems in Chapter 4, several (large) linear systems of equations arise:

- for the harmonic extension problem

$$D^{\mathrm{f}}\underline{d}^{\mathrm{f}} = \underline{f}, \tag{5.1.1}$$

- for the structure problem

$$A^{\mathrm{s}}\underline{d}^{\mathrm{s}} = \underline{f}^{\mathrm{s}}, \tag{5.1.2}$$

- for the (linearized) Navier-Stokes problem

$$\begin{pmatrix} A^{\mathrm{f}} & (B_1^{\mathrm{f}})^T \\ B_2^{\mathrm{f}} & -C^{\mathrm{f}} \end{pmatrix} \begin{pmatrix} \underline{u} \\ \underline{p} \end{pmatrix} = \begin{pmatrix} \underline{f}^{\mathrm{f}} \\ \underline{g}^{\mathrm{f}} \end{pmatrix}, \tag{5.1.3}$$

where the matrices D^{f}, A^{s}, A^{f}, B_1^{f}, B_2^{f}, C^{f} are sparse. Problems (5.1.1), (5.1.2), and (5.1.3) can be characterized into two groups:

1. Symmetric and positive definite (SPD) systems from the primal variational formulations: (5.1.1) and (5.1.2),
2. Saddle point problem: (5.1.3),

and will be solved by fast iterative methods.

Let us consider an SPD system in the form of

$$Ax = b. \tag{5.1.4}$$

One standard Krylov subspace method to solve this linear system is the (preconditioned) conjugate gradient method ((P)CG). As is well known when applying this method without any preconditioning the iteration numbers required for obtaining a fixed relative accuracy

$\varepsilon \in (0,1)$ are proportional to $\sqrt{\kappa(A)}$, where $\kappa(A)$ is the condition number of the stiffness matrix A and $\kappa(A) = \mathcal{O}(h^{-2})$ for a second order elliptic problem. Here h denotes the mesh size. Using a preconditioning technique, we solve an equivalent system of equations

$$\hat{A}^{-1}Ax = \hat{A}^{-1}b, \tag{5.1.5}$$

for which the SPD matrix $\hat{A}$ is a preconditioner aiming at reducing the condition number of the preconditioned system matrix $\hat{A}^{-1}A$. If the preconditioner $\hat{A}$ is optimal, i.e. $\kappa(\hat{A}^{-1}A) = \mathcal{O}(1)$ and the number of arithmetical operations for realizing $\hat{A}^{-1}$ (only the action of applying $\hat{A}^{-1}$ to a given vector) is proportional to the number of unknowns n, then for a relative error reduction of the residual by a factor ϵ, the number of arithmetical operations is proportional to n. We refer to [12, 74, 111, 139, 121] for details.

A good preconditioner for A in (5.1.4) is to apply a multigrid method (MG) to the system (5.1.4). This method can significantly reduce the error by combining the so-called smoothing steps and coarse grid corrections. In order to apply this method, we need to build a hierarchy of linear systems on different levels. According to the way how the hierarchy is constructed, two types of this method have been developed. If a hierarchy of meshes is available, the construction of these operators is easily obtained by using geometric information of two consecutive meshes. Or instead, the coarse system matrices are directly built from the discretization of differential operators on all level. This method is the so-called geometric multigrid method (GMG). For detailed discussions concerning its convergence analysis and applications, we refer to [73, 29]. If either only the matrix or the finest mesh information is available, the matrix hierarchy will not be built on the hierarchy of meshes, but on purely algebraic information. Thus it leads to the algebraic multigrid methods (AMG), see [31, 145, 146, 132, 71, 153, 69, 91, 135, 154, 164, 24]. It is usually more robust and efficient when combining MG preconditioners and Krylov subspaces accelerations, see [84, 85].

In our situations, due to the mesh generator used and complex structures of hybrid meshes, a hierarchy of such meshes is not available. Thus the AMG combined with a CG acceleration for solving (5.1.4) will be applied to the discrete structure sub-problem and harmonic extension problem in the FSI simulation.

Let us denote the saddle point problem arising from the discretization of fluid sub-problem by

$$\begin{pmatrix} A & {B_1}^T \\ B_2 & -C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}. \tag{5.1.6}$$

Note that for the Stokes problem, the block A is SPD, $B_1 = B_2 = B$ have full rank, C is symmetric and positive semi-definite (SPSD), and the (negative) Schur complement

$$S = B_2 A^{-1} B_1^T + C \tag{5.1.7}$$

is SPD. For the Oseen problem (linearized Navier-Stokes problem), A is unsymmetric and in general $B_1 \neq B_2$. However, for simplicity, we will not deal with all these situations in the remaining of this chapter, but restrict to the case that $B_1 = B_2$.

One method to solve (5.1.6) is a (preconditioned) Krylov subspace method which is capable of solving indefinite and unsymmetric problems: e.g. the generalized minimal residual method (GMRES) (see [140]) or the stabilized bi-conjugate gradient method (BiCGstab) (see [45]). Another method is obtained by iteratively decoupling the velocity and pressure unknowns, which relies on good preconditioners $\hat{A}^{-1}$ for the block A, and $\hat{S}^{-1}$ for the modified

Schur complement $\tilde{S} = B[\text{diag}(A)]^{-1}B^T + C$. The semi-implicit method for pressure-linked equations (SIMPLE) (see [120, 119]) and inexact Uzawa methods (see [177, 178, 9, 35, 142]) are among the most popular versions of these types. Another method stems from previous contributions of Markus Wabro (see [161, 163, 162]), where an all-at-once AMG method for the coupled system was developed.

For an overview of numerical methods for saddle point problems, we refer to [21].

In the next, we will firstly describe ingredients of a standard AMG. The extension to the saddle point problem needs some considerations, e.g., how to algebraically construct a stabilized P_1-P_1 hierarchy such that the discrete inf-sup conditions are fulfilled on all levels. This point has been addressed in [161, 163, 162] for the P_1-P_1 element and other elements. We will extend the stabilized P_1-P_1 hierarchy result known for tetrahedral meshes to hybrid meshes and apply the AMG to the fluid sub-problem in the FSI simulation.

5.2 A general AMG framework

First of all, we describe the components of a general AMG method for the system of linear equations arising from finite element discretization of primal variational formulations:

$$A_1 x = b_1, \tag{5.2.1}$$

where A_1 is an SPD $n_1 \times n_1$ matrix arising from the discretization of second order elliptic partial differential operators.

The index $1 \leq l \leq L$ will be used to indicate the levels of a hierarchy, i.e. index 1 refers to the finest level and L to the coarsest level. Note that in GMG the numbering is usually reversed.

5.2.1 AMG ingredients

First of all, we need to define a full rank prolongation matrix

$$P_{l+1}^l : \mathbb{R}^{n_{l+1}} \to \mathbb{R}^{n_l} \tag{5.2.2}$$

by using some coarsening strategies based on the property of the matrix A_l, where $n_{l+1} < n_l$ and $l = 1, ..., L-1$.

We also need a restriction matrix

$$R_l^{l+1} : \mathbb{R}^{n_l} \to \mathbb{R}^{n_{l+1}} \tag{5.2.3}$$

which is defined by $R_l^{l+1} = \left(P_{l+1}^l\right)^T$.

Then the matrix on the next coarse level $l+1$ can be built by the Galerkin projection

$$A_{l+1} = R_l^{l+1} A_l P_{l+1}^l. \tag{5.2.4}$$

Additionally, we need on each level l an iterative method (e.g., Gauss-Seidel or damped ω-Jacobia) of the form

$$x_l^{k+1} = \mathcal{S}_l(x_l^k, b_l), \tag{5.2.5}$$

called the smoother, for the problem $A_l x_l = b_l$, where k is the iteration index.

The problem

$$A_L x_L = b_L \tag{5.2.6}$$

on the coarsest level L is solved exactly by a direct solver or approximately by an iterative solver.

5.2.2 An AMG algorithm

We now describe the AMG algorithm (see Algorithm 5.2.1), where m_{pre} and m_{post} are the number of presmoothing steps (steps 1-3) and postsmoothing steps (steps 14-16), respectively. The steps 4-13 are referred to as "coarse grid correction". The basic AMG method is

Algorithm 5.2.1 AMG(A_l, x_l, b_l)

1: **for** $k = 1$ to m_{pre} **do**
2: $x_l = \mathcal{S}_l(x_l, b_l)$,
3: **end for**
4: $b_{l+1} = R_l^{l+1}(b_l - A_l x_l)$,
5: **if** $l + 1 = L$ **then**
6: Solve $A_L x_L = b_L$,
7: **else**
8: $x_{l+1} = 0$,
9: **for** $k = 1, .., \mu$ **do**
10: x_{l+1} =AMG(A_{l+1}, x_{l+1}, b_{l+1}),
11: **end for**
12: **end if**
13: $x_l = x_l + P_{l+1}^l x_{l+1}$,
14: **for** $k = 1$ to m_{post} **do**
15: $x_l = \mathcal{S}_l(x_l, b_l)$,
16: **end for**
17: return x_l.

realized by repeated application of this algorithm until it fulfills a certain stopping criteria. For $\mu = 1$ and $\mu = 2$, the iterations of Algorithm 5.2.1 are called V-cycle and W-cycle, respectively.

5.2.3 Basic convergence analysis

In the GMG case, the W-cycle convergence is shown by proving the two-grid convergence. The iteration operator for the two grid method (without postsmoothing: $m_{\text{pre}} = m$, $m_{\text{post}} = 0$) is given by

$$\mathcal{M}_l^{l+1} := \left(I - P_{l+1}^l A_{l+1}^{-1} R_l^{l+1} A_l\right) S_l^m, \tag{5.2.7}$$

where m is the number of presmoothing steps and S_l the iteration matrix for a smoothing step. Various convergence analyses were given in [14, 11, 92, 96, 50, 72]. Here we make use of an idea employed in [26, 73], where a product splitting

$$\mathcal{M}_l^{l+1} = \left[A_l^{-1} - P_{l+1}^l A_{l+1}^{-1} R_l^{l+1}\right] [A_l S_l^m] \tag{5.2.8}$$

is applied to the iteration operator $\mathcal{M}_l^{l+1}$. Once the *approximation property*

$$\left\| A_l^{-1} - P_{l+1}^l A_{l+1}^{-1} R_l^{l+1} \right\|_{l_2} \leq \frac{c}{\|A_l\|_{l_2}} \tag{5.2.9}$$

with a positive constant c and the *smoothing property*

$$\|A_l S_l^m\|_{l_2} \leq \eta(m) \|A_l\|_{l_2} \tag{5.2.10}$$

with the *smoothing rate* $\eta(m) \to 0$ as $m \to \infty$ are established, the two-grid convergence immediately follows from these two properties.

Note, it has been shown in [26] that the V-cycle converges uniformly with respect to the number of levels under an assumption of fully elliptic regularity.

The convergence analysis for the AMG is mainly taken from [135, 110] and is mostly restricted to an SPD problem. Let $T_l := I - P_{l+1}^l A_{l+1}^{-1} R_l^{l+1} A_l$ be the coarse grid correction operator. If the *smoothing property*

$$\|S_l e\|_{A_l}^2 \leq \|e\|_{A_l}^2 - \alpha \|A_l e\|_{D_l^{-1}}^2 \tag{5.2.11}$$

and if additionally the *approximation property*

$$\|T_l e\|_{A_l}^2 \leq \beta \|A_l T_l e\|_{D_l^{-1}}^2 \tag{5.2.12}$$

hold for all e, where α and β are constants independent of e, and D_l is the diagonal of A_l, then the two-grid algorithm with one postsmoothing step converges, i.e.

$$\|S_l T_l e\|_{A_l}^2 \leq \left(1 - \frac{\alpha}{\beta}\right) \|e\|_{A_l}^2 . \tag{5.2.13}$$

5.3 An AMG based on coarse (C) and fine (F) nodes splitting

Assume that the finite element discretization is based on nodal degrees of freedom. The classical AMG introduced for the scalar elliptic problems in [145, 32, 135] are based on nodal splittings into sets of C-nodes which will also be used on the coarse level, and F-nodes which are only used on the fine level. This method has been extended to systems of PDEs in a natural blockwise fashion, e.g., see its application to the linear elasticity problem in [69], which is used for constructing the AMG solver for the structure sub-problem in the FSI simulation.

Here we will only present a summary on how to make a C-node selection and how to construct prolongations. For a complete description of these methods, we refer to [145, 32, 135, 69]. For other types of AMG methods using different coarsening strategies and prolongation construction, we refer to [34, 83, 77, 93, 94, 71].

5.3.1 Coarse node selection

We use a very easy algorithm (see Algorithm 5.3.1) introduced in [90] to implement the coarse node selection (C/F nodes splitting). It will be applied to the AMG solvers for the FSI sub-problems. According to the concept of *strong couplings* between unknowns, there

Algorithm 5.3.1 Red-black coloring

1: Choose an uncolored node,
2: This node is colored black (C-nodes),
3: All uncolored neighbors are colored red (F-nodes),
4: If not all nodes are colored, go to step 1.

are various ways to define the neighbors in Algorithm 5.3.1, e.g., for an M-matrix or nearly M-matrix (*essentially positive type*), see [145, 135], and for the matrix arising from linear elasticity problem, see [69]. For completeness, the definition of an M-matrix and *essentially positive type* matrix is given in the Appendix 5.8.

5.3.2 Construction of prolongations

The C/F nodes splitting introduces a corresponding grouping of the unknowns: the F-unknowns (living only on the F-nodes) and the C-unknowns (living on the C-nodes), say $(x_{\mathrm{F}}^l, x_{\mathrm{C}}^l)$, which leads to a partitioned structure of a linear system on a level l (see [145]):

$$\begin{pmatrix} A_{\mathrm{FF}}^l & A_{\mathrm{FC}}^l \\ A_{\mathrm{CF}}^l & A_{\mathrm{CC}}^l \end{pmatrix} \begin{pmatrix} x_{\mathrm{F}}^l \\ x_{\mathrm{C}}^l \end{pmatrix} = \begin{pmatrix} b_{\mathrm{F}}^l \\ b_{\mathrm{C}}^l \end{pmatrix}. \tag{5.3.1}$$

A natural way to construct the prolongation is to leave C-unknowns unchanged, and properly interpolate for F-unknowns, i.e.

$$P_{l+1}^l = \begin{pmatrix} P_{\mathrm{C}}^{\mathrm{F}} \\ I \end{pmatrix}, \tag{5.3.2}$$

where $P_{\mathrm{C}}^{\mathrm{F}}$ is an interpolation operator from C-unknowns to F-unknowns. There are various ways to define $P_{\mathrm{C}}^{\mathrm{F}}$, e.g., a simple version we used in the implementation is taking an average for an F-unknown over all connected C-unknowns:

$$(P_{\mathrm{C}}^{\mathrm{F}})_{i,j} = \begin{cases} \frac{1}{m_i} & \text{if } j \text{ is a neighboring C-node of an F-node } i, \\ 0 & \text{otherwise,} \end{cases} \tag{5.3.3}$$

where m_i is the number of neighbouring C-nodes of an F-node i, see [90, 88].

For details concerning how to construct prolongations for an M-matrix or *essentially positive type* matrix, we refer to [110, 145, 146], and for linear elasticity system, we refer to [69].

5.4 Coarsening strategy for the saddle point problem

If we apply the coarsening strategy to the saddle point problem (5.1.6) in a straightforward way, it will lead to a mixture of velocity and pressure unknowns on coarse levels. Therefore, in [161, 162, 163], it is suggested that the velocity and pressure unknowns are coarsened separately. Thus a prolongation operator is chosen as follows

$$P_{l+1}^l = \begin{pmatrix} I_{l+1}^l & \\ & J_{l+1}^l \end{pmatrix}, \tag{5.4.1}$$

where

$$J_{l+1}^l : \mathbb{R}^{m_{l+1}} \to \mathbb{R}^{m_l} \tag{5.4.2}$$

is the prolongation matrix of (5.3.2) for a scalar problem, and the prolongation matrix

$$I_{l+1}^l : (\mathbb{R}^3)^{n_{l+1}} \to (\mathbb{R}^3)^{n_l} \tag{5.4.3}$$

is defined as $\begin{pmatrix} I_{\mathrm{C}}^{\mathrm{F}} \\ \hat{I} \end{pmatrix}$ with

$$(I_{\mathrm{C}}^{\mathrm{F}})_{i,j} = \begin{cases} \frac{1}{m_i}\bar{I} & \text{if } j \text{ is a neighboring C-node of an F-node } i, \\ 0\bar{I} & \text{otherwise,} \end{cases} \tag{5.4.4}$$

a $3n_{l+1} \times 3n_{l+1}$ identity matrix $\hat{I}$, and a 3×3 identity matrix $\bar{I}$.

The corresponding restriction matrix is given by

$$R_l^{l+1} = \left(P_{l+1}^l\right)^T. \tag{5.4.5}$$

The system matrix on the level $l+1$ is given by

$$\mathcal{K}_{l+1} = \begin{pmatrix} A_{l+1} & B_{l+1}^T \\ B_{l+1} & -C_{l+1} \end{pmatrix} \tag{5.4.6}$$

with

$$\begin{aligned} A_{l+1} &= I_l^{l+1} A_l I_{l+1}^l, \quad B_{l+1}^T = I_l^{l+1} B_l^T J_{l+1}^l, \\ B_{l+1} &= J_l^{l+1} B_l I_{l+1}^l, \quad C_{l+1} = J_l^{l+1} C_l J_{l+1}^l \end{aligned} \tag{5.4.7}$$

and

$$I_l^{l+1} = \left(I_{l+1}^l\right)^T \text{ and } J_l^{l+1} = \left(J_{l+1}^l\right)^T.$$

However this direct coarsening strategy does not necessarily lead to a stabilized system on the coarse level since the stability is not guaranteed. In the next sections 5.5 and 5.6, we will discuss the stability issue on the coarse levels.

5.5 The stabilized P_1-P_1 hierarchy on hybrid meshes

As we discussed in Section 4.6.3 for the fluid assembling process, we represent $q_h \in Q_h$ by

$$q_h = \sum_{j=1}^{m} q_j \varphi_j, \tag{5.5.1}$$

where $q_j \in \mathbb{R}$ and $\varphi_j, j = 1, ..., m$ are basis functions of Q_h.

The coefficient vector for the pressure function representation is given by

$$\underline{q} = (q_j) \in \mathbb{R}^m =: \underline{Q}_h. \tag{5.5.2}$$

In a similar way, we represent v_h by

$$v_h = \sum_{j=1}^{n} v_j \varphi_j, \tag{5.5.3}$$

where $v_j \in \mathbb{R}^3$ and $\varphi_j, j = 1, ..., n$ are basis functions for each component of V_h.

The coefficient vector for the velocity function representation is given by

$$\underline{v} = (v_j) \in \left(\mathbb{R}^3\right)^n =: \underline{V}_h. \tag{5.5.4}$$

So, by (5.5.1)-(5.5.4), to each vector $\underline{v} \in \underline{V}_h$ and $\underline{q} \in \underline{Q}_h$, there exists a unique finite element function $v_h \in V_h$ and $q_h \in Q_h$, and vice versa. This determines two *FE-isomorphisms* between the spaces of coefficient vectors and the spaces of finite element functions:

$$\phi_V : \underline{V}_h \to V_h, \ \phi_Q : \underline{Q}_h \to Q_h. \tag{5.5.5}$$

In an analogous way, for a coarse level l, i.e. $l \in \{2, ..., L\}$, the spaces of the coefficient vectors are denoted by

$$\underline{V}_l := \left(\mathbb{R}^3\right)^{n_l}, \ \underline{Q}_l := \mathbb{R}^{m_l}, \tag{5.5.6}$$

where n_l and m_l are the numbers of unknowns for the velocity and pressure on the level l, respectively.

The finite element spaces on coarse levels l are given by

$$\begin{aligned} V_l &= \{v \in V_h : \exists \underline{v}_l \in \underline{V}_l \text{ such that } \underline{v} = I_2^1 I_3^2 ... I_l^{l-1} \underline{v}_l\}, \\ Q_l &= \{p \in Q_h : \exists \underline{q}_l \in \underline{Q}_l \text{ such that } \underline{p} = J_2^1 J_3^2 ... J_l^{l-1} \underline{q}_l\}, \end{aligned} \tag{5.5.7}$$

i.e. any functions from the spaces V_l and Q_l have unique representations in the spaces V_h and Q_h by properly chosen interpolation operators.

Then, to each vector $\underline{v}_l \in \underline{V}_l$ and $\underline{q}_l \in \underline{Q}_l$, there exists a unique finite element function $v_l \in V_l$ and $q_l \in Q_l$, and vice versa. This determines two *FE-AMG-isomorphisms* which associate the spaces of coefficient vectors and the spaces of finite element functions to each other:

$$\phi_V^l : \underline{V}_l \to V_l, \ \phi_Q^l : \underline{Q}_l \to Q_l. \tag{5.5.8}$$

We will show on each level a discrete inf-sup condition of the form

$$\sup_{(0,0) \neq (\underline{v},\underline{q}) \in \underline{V}_l \times \underline{Q}_l} \frac{\mathcal{B}_l\left(\underline{u}, \underline{p}; \underline{v}, \underline{q}\right)}{\|\underline{v}\|_{A_l} + \|\underline{q}\|_{M_l}} \geq \zeta_l \left(\|\underline{u}\|_{A_l} + \|\underline{p}\|_{M_l}\right) \tag{5.5.9}$$

for all $\left(\underline{u}, \underline{p}\right) \in \underline{V}_l \times \underline{Q}_l$, where

$$\mathcal{B}_l\left(\underline{u}, \underline{p}; \underline{v}, \underline{q}\right) = \underline{u}^T A_l \underline{v} + \underline{p}^T B_l \underline{v} + \underline{u}^T B_l^T \underline{q} - \underline{p} C_l \underline{q}, \tag{5.5.10}$$

$\zeta_l > 0$ may depend on l, and M_l is the Galerkin projection of the mass matrix M_1 to level l.

According to [161], we define the coarse level matrix C_{l+1} with a stabilization scaling in Algorithm 5.5.1, where D_l is the diagonal of the Galerkin projection of A_l, h is the

Algorithm 5.5.1 Constructure of C_l

$\tilde{C}_1 = C_1$,

1: **for** $l \geq 1$ **do**

2: $\quad \tilde{C}_{l+1} = J_l^{l+1} \tilde{C}_l J_{l+1}^l, \ C_{l+1} = \frac{\lambda_{\max}\left(D_l^{-1} M_l\right)}{h^2} \tilde{C}_{l+1}$,

3: **end for**

discretization parameter on the finest level 1.

The proof follows the ideas given in [157, 61, 60, 161]. For the final result (see Theorem 5.3), we need some preliminary results (see Lemma 5.1 and Lemma 5.2).

5.5.1 An inequality on the finest level

First of all, we prove an inequality on the hybrid mesh, which is related to the argument of Verfüth's trick in [157].

Recall the spaces $V_h = [V_h^H]^3 \cap [C_0(\overline{\Omega})]^3$ and $Q_h = V_h^H \cap L_0^2(\Omega)$ defined in (4.5.11) and (4.5.12) for the velocity and pressure, respectively.

In order to make the proof shorter, we use some abbreviations for the norms: $\|\cdot\|_0$ denotes $\|\cdot\|_{L^2(\Omega)}$, $\|\cdot\|_{0,T}$ denotes $\|\cdot\|_{L^2(T)}$ and $\|\cdot\|_1$ denotes $\|\cdot\|_{H^1(\Omega)}$.

Lemma 5.1. *Assume that there exists a constant $\underline{\alpha}$ such that for all elements $T \subset M \in \mathcal{M}_h$ the diameter h_T satisfies*

$$\underline{\alpha} h \leq h_T. \tag{5.5.11}$$

Then for the spaces V_h and Q_h, there exist constants c_1 and c_2 such that

$$\sup_{0 \neq v \in V_h} \frac{(\nabla \cdot v, p)}{\|v\|_1} \geq c_1 \|p\|_0 - c_2 |p|_h, \tag{5.5.12}$$

for all $p \in Q_h$, where $|\cdot|_h$ denotes the seminorm

$$|q|_h = \left(\sum_{M \in \mathcal{M}_h} \sum_{T \in M} \sigma(T) \|\nabla q\|_{0,T}^2 \right)^{1/2} \tag{5.5.13}$$

with a mesh dependent parameter $\sigma(T) = O(h_T^2)$ defined in (4.5.9).

Proof. Since $p \in L_0^2(\Omega)$, then there exists a non-trivial $w \in [H_0^1(\Omega)]^3$ such that

$$(\nabla \cdot w, p) \geq c_3 \|p\|_0 \|w\|_1 \tag{5.5.14}$$

with a constant c_3.

Taking $\tilde{w} = I_C w \in V_h \cap [H_0^1(\Omega)]^3$, where the Clément operator I_C is defined in (4.2.26) for w, and by the estimates in Lemma 4.4, it is easy to see that

$$\|\tilde{w}\|_1 \leq c_4 \|w\|_1, \tag{5.5.15}$$

and with the additional mesh assumption of (5.5.11), we have

$$\left(\sum_{M \in \mathcal{M}_h} \sum_{T \in M} h_T^{-2} \|w - \tilde{w}\|_{0,T}^2, \right)^{1/2} \leq c_5 \|w\|_1, \tag{5.5.16}$$

where c_4 and c_5 are constants.

Using integration by parts on each $T \subset M \in \mathcal{M}_h$, the above interpolation estimates and inequalities (2.3.1), (2.3.3), we obtain

$$\begin{aligned}
(\nabla \cdot \tilde{w}, p) &= (\nabla \cdot (\tilde{w} - w), p) + (\nabla \cdot w, p) \geq (\nabla \cdot (\tilde{w} - w), p) + c_3 \|w\|_1 \|p\|_0 \\
&= \sum_{M \in \mathcal{M}_h} \sum_{T \in M} (w - \tilde{w}, \nabla p) + c_3 \|w\|_1 \|p\|_0 \\
&\geq - \left(\sum_{M \in \mathcal{M}_h} \sum_{T \in M} h_T^{-2} \|w - \tilde{w}\|_{0,T}^2 \right)^{1/2} \left(\sum_{M \in \mathcal{M}_h} \sum_{T \in M} h_T^2 \|\nabla p\|_{0,T}^2 \right)^{1/2} \\
&\quad + c_3 \|w\|_1 \|p\|_0 \\
&\geq \left(-c_5 \left(\sum_{M \in \mathcal{M}_h} \sum_{T \in M} h_T^2 \|\nabla p\|_{0,T}^2 \right)^{1/2} + c_3 \|p\|_0 \right) \|w\|_1 \\
&\geq \left(-\frac{c_5}{c_4} \left(\sum_{M \in \mathcal{M}_h} \sum_{T \in M} h_T^2 \|\nabla p\|_{0,T}^2 \right)^{1/2} + \frac{c_3}{c_4} \|p\|_0 \right) \|\tilde{w}\|_1.
\end{aligned} \tag{5.5.17}$$

Together with

$$\left(\sum_{M \in \mathcal{M}_h} \sum_{T \in M} h_T^2 \|\nabla p\|_{0,T}^2 \right)^{1/2} \leq c_6 |p|_h, \tag{5.5.18}$$

where c_6 is constant, we obtain

$$\frac{(\nabla \cdot \tilde{w}, p)}{\|\tilde{w}\|_1} \geq c_7 \|p\|_0 - c_8 |p|_h \tag{5.5.19}$$

with constants c_7 and c_8.

Finally it is clear that for $0 \neq z \in V_h$, we have

$$\left| \frac{(\nabla \cdot z, p)}{\|z\|_1} \right| \leq \frac{\|z\|_1 \|p\|_0}{\|z\|_1} = \|p\|_0. \tag{5.5.20}$$

We choose a fixed $0 \neq z \in V_h$ and let $v = \|\tilde{w}\|_1^{-1} \tilde{w} + \delta \|z\|_1^{-1} z$ and $\delta > 0$. Then if $\delta < c_7$, we have

$$(\nabla \cdot v, p) = \frac{(\nabla \cdot \tilde{w}, p)}{\|\tilde{w}\|_1} + \delta \frac{(\nabla \cdot z, p)}{\|z\|_1} \geq c_7 \|p\|_0 - c_8 |p|_h - \delta \|p\|_0 \geq c_9 \|p\|_0 - c_8 |p|_h, \tag{5.5.21}$$

where $c_9 = c_7 - \delta > 0$. Since $\|v\|_1 \leq 1 + \delta$, it follows that

$$\frac{(\nabla \cdot v, p)}{\|v\|_1} \geq \frac{1}{1+\delta} \left(c_9 \|p\|_0 - c_8 |p|_h \right) = c_1 \|p\|_0 - c_2 |p|_h \tag{5.5.22}$$

with $c_1 = \frac{c_9}{1+\delta}$ and $c_2 = \frac{c_8}{1+\delta}$. Thus the claimed result follows. □

We now prove similar inequalities on all levels.

5.5.2 Inequalities on all levels

It should be mentioned that the result for pure tetrahedral meshes has been shown in [161]. We extend it to hybrid meshes.

Lemma 5.2. *Assume the same assumption on the meshes as in Lemma 5.1. Moreover, we assume that A_l is symmetric and essentially positive type (see Appendix 5.8) and that for all $v_l \in V_l$, there always exists a $\underline{\Pi}_l^{l+1} : \underline{V}_l \to \underline{V}_{l+1}$ such that*

$$\|\underline{v}_l - I_{l+1}^l \underline{\Pi}_l^{l+1} \underline{v}_l\|_{D_l}^2 \leq \beta_1 \|\underline{v}_l\|_{A_l}^2 \tag{5.5.23}$$

for $\underline{v}_l \in \underline{V}_l$ with a positive constant β_1. Here D_l denotes the diagonal of A_l.

Then there exist positive constants c_l and d_l such that for all levels $l \in \{1, ..., L\}$

$$\sup_{0 \neq \underline{v} \in \underline{V}_l} \frac{\underline{v} B_l^T \underline{p}}{\|\underline{v}\|_{A_l}} \geq c_l \|\underline{p}\|_{M_l} - d_l \left(\underline{p}^T C_l \underline{p} \right)^{1/2} \tag{5.5.24}$$

for all $\underline{p} \in \underline{Q}_l$, where $l = 1$ is the finest level.

Proof. It is easy to see

$$\|\underline{x}_l\|_{M_l}^2 \leq \lambda_{\max}\left(D_l^{-1}M_l\right)\|\underline{x}_l\|_{D_l}^2 \tag{5.5.25}$$

for all $\underline{x}_l \in \underline{V}_l$.

Hence combined with (5.5.23), we have

$$\|\underline{v}_l - I_{l+1}^l\underline{\Pi}_l^{l+1}\underline{v}_l\|_{M_l}^2 \leq \lambda_{\max}\left(D_l^{-1}M_l\right)\beta_1\|\underline{v}_l\|_{A_l}^2. \tag{5.5.26}$$

Therefore

$$\|v_l - \Pi_l^{l+1}v_l\|_0^2 \leq \lambda_{\max}\left(D_l^{-1}M_l\right)\beta_1\|v_l\|_1^2. \tag{5.5.27}$$

Using the property (5.8.3) of *essentially positive type* matrices, it follows that

$$\|I_{l+1}^l\underline{\Pi}_l^{l+1}\underline{v}_l\|_{A_l} - \|\underline{v}_l\|_{A_l} \leq \sqrt{\frac{2}{c}}\|I_{l+1}^l\underline{\Pi}_l^{l+1}\underline{v}_l - \underline{v}_l\|_{D_l} \leq \sqrt{\frac{2\beta_1}{c}}\|\underline{v}_l\|_{A_l}. \tag{5.5.28}$$

So

$$\|I_{l+1}^l\underline{\Pi}_l^{l+1}\underline{v}_l\|_{A_l} \leq \underbrace{\left(1+\sqrt{\frac{2\beta_1}{c}}\right)}_{=:\beta_2}\|\underline{v}_l\|_{A_l}. \tag{5.5.29}$$

We will prove (5.5.24) by induction. For the finest level, we have (5.5.12) from Lemma 5.1, i.e. (5.5.24) holds for $l = 1$. Assume for a level $l \geq 1$ there exists constants c_l and d_l such that (5.5.24) holds, i.e. there exists a $w_l \in V_l$ such that

$$(\nabla\cdot w_l, p_l) \geq c_l\|w_l\|_1\|p_l\|_0 - d_l\|w_l\|_1\left(\underline{p}_l^T C_l\underline{p}_l\right)^{1/2} \tag{5.5.30}$$

for all $p_l \in Q_l$.

Set $w_{l+1} = \Pi_l^{l+1}w_l$, then by integration by parts over each $T \subset M$ and using inequalities (2.3.1), (2.3.3),

$$\begin{aligned}
(\nabla\cdot w_{l+1}, p_{l+1}) &= \left(\nabla\cdot\left(w_{l+1}-w_l\right), p_{l+1}\right) + (\nabla\cdot w_l, p_{l+1}) \\
&= \sum_{M\in\mathcal{M}_h}\sum_{T\in M}(w_l - w_{l+1}, \nabla p_{l+1}) + (\nabla\cdot w_l, p_{l+1}) \\
&\geq -\left(\sum_{M\in\mathcal{M}_h}\sum_{T\in M}h_T^{-2}\|w_l - w_{l+1}\|_{0,T}^2\right)^{1/2}\left(\sum_{M\in\mathcal{M}_h}\sum_{T\in M}h_T^2\|\nabla p_{l+1}\|_{0,T}^2\right)^{1/2} \\
&\quad + (\nabla\cdot w_l, p_{l+1}).
\end{aligned} \tag{5.5.31}$$

Now by using (5.5.11) and (5.5.27), we have

$$\sum_{M\in\mathcal{M}_h}\sum_{T\in M}h_T^{-2}\|w_l - w_{l+1}\|_{0,T}^2 \leq (\underline{\alpha}h)^{-2}\|w_l - w_{l+1}\|_0^2 \leq \frac{\beta_1}{\underline{\alpha}^2}\frac{\lambda_{\max}\left(D_l^{-1}M_l\right)}{h^2}\|w_l\|_1^2. \tag{5.5.32}$$

Using the constructrure of C_l in Algorithm 5.5.1, one obtains

$$\frac{\underline{p}_{l+1}^T C_{l+1}\underline{p}_{l+1}}{\underline{p}_{l+1}^T J_l^{l+1}C_l J_{l+1}^l\underline{p}_{l+1}} = \sqrt{\frac{\lambda_{\max}\left(D_l^{-1}M_l\right)}{\lambda_{\max}\left(D_{l-1}^{-1}M_{l-1}\right)}} =: \xi_l \tag{5.5.33}$$

and

$$\sum_{M\in\mathcal{M}_h}\sum_{T\in M} h_T^2\|\nabla p_{l+1}\|_{0,T}^2 = \underline{p}_{l+1}^T\tilde{C}_{l+1}\underline{p}_{l+1} = \frac{h^2}{\lambda_{\max}\left(D_l^{-1}M_l\right)}\underline{p}_{l+1}^T C_{l+1}\underline{p}_{l+1}. \tag{5.5.34}$$

Using (5.5.30) and (5.5.33), we have

$$\begin{aligned}(\nabla\cdot w_l, p_{l+1}) &\geq c_l\|w_l\|_1\|p_{l+1}\|_0 - d_l\|w_l\|_1\left(\underline{p}_{l+1}^T J_l^{l+1}C_l J_{l+1}^l\underline{p}_{l+1}\right)^{1/2}\\ &= c_l\|w_l\|_1\|p_{l+1}\|_0 - d_l\xi_l\|w\|_1\left(\underline{p}_{l+1}^T C_{l+1}\underline{p}_{l+1}\right)^{1/2}.\end{aligned} \tag{5.5.35}$$

It then follows that

$$\begin{aligned}(\nabla\cdot w_{l+1}, p_{l+1}) &\geq -\frac{\sqrt{\beta_1}}{\underline{\alpha}}\|w_l\|_1\left(\underline{p}_{l+1}^T C_{l+1}\underline{p}_{l+1}\right)^{1/2} + (\nabla\cdot w_l, p_{l+1})\\ &\geq -\left(\frac{\sqrt{\beta_1}}{\underline{\alpha}} + d_l\xi_l\right)\|w_l\|_1\left(\underline{p}_{l+1}^T C_{l+1}\underline{p}_{l+1}\right)^{1/2} + c_l\|p_{l+1}\|_0\|w_l\|_1.\end{aligned} \tag{5.5.36}$$

Using (5.5.29), we obtain

$$\frac{(\nabla\cdot w_{l+1}, p_{l+1})}{\|w_{l+1}\|_1} \geq \frac{1}{\beta_2}\frac{(\nabla\cdot w_{l+1}, p_{l+1})}{\|w_l\|_1} \geq c_{l+1}\|p_{l+1}\|_0 - d_{l+1}\left(\underline{p}_{l+1}^T C_{l+1}\underline{p}_{l+1}\right)^{1/2} \tag{5.5.37}$$

with parameters

$$\begin{aligned}c_{l+1} &:= \frac{c_l}{\beta_2},\\ d_{l+1} &:= \frac{\sqrt{\beta_1}}{\beta_2\underline{\alpha}} + \frac{d_l\xi_l}{\beta_2}.\end{aligned} \tag{5.5.38}$$

Thus the claimed results follows. □

5.5.3 The stabilized hierarchy on hybrid meshes

We are now ready for showing the stability for the stabilized P_1-P_1 hierarchy following ideas in [161, 61, 157]. For completeness, we will present the proof.

Theorem 5.3. *Suppose the assumptions of Lemma 5.2 hold. Then we have the stability on all levels l*

$$\sup_{(0,0)\neq(\underline{v},\underline{q})\in\underline{V}_l\times\underline{Q}_l}\frac{\mathcal{B}_l\left(\underline{u},\underline{p};\underline{v},\underline{q}\right)}{\|\underline{v}\|_{A_l}+\|\underline{q}\|_{M_l}} \geq \zeta_l\left(\|\underline{u}\|_{A_l}+\|\underline{p}\|_{M_l}\right) \tag{5.5.39}$$

for all $\left(\underline{u},\underline{p}\right)\in\underline{V}_l\times\underline{Q}_l$ with some $\zeta_l>0$.

Proof. Let $\underline{w}\in\underline{V}_l$ be chosen such that the supremum of $\underline{v}B_l^T\underline{p}/\|\underline{v}\|_{A_l}$ for all $\underline{p}\in\underline{Q}_l$ in (5.5.24) is attained and that $\|\underline{w}\|_{A_l} = \|\underline{p}\|_{M_l}$.

Using Lemma 5.2 and the arithmetic-geometric-mean inequality (2.3.4), we obtain

$$\begin{aligned}\mathcal{B}_l\left(\underline{u},\underline{p};\underline{w},0\right) =& \underline{u}^T A_l\underline{w} + \underline{p}^T B_l\underline{w}\\ \geq& -\|\underline{u}\|_{A_l}\|\underline{w}\|_{A_l} + c_l\|\underline{w}\|_{A_l}\|\underline{p}\|_{M_l} - d_l\|\underline{w}\|_{A_l}\left(\underline{p}^T C_l\underline{p}\right)^{1/2}\\ =& -\|\underline{u}\|_{A_l}\|\underline{p}\|_{M_l} + c_l\|\underline{p}\|_{M_l}^2 - d_l\|\underline{p}\|_{M_l}\left(\underline{p}^T C_l\underline{p}\right)^{1/2}\\ \geq& -\frac{1}{2\epsilon}\|\underline{u}\|_{A_l}^2 - \frac{\epsilon}{2}\|\underline{p}\|_{M_l}^2 + c_l\|\underline{p}\|_{M_l}^2 - \frac{d_l}{2\epsilon}\underline{p}^T C_l\underline{p} - \frac{d_l\epsilon}{2}\|\underline{p}\|_{M_l}^2\\ =& -\theta_1\|\underline{u}\|_{A_l}^2 + \theta_2\|\underline{p}\|_{M_l}^2 - \theta_3\underline{p}^T C_l\underline{p}\end{aligned} \tag{5.5.40}$$

with c_l and d_l given in Lemma 5.2 and positive constants $\theta_1 := \frac{1}{2\epsilon}$, $\theta_2 := c_l - \frac{\epsilon}{2}(1+d_l)$ and $\theta_3 := \frac{d_l}{2\epsilon}$ if ϵ is sufficiently small.

Denote $\left(\underline{v}, \underline{q}\right) = \left(\underline{u} + \delta\underline{w}, -\underline{p}\right)$ with a constant $0 < \delta < \min\left(\frac{1}{\theta_1}, \frac{1}{\theta_3}\right)$, then using the above result (5.5.40), we have

$$\begin{aligned}\mathcal{B}_l\left(\underline{u},\underline{p};\underline{v},\underline{q}\right) =& \mathcal{B}_l\left(\underline{u},\underline{p};\underline{u}+\delta\underline{w},-\underline{p}\right)\\ =& \mathcal{B}_l\left(\underline{u},\underline{p};\underline{u},-\underline{p}\right) + \delta\mathcal{B}_l\left(\underline{u},\underline{p};\underline{w},0\right)\\ \geq& \|\underline{u}\|_{A_l}^2 + \underline{p}^T C_l \underline{p} - \delta\theta_1\|\underline{u}\|_{A_l}^2 + \delta\theta_2\|\underline{p}\|_{M_l}^2 - \delta\theta_3 \underline{p}^T C_l \underline{p}\\ \geq& \theta_4\left(\|\underline{u}\|_{A_l}^2 + \|\underline{p}\|_{M_l}^2\right)\end{aligned} \tag{5.5.41}$$

with a properly chosen constant $\theta_4 > 0$.

Together with

$$\|\underline{v}\|_{A_l} + \|\underline{q}\|_{M_l} = \|\underline{u} + \delta w\|_{A_l} + \|q\|_{M_l} \leq (1+\delta)\left(\|\underline{u}\|_{A_l} + \|\underline{p}\|_{M_l}\right), \tag{5.5.42}$$

we obtain

$$\frac{\mathcal{B}_l\left(\underline{u},\underline{p};\underline{v},\underline{q}\right)}{\|\underline{v}\|_{A_l} + \|\underline{q}\|_{M_l}} \geq \theta_5 \frac{\|\underline{u}\|_{A_l}^2 + \|\underline{p}\|_{M_l}^2}{\|\underline{u}\|_{A_l} + \|\underline{p}\|_{M_l}} \geq \frac{\theta_5}{2}\left(\|\underline{u}\|_{A_l} + \|\underline{p}\|_{M_l}\right) \tag{5.5.43}$$

with $\theta_5 = \frac{\theta_4}{1+\delta}$. □

5.6 Scaling for stabilized convection terms

Another non-standard h-dependent term of order h comes from the SUPG stabilization for the convection-dominant term. Denote by $A_{s,1}$ the matrix assembled from the finest level for this stabilization. As suggested in [161], we do the coarsening for $A_{s,l}$ with a suitable scaling factor, i.e.

$$A_{s,l+1} = \left(\frac{n_l}{n_{l+1}}\right)^{1/3} \tilde{I}_l^{l+1} A_{s,l} \tilde{I}_{l+1}^l. \tag{5.6.1}$$

where n_l is the number of nodes on level l, and we do the standard coarsening for the remaining part of A_l.

5.7 Smoothers for the saddle point problem

Once the systems of equations on all levels are constructed, we need a smoother for the coupled system on each level. We use the Braess-Sarazin smoother and the Schwarz-type smoothers.

5.7.1 Braess-Sarazin smoother

This smoother introduced in [27] has a smoothing property with a rate of $\mathcal{O}\left(\frac{1}{m}\right)$. It consists of applications of the *inexact symmetric Uzawa algorithm*:

$$\begin{aligned}\hat{A}(\hat{u}_{k+1} - u_k) &= f - Au_k - B^T p_k,\\ \hat{S}(p_{k+1} - p_k) &= B\hat{u}_{k+1} - Cp_k - g,\\ \hat{A}(u_{k+1} - \hat{u}_{k+1}) &= -B^T(p_{k+1} - p_k),\end{aligned} \tag{5.7.1}$$

where $\hat{A}$ and $\hat{S}$ are symmetric positive definite preconditioners for A and the (negative) inexact Schur complement $C + B\hat{A}^{-1}B^T$, respectively. This can be seen as a preconditioned Richardson method applied to the system (5.1.6) with a symmetric and indefinite preconditioner

$$\hat{K} = \begin{pmatrix} \hat{A} & B^T \\ B & B\hat{A}^{-1}B^T - \hat{S} \end{pmatrix}. \tag{5.7.2}$$

Sufficient conditions for obtaining the smoothing property are given in the following theorem, see [176].

Theorem 5.4 (W. Zulehner)**.** *Let A and C be symmetric positive semi-definite matrices, and $\hat{A}$ and $\hat{S}$ symmetric positive definite matrices, satisfying*

$$\begin{aligned} &\hat{A} \geq A, \\ &\hat{S} \leq C + B\hat{A}^{-1}B^T, \\ &\hat{S} \geq \frac{1}{(1+\beta)}(C + B\hat{A}^{-1}B^T), \end{aligned} \tag{5.7.3}$$

and

$$\|\hat{A}\|_{l_2} \leq c_1 h^{-2}, \quad \|\hat{A}^{-1}\|_{l_2} \leq c_2 h^2, \quad \|B\|_{l_2} \leq c_3 h^{-2}, \quad \|C\|_{l_2} \leq c_4 h^{-2}. \tag{5.7.4}$$

Then the smoothing property

$$\|K\mathcal{S}^m\|_{l_2} \leq \eta(m) h^{-2} \tag{5.7.5}$$

holds with

$$\eta(m) = \max\left[(1+\rho)\rho^{m-1}, \frac{(m-1)^{m-1}}{m^m}\right] \quad \textit{and} \quad \rho = \beta + \sqrt{\beta^2 + \beta}.$$

It is easy to show that for $\beta < 1/3$ and $m > 1$, we get the smoothing rate $\mathcal{O}(\frac{1}{m})$.

5.7.2 Schwarz-type smoothers

This patch smoother was introduced in [155] when using finite volume discretization of the Navier-Stokes equations on a staggered grid (pressure nodes at cell-centers and velocity nodes at cell-faces). The smoothing procedure is performed by solving small local sub-problems set up for each cell with one pressure degree of freedom and the connected velocity unknowns (see Figure 5.1) and by using a multiplicative Schwarz iteration. For the additive version, the smoothing property was analyzed under a finite element framework in [141].

We introduce local prolongations for each fixed level l such that local sub-problems can be set up:

$$\pi_j : \mathbb{R}^{n_{l,j}} \to \mathbb{R}^{n_l}, \quad \sigma_j : \mathbb{R}^{m_{l,j}} \to \mathbb{R}^{m_l},$$

where j is the index of sub-problems, $n_{l,j}$ and $m_{l,j}$ are local dimensions with associated restrictions:

$$\pi_j^T : \mathbb{R}^{n_l} \to \mathbb{R}^{n_{l,j}}, \quad \sigma_j^T : \mathbb{R}^{m_l} \to \mathbb{R}^{m_{l,j}}.$$

We assume

$$\sum_j \pi_j \pi_j^T = I \tag{5.7.6}$$

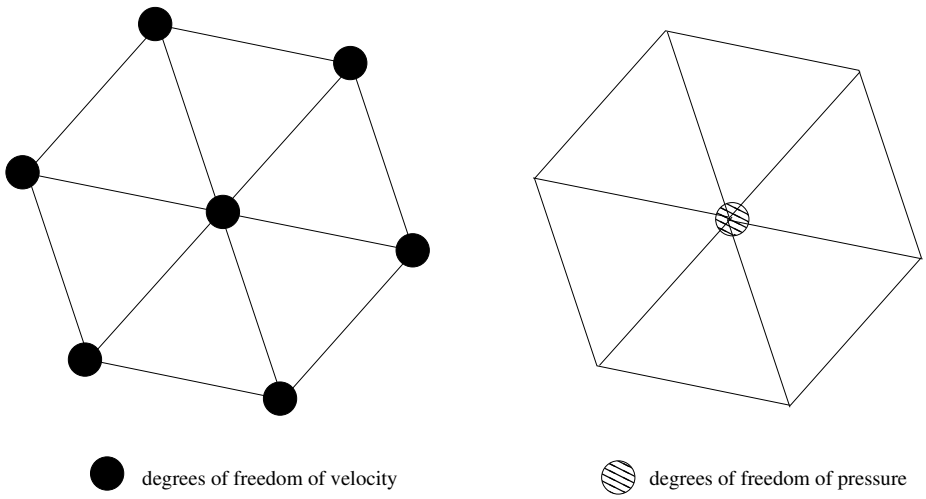


Figure 5.1: Degrees of freedom for velocity and pressure on one cell.

and

$$\sum_j \sigma_j \sigma_j^T \tag{5.7.7}$$

is nonsingular. Starting from previous approximations u^k and p^k of the exact solutions u and p, we have the iterative methods of the following form:

$$u^{k+1} = u^k + \sum_j \pi_j w^k, \quad p^{k+1} = p^k + \sum_j \sigma_j r^k,$$

where (w^k, r^k) solves the local saddle point problem:

$$\begin{pmatrix} \hat{A}_j & B_j^T \\ B_j & \frac{1}{\beta}\left[(\beta-1)B_j\hat{A}_j^{-1}B_j^T - C_j\right] \end{pmatrix} \begin{pmatrix} w_j^k \\ r_j^k \end{pmatrix} = \begin{pmatrix} \pi_j^T\left[f - Au^k - B^T p^k\right] \\ \sigma_j^T\left[g - Bu^k + Cp^k\right] \end{pmatrix}.$$

with a relaxation parameter β and the following relations:

$$\begin{aligned} \pi_j^T \hat{A} &= \hat{A}_j \pi_j^T, \\ \sigma_j^T B &= B_j \pi_j^T, \\ C_j &= \sigma_j^T C \sigma_j, \end{aligned} \tag{5.7.8}$$

where $\hat{A}$ is a preconditioner for A. Then we have the following theorem (see [141]):

Theorem 5.5 (J. Schöberl and W. Zulehner)**.** *The additive Schwarz method can be reinterpreted as a preconditioned Richardson iteration with the preconditioner*

$$\hat{\mathcal{K}} = \begin{pmatrix} \hat{A} & B^T \\ B & B\hat{A}^{-1}B^T - \hat{S} \end{pmatrix}, \tag{5.7.9}$$

where

$$\hat{S} = (\sum_j \varphi_j \hat{S}_j^{-1} \varphi_j^T)^{-1}, \quad \hat{S}_j = \frac{1}{\beta}(C_j + B_j \hat{A}_j^{-1} B_j^T).$$

Then it can be analyzed in the class of *inexact symmetric Uzawa* method (5.7.1). The smoothing property of this method is summarized in the following theorem (see [141]):

Theorem 5.6 (J. Schöberl and W. Zulehner)**.** *Using the preconditioner $\hat{\mathcal{K}}$ as in (5.7.9) with an SPD $n \times n$ matrix $\hat{A}$ and an SPD $m \times m$ matrix $\hat{S}$, satisfying*

$$\hat{A} \geq A,$$
$$\hat{S} \geq C + B\hat{A}^{-1}B^T,$$

and

$$\|\hat{K} - K\|_{l_2} \leq c\|K\|_{l_2}$$

Then we get the smoothing property

$$\|K\mathcal{S}^m\|_{l_2} \leq \eta(m)\|K\|_{l_2}$$

with $\eta(m) = O(\frac{1}{\sqrt{m}})$.

We present a method of constructing the local sub-problems which satisfy the requirements (5.7.6), (5.7.7) and (5.7.8). For each sub-problem, we pick up one pressure unknown associated to a pressure vertex and the velocity unknowns associated to the velocity vertices which are connected to this pressure vertex.

Then σ_j is chosen as the canonical embedding from $\mathbb{R}$ to $\mathbb{R}^{m_l}$. For π_j, we first construct the canonical embedding $\hat{\pi}_j : \mathbb{R}^{n_{l,j}} \to \mathbb{R}^{n_l}$, and by scaling each row i of it by $1/\sqrt{s_i}$, where s_i is the number of sub-problems containing the velocity vertex x_i, we obtain π_j.

A computationally cheap version for constructing $\hat{A}$ is given by

$$\hat{A} = \frac{1}{\alpha}\mathrm{diag}(A) \tag{5.7.10}$$

with a proper chosen scaling factor α such that $\hat{A} \geq A$. The construction of $\hat{A}_j$ is then given by

$$\hat{A}_j = \pi_j^T \hat{A} \pi_j. \tag{5.7.11}$$

For B_j, we firstly construct

$$\hat{B}_j = \sigma_j^T B \pi_j, \tag{5.7.12}$$

and then B_j is obtained by scaling each row i of $\hat{B}_j$ with $\sqrt{s_i}$.

For completeness, we describe some other smoothers:

5.7.3 A standard smoother for the squared system

One possibility would be applying Richardson iterations on the squared system

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix}^T \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix}^T \begin{pmatrix} f \\ g \end{pmatrix} \tag{5.7.13}$$

as introduced in [158], which leads to a smoothing rate $\eta(m) = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$ where m denotes the smoothing steps.

5.7.4 Transforming smoothers

In [168, 169], so called transforming smoothers are introduced by applying damped Jacobi, Gauss-Seidel and ILU smoothers for the transformed system

$$K_1 = KK_2^{-1}, \tag{5.7.14}$$

where

$$K = \underbrace{\begin{pmatrix} A & 0 \\ B & E \end{pmatrix}}_{=:K_1} \underbrace{\begin{pmatrix} I & A^{-1}B^T \\ 0 & -E^{-1}S \end{pmatrix}}_{=:K_2} \tag{5.7.15}$$

with an arbitrary positive definite matrix E, e.g. $E = S$. A smoothing rate of order $\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$ for damped Jacobi, and $\mathcal{O}\left(\frac{\ln m}{\sqrt{m}}\right)$ for Gauss-Seidel and ILU will be obtained.

5.8 Appendix on M-matrices and matrices of essentially positive type

The M-matrix is defined as in [74].

Definition 5.7. *A matrix A is called M-matrix if*

1. $A_{i,i} > 0$ *for all* i,
2. $A_{i,j} \leq 0$ *for all* $i \neq j$,
3. A *is regular and* $A^{-1} \geq 0$ *component-wise.*

In many applications, small positive off-diagonal entries will appear. The class of *essentially positive type* matrices was introduced for capturing these *almost* M-matrices with small positive off-diagonal entries, see [145, 31].

Definition 5.8. *A positive definite matrix A is called essentially positive type if there exists a constant $c > 0$ such that for all e,*

$$\sum_{i,j}(-A_{i,j})(e_i - e_j)^2 \geq c\sum_{i,j}(-A_{i,j}^-)(e_i - e_j)^2, \tag{5.8.1}$$

where $A_{i,j}^-$ is defined as

$$A_{i,k}^- = \begin{cases} A_{i,k} & \text{if } A_{i,k} < 0, \\ 0 & \text{else .} \end{cases} \tag{5.8.2}$$

For this type of matrix, a property holds (see [161]).

Lemma 5.9. *For an essentially positive type matrix,*

$$\frac{2}{c}D \geq A, \tag{5.8.3}$$

where D is the diagonal of A, c is given in (5.8.1), and we write $A \geq B$ ($A > B$) if $A - B$ is positive semi-definite (positive definite).

5.9 Summary

The task of this chapter was to apply the AMG to the structure and fluid sub-problems. We recalled the ingredients of the AMG in general. The main contribution of this chapter is the AMG application to the saddle point problem arising from the stabilized finite element discretization on hybrid meshes for the fluid sub-problem. We are able to construct stabilized P_1-P_1 hierarchy for hybrid meshes.

Chapter 6

Numerical simulations

In this chapter, we firstly report some numerical results for the structure and fluid sub-problems on the unit cube domain (see Figure 6.1). Then we present test results for the

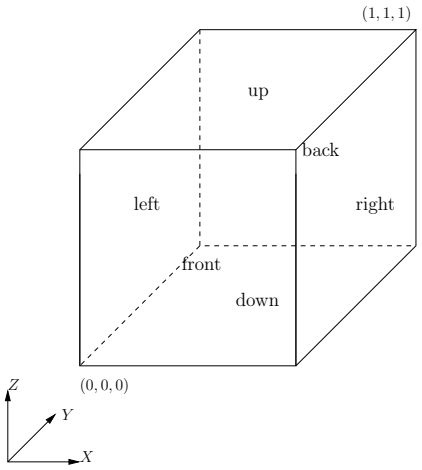


Figure 6.1: A domain of unit cube

fluid-structure interaction problem on a cylindrical domain consisting of the inner cylindrical fluid domain of length 50mm and radius 5mm and the surrounding structure domain with thickness 0.5mm, see the cutting plane along the axis in Figure 6.2.

For the unit cube domain, we have the following four levels of tetrahedral meshes (see Figure 6.3(a)) and hexahedral meshes (see Figure 6.3(b)), see Table 6.1. On the tetrahedral meshes, we use the standard linear finite element discretization, and on the hexahedral meshes, we use the finite element discretization as we discussed in Chapter 4. We compare the numerical results on these two types of meshes.

Table 6.1: Four levels of uniform meshes on a unit cube

Levels	L_1	L_2	L_3	L_4
Number of vertices	$125(=5^3)$	$729(=9^3)$	$4{,}913(=17^3)$	$35{,}937(=33^3)$
Mesh size (h)	1/4	1/8	1/16	1/32

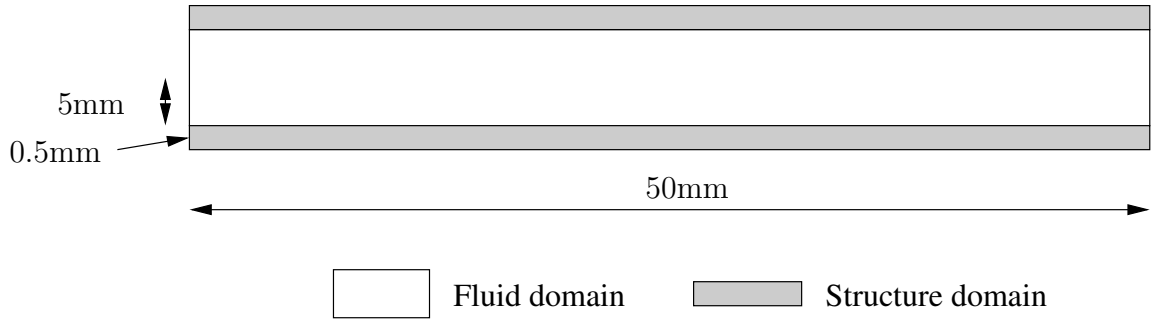
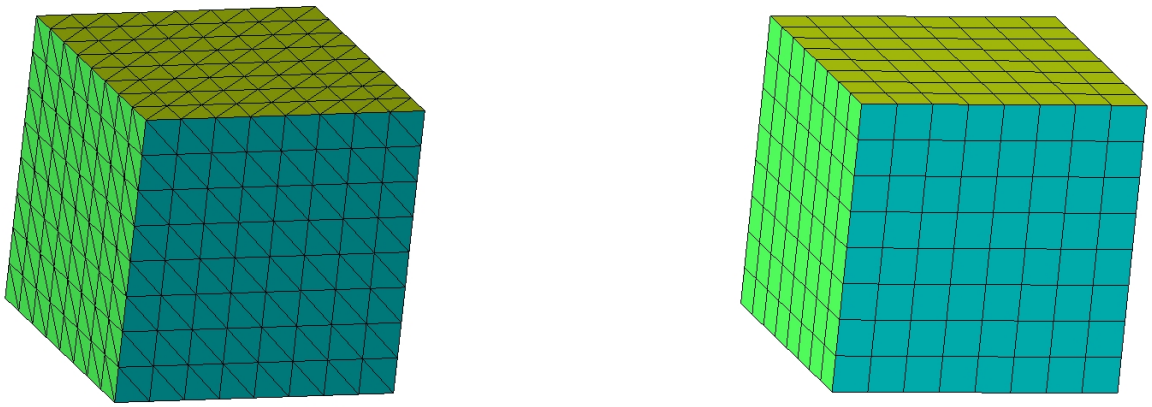


Figure 6.2: The cutting plane of the domain for the fluid-structure interaction problem

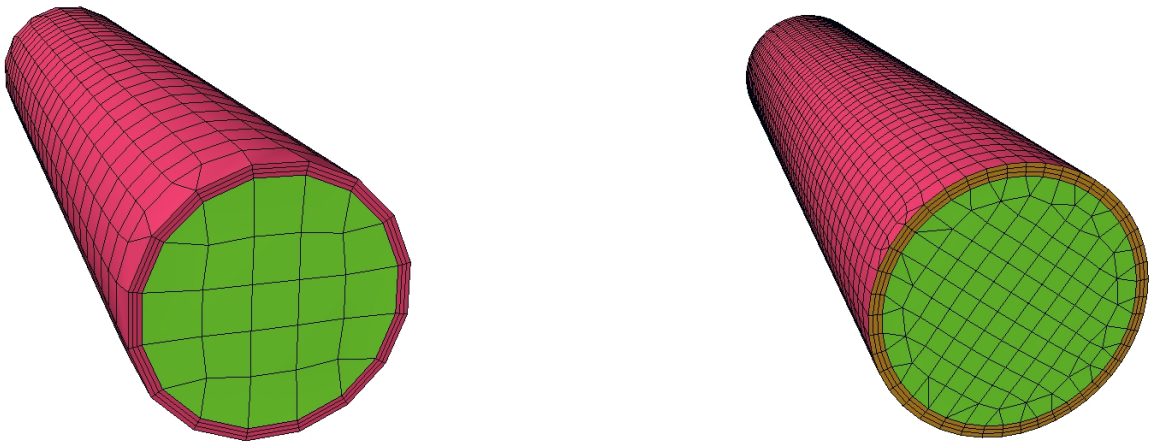


(a) Tetrahedral mesh

(b) Hexahedral mesh

Figure 6.3: Two uniform meshes

For the FSI problem, we use two hybrid meshes which contains tetrahedral, prismatic and hexahedral elements, see their illustrations in Figure 6.4. For the coarse mesh, it has about $4,000$ vertices, and for the fine mesh, about $20,000$ vertices.



(a) FSI coarse mesh (b) FSI fine mesh

Figure 6.4: FSI coarse and fine meshes

All these meshes are generated and offered by courtesy from Dipl.-Ing. Ferdinand Kickinger, CSS CAE Software Solutions[1], Wolfkersbhelstr. 23, A-3730 Eggenburg, Austria, who was a former member of the Institute of Computational Mathematics (NuMa) of the Johannes Kepler University Linz.

6.1 Numerical results for the structure problem

The material parameters for the structure problem are given by

$$\mu^l = 1.15 \times 10^6, \quad \lambda^l = 1.73 \times 10^6, \quad \rho_{\rm s} = 1.2. \tag{6.1.1}$$

6.1.1 Numerical study for an elastostatic model problem

We firstly test an elastostatic model problem:

$$-\mathrm{div}(\sigma_{\rm s}(d)) = f_{\rm s} \quad \text{in } \Omega, \tag{6.1.2}$$

with the right hand side given by

$$f_{\rm s} = \Big(0, \mu^l \pi^2 \sin(\pi x), 0\Big)^T. \tag{6.1.3}$$

We prescribe the Dirichlet boundary conditions on the left and right faces of the unit cube domain

$$d = (0, \sin(\pi x), 0)^T, \tag{6.1.4}$$

[1] http://www.meshing.at

the Neumann boundary conditions on the other faces

$$\sigma_{\mathrm{s}}(d)n = \left(\mu^l \pi \cos(\pi x) n_2, \mu^l \pi \cos(\pi x) n_1, 0\right)^T, \tag{6.1.5}$$

where $n = (n_1, n_2, n_3)^T$ is a unitary normal vector.

The analytical solution for this model problem is given by

$$d = (0, \sin(\pi x), 0)^T. \tag{6.1.6}$$

The following numerical results are given:

1. relative discretization errors with respect to L^2 norms (see Table 6.2) and H^1 seminorms (see Table 6.3),
2. number of preconditioned conjugate gradient (PCG) iterations with AMG preconditioning (see Section 5.3) on four levels of meshes (see Table 6.1) for a relative residual error reduction by a factor of 10^{-12} (see Table 6.4)
3. Figure 6.5 shows the deformed structure domain (the deformation is amplified by a factor of 10).

Table 6.2: $\|d_h - d\|_{L^2}/\|d\|_{L^2}$ on meshes of four levels (see Table 6.1)

Mesh size	Hexahedral mesh	Tetrahedral mesh
$h = 1/4$	0.0311151	0.0369359
$h = 1/8$	0.00823565	0.0100763
$h = 1/16$	0.00210126	0.00261605
$h = 1/32$	0.000528806	0.000663414

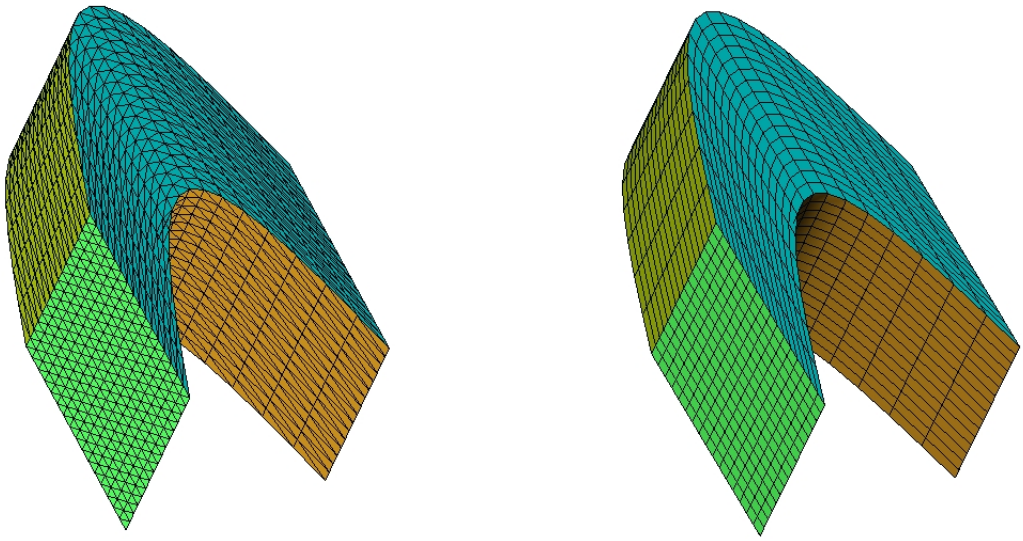
Table 6.3: $|d_h - d|_{H^1}/|d|_{H^1}$ on meshes of four levels (see Table 6.1)

Mesh size	Hexahedral mesh	Tetrahedral mesh
$h = 1/4$	0.213279	0.169604
$h = 1/8$	0.103278	0.0812706
$h = 1/16$	0.0512285	0.0402202
$h = 1/32$	0.025563	0.0200577

Table 6.4: Number of PCG iterations on meshes of four levels (see Table 6.1)

Mesh size	Hexahedral mesh	Tetrahedral mesh
$h = 1/4$	9	13
$h = 1/8$	9	18
$h = 1/16$	10	16
$h = 1/32$	10	16

From the numerical results, it is easy to see that we obtain $O(h^2)$ and $O(h)$ convergence rates with respect to the L^2 norms and H^1 seminorms of the discretization errors in accordance with the theorem. The iteration numbers are independent of mesh levels.



(a) Deformation with the tetrahedral mesh

(b) Deformation with the hexahedral mesh

Figure 6.5: Deformed domains of the elastostatic model problem on level 3 with an amplification factor 10.

6.1.2 Numerical study for an elastodynamic model problem

We then test an elastodynamic model problem:

$$\rho_{\mathrm{s}} \frac{\partial^2 d}{\partial t^2} - \mathrm{div}(\sigma_{\mathrm{s}}(d)) = f_{\mathrm{s}} \quad \text{in } \Omega, \tag{6.1.7}$$

with the right hand side given by

$$f_{\mathrm{s}} = 0. \tag{6.1.8}$$

We prescribe Dirichlet boundary conditions on the left and right faces of the unit cube domain

$$d = (0, \sin(\pi x)\sin(c\pi t), 0)^T, \tag{6.1.9}$$

where $c = \sqrt{\mu^l/\rho_{\mathrm{s}}}$, the Neumann boundary conditions on the other faces

$$\sigma_{\mathrm{s}}(d)n = \left(\mu^l \pi \cos(\pi x)\sin(c\pi t)n_2, \mu^l \pi \cos(\pi x)\sin(c\pi t)n_1, 0\right)^T, \tag{6.1.10}$$

and the initial data

$$d_0 = (0,0,0)^T, \quad d_{t,0} = (0, c\pi \sin(\pi x), 0)^T. \tag{6.1.11}$$

The analytical solution for this model problem is given by

$$d = (0, \sin(\pi x)\sin(c\pi t), 0)^T. \tag{6.1.12}$$

The following simulation results are presented:

1. relative discretization error respect to L^2 norms (see Table 6.5) and H^1 seminorms (see Table 6.6),

2. number of preconditioned conjugate gradient (PCG) iterations with AMG preconditioning (see Section 5.3) on four levels for a relative residual error reduction by a factor of 10^{-12} at each time step (see Table 6.7),

3. Figure 6.6 shows the deformed structure domains for different time levels (the deformation is amplified by a factor of 5).

Table 6.5: $\|d_h - d\|_{L^2}/\|d\|_{L^2}$ at time $t = 2.5e-04s$ on meshes of four levels (see Table 6.1)

Time step size	Mesh size	Hexahedral mesh	Tetrahedral mesh
$\delta t = 2.5e-04$	$h = 1/4$	0.122331	0.121997
$\delta t = 6.25e-05$	$h = 1/8$	0.0345195	0.0348679
$\delta t = 1.5625e-05$	$h = 1/16$	0.00861375	0.00870446
$\delta t = 3.90625e-06$	$h = 1/32$	0.00214757	0.00217012

Table 6.6: $|d_h - d|_1/|d|_1$ at time $t = 2.5e-04s$ on meshes of four levels (see Table 6.1)

Time step size	Mesh size	Hexahedral mesh	Tetrahedral mesh
$\delta t = 2.5e-04$	$h = 1/4$	0.23636	0.199915
$\delta t = 1.25e-04$	$h = 1/8$	0.107719	0.0870238
$\delta t = 6.25e-05$	$h = 1/16$	0.0518171	0.040979
$\delta t = 3.125e-05$	$h = 1/32$	0.0256378	0.0201541

Table 6.7: Number of PCG iterations per time step on meshes of four levels (see Table 6.1)

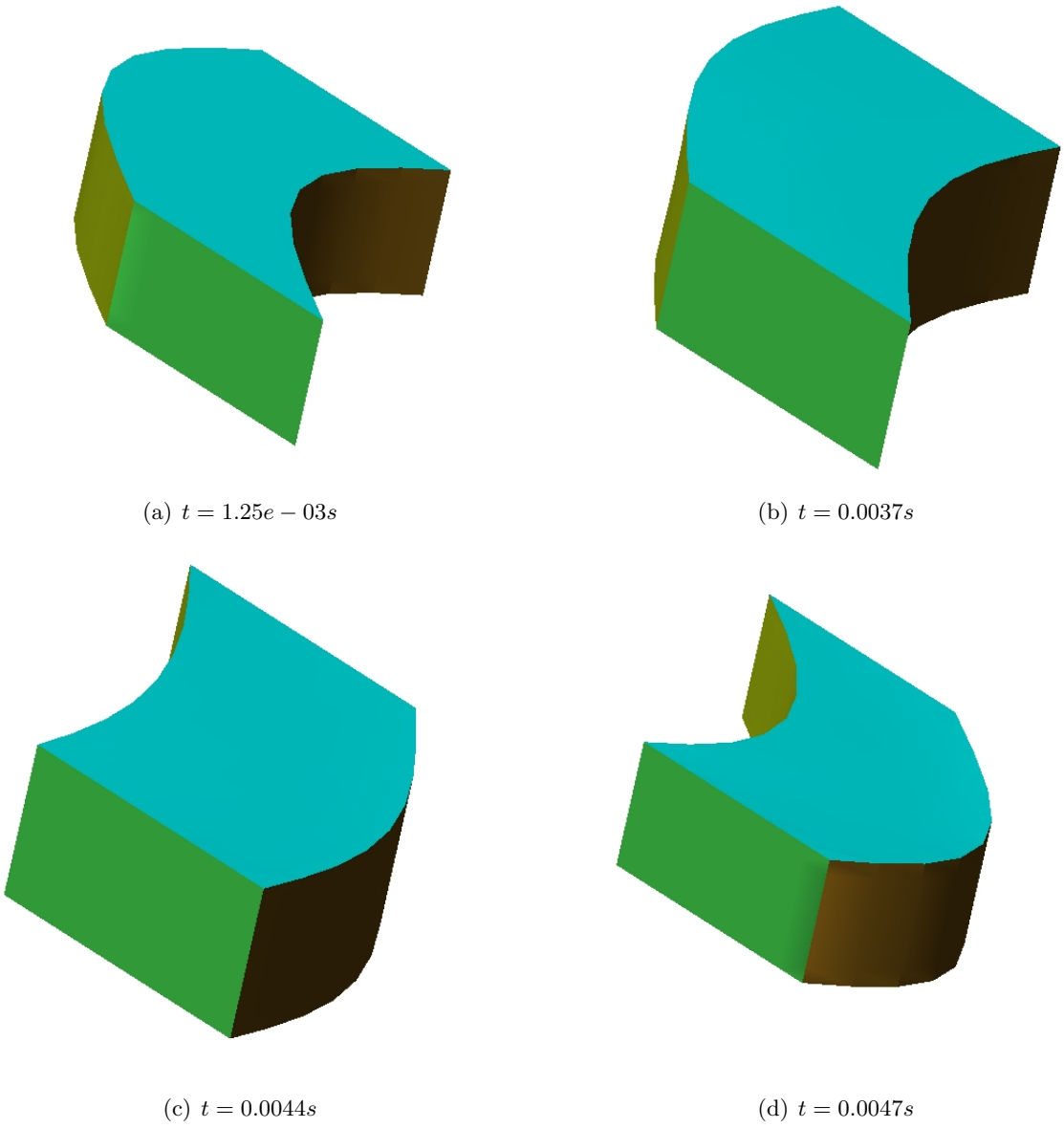
Mesh size	Hexahedral mesh	Tetrahedral mesh
$h = 1/4$	7	11
$h = 1/8$	7	10
$h = 1/16$	9	9
$h = 1/32$	15	14

As we can see from the numerical results, we obtain $O(h^2)$ and $O(h)$ convergence rates with respect to the L^2 norms and H^1 seminorms in accordance with the theorem. The iteration numbers are independent of mesh levels.

Note that we use a first order Newmark method (see Section 3.3.1) for the time discretization. So, from one level to the next level, for measuring the L^2 norms, we reduce the time step size by half, and for the H^1 seminorms, we reduce it by fourth. By these choices, we observe the expected reduction of the discretization errors by a factor of 1/4 and 1/2 in space, with respect to the L^2 norms and H^1 seminorms.

6.2 Numerical results for the fluid problem

In this section, we will firstly compare the AMG performance using different smoothers for the stationary Stokes problem. Then we will concentrate on the convergence results and



(a) $t = 1.25e - 03s$

(b) $t = 0.0037s$

(c) $t = 0.0044s$

(d) $t = 0.0047s$

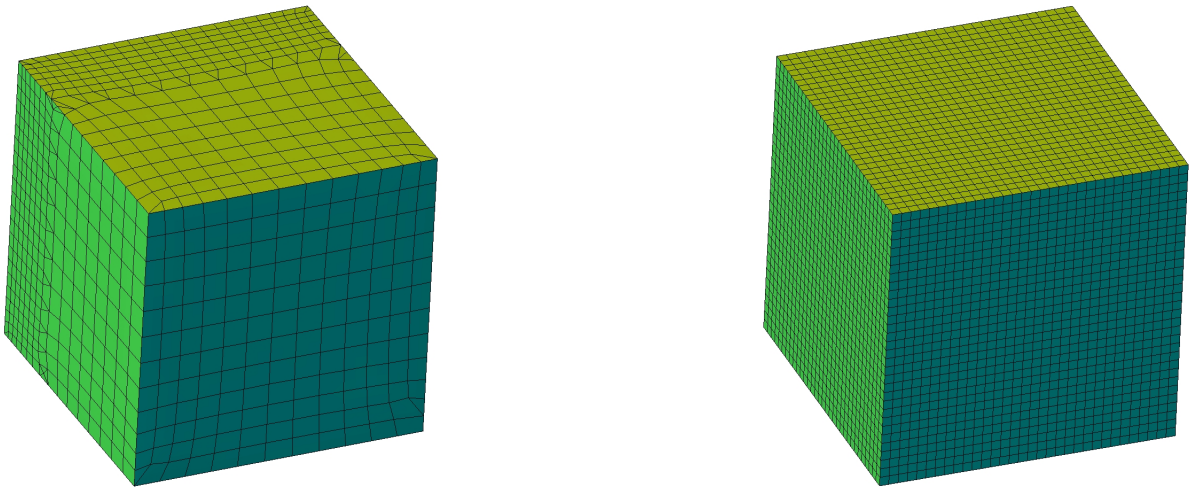
Figure 6.6: Deformed domains at different time levels.

AMG performance for the stationary and nonstationary Stokes problems, and the Oseen problem. The material parameters for the fluid problems are given by

$$\mu = 0.035, \quad \rho_{\mathrm{f}} = 1.0. \tag{6.2.1}$$

6.2.1 AMG performance with different smoothers for a stationary Stokes model problem

In this section, we use the AMG solver for the saddle point problem (see Section 5.4-Section 5.7), and we compare the AMG performance with different smoothers (see Section 5.7) applied to the stationary Stokes problem. We test the problem on a general hybrid mesh containing all four types of elements (see Figure 6.7(a)) and a specific mesh containg only hexahedral elements (see Figure 6.7(b)). The first mesh has $7,603$ vertices. The overall



(a) A hybrid mesh

(b) A hexahedral mesh

Figure 6.7: Two meshes for testing different smoothers.

number of velocity and pressure unknowns from the MINI-element discretization on this mesh is $30,412$. The second mesh is from the level L_4 described in Table 6.1.

We given an example for the Stokes problem:

$$\begin{cases} -2\mu \mathrm{div}\varepsilon(u) + \nabla p = f_{\mathrm{f}} & \text{in } \Omega, \\ \mathrm{div} u = 0 & \text{in } \Omega, \end{cases} \tag{6.2.2}$$

with the right hand side given by

$$f_{\mathrm{f}} = \begin{pmatrix} 3\mu\pi^2 \sin(x)\cos(y)\cos(z) - \pi \sin(x)\cos(y)\cos(z) \\ -6.0\mu\pi^2 \cos(x)\sin(y)\cos(z) - \pi \cos(x)\sin(y)\cos(z) \\ 3.0\mu\pi^2 \cos(x)\cos(y)\sin(z) - \pi \cos(x)\cos(y)\sin(z) \end{pmatrix}. \tag{6.2.3}$$

We prescribe Dirichlet boundary conditions on the up, down, front and back faces of the unit cube

$$u = \begin{pmatrix} \sin(\pi x)\cos(\pi y)\cos(\pi z) \\ -2\cos(\pi x)\sin(\pi y)\cos(\pi z) \\ \cos(\pi x)\cos(\pi y)\sin(\pi z) \end{pmatrix}, \tag{6.2.4}$$

the Neumann boundary conditions on the left and right faces

$$\begin{aligned}
&\mu\pi \begin{pmatrix} -\cos(x)\cos(y)\cos(z) + \cos(x)\cos(y)\cos(z) + \sin^3(1.0) \\ -2.0\sin(x)\sin(y)\cos(z) \\ \sin(x)\cos(y)\sin(z) \end{pmatrix}, \\
&\mu\pi \begin{pmatrix} \cos(x)\cos(y)\cos(z) - \cos(x)\cos(y)\cos(z) - \sin^3(1.0) \\ 2\sin(x)\sin(y)\cos(z) \\ -\sin(x)\cos(y)\sin(z) \end{pmatrix},
\end{aligned} \tag{6.2.5}$$

respectively.

The analytical solution for this model problem is given by

$$u = \begin{pmatrix} \sin(\pi x)\cos(\pi y)\cos(\pi z) \\ -2\cos(\pi x)\sin(\pi y)\cos(\pi z) \\ \cos(\pi x)\cos(\pi y)\sin(\pi z) \end{pmatrix}, \quad p = \cos(\pi x)\cos(\pi y)\cos(\pi z) + \sin(\pi)^3. \tag{6.2.6}$$

We set the relative residual error by 10^{-8} as the stop criterion for all AMG iterations with different smoothers. The comparison of three smoothers, Braess-Sarazin (BS), additive Schwarz (AS) and multiplicative Schwarz (MS) on the hybrid mesh is given in Table 6.8.

Table 6.8: Comparison of three smoothers on the hybrid mesh (see Figure 6.7(a)).

	Smoothers		
	BS	AS	MS
#AMG iterations	13	12	6
#Smoothing steps (pre+post)	12 (6+6)	30 (15+15)	4 (2+2)
Time in seconds	129s	373s	78s

The comparison on the hexahedral mesh is listed in Table 6.9.

Table 6.9: Comparison of three smoothers on the hexahedral mesh (see Figure 6.7(b)).

	Smoothers		
	BS	AS	MS
#AMG iterations	15	7	4
#Smoothing steps (pre+post)	12 (6+6)	30 (15+15)	8 (4+4)
Time in seconds	794s	1107s	524s

Although it turns out that the multiplicative Schwarz smoother is more efficient than the Braess-Sarazin smoother which is more efficient than the additive Schwarz smoother, the AMG with the Braess-Sarazin smoother is more reliable than the other two. For the Schwarz smoothers, it is essential to adjust some scaling factors in order to obtain the convergence.

Hence, from now on, we will use the Braess-Sarazin smoother in the AMG solver for solving the fluid problems.

6.2.2 Numerical study for a stationary Stokes model problem

We use the same example for the Stokes problem as discribed in Section 6.2.1.

The following results will be presented:

1. relative discretization error with respect to L^2 norms and H^1 seminorms on hexahedral (see Table 6.10) and tetrahedral (see Table 6.11) meshes,

2. number of AMG iterations (see Table 6.12) and generalized minimal residual method (GMRES) iterations (see Table 6.13) with AMG preconditioning on four levels for a relative residual error reduction by a factor of 10^{-10},

Table 6.10: Relative discretization errors on hexahedral meshes (see Figure 6.3(b))

Mesh size	$\frac{\|p-p_h\|_{L^2}}{\|p\|_{L^2}}$	$\frac{\|u-u_h\|_{L^2}}{\|u\|_{L^2}}$	$\frac{\|u-u_h\|_{H^1}}{\|u\|_{H^1}}$
$h = 1/4$	0.0324146	0.0466122	0.296207
$h = 1/8$	0.00590018	0.018217	0.139401
$h = 1/16$	0.00125606	0.00509055	0.0684342
$h = 1/32$	0.000285436	0.0013194	0.0340417

Table 6.11: Relative discretization errors on tetrahedral meshes (see Figure 6.3(a))

Mesh size	$\frac{\|p-p_h\|_{L^2}}{\|p\|_{L^2}}$	$\frac{\|u-u_h\|_{L^2}}{\|u\|_{L^2}}$	$\frac{\|u-u_h\|_{H^1}}{\|u\|_{H^1}}$
$h = 1/4$	0.195372	0.169756	0.604462
$h = 1/8$	0.0429989	0.0534867	0.262313
$h = 1/16$	0.010816	0.0157134	0.126603
$h = 1/32$	0.00303433	0.00418255	0.0626473

Table 6.12: Number of AMG iterations on meshes of four levels (see Table 6.1)

Mesh size	Hexahedral mesh	Tetrahedral mesh
$h = 1/4$	7	10
$h = 1/8$	9	10
$h = 1/16$	10	10
$h = 1/32$	10	10

As expected, from the numerical results, we observe the $O(h^2)$ convergence rate with respect to the L^2 norm for both pressure and velocity, and $O(h)$ convergence rate with respect to the H^1 seminorm for the velocity. The iteration numbers using AMG and PGMRES (with AMG preconditioning) methods are independent of mesh levels.

6.2.3 Numerical study for a time dependent Stokes model problem

The example for time-dependent Stokes problem is given by the following equations:

$$\begin{cases} \rho_{\mathrm{f}} \dfrac{\partial u}{\partial t} - 2\mu \mathrm{div}\varepsilon(u) + \nabla p = f_{\mathrm{f}} & \text{in } \Omega, \\ \mathrm{div} u = 0 & \text{in } \Omega, \end{cases} \tag{6.2.7}$$

Table 6.13: Number of PGMRES iterations on meshes of four levels (see Table 6.1)

Mesh size	Hexahedral mesh	Tetrahedral mesh
$h = 1/4$	6	9
$h = 1/8$	9	12
$h = 1/16$	8	12
$h = 1/32$	9	12

with the right hand side give by

$$f_{\mathrm{f}} = \sin(t) \begin{pmatrix} 3\mu\pi^2 \sin(x)\cos(y)\cos(z) - \pi \sin(x)\cos(y)\cos(z) \\ -6.0\mu\pi^2 \cos(x)\sin(y)\cos(z) - \pi \cos(x)\sin(y)\cos(z) \\ 3.0\mu\pi^2 \cos(x)\cos(y)\sin(z) - \pi \cos(x)\cos(y)\sin(z) \end{pmatrix}. \tag{6.2.8}$$

We prescribe Dirichlet boundary conditions on the up, down, front and back faces of the unit cube

$$u = \sin(t) \begin{pmatrix} \sin(\pi x)\cos(\pi y)\cos(\pi z) \\ -2\cos(\pi x)\sin(\pi y)\cos(\pi z) \\ \cos(\pi x)\cos(\pi y)\sin(\pi z) \end{pmatrix}, \tag{6.2.9}$$

the Neumann boundary conditions on the left and right faces

$$\begin{aligned} &\mu\pi \sin(t) \begin{pmatrix} -\cos(x)\cos(y)\cos(z) + \cos(x)\cos(y)\cos(z) + \sin^3(1.0) \\ -2.0\sin(x)\sin(y)\cos(z) \\ \sin(x)\cos(y)\sin(z) \end{pmatrix}, \\ &\mu\pi \sin(t) \begin{pmatrix} \cos(x)\cos(y)\cos(z) - \cos(x)\cos(y)\cos(z) - \sin^3(1.0) \\ 2\sin(x)\sin(y)\cos(z) \\ -\sin(x)\cos(y)\sin(z) \end{pmatrix}, \end{aligned} \tag{6.2.10}$$

respectively.

The analytical solution for this model problem is given by

$$u = \sin(t) \begin{pmatrix} \sin(\pi x)\cos(\pi y)\cos(\pi z) \\ -2\cos(\pi x)\sin(\pi y)\cos(\pi z) \\ \cos(\pi x)\cos(\pi y)\sin(\pi z) \end{pmatrix} \tag{6.2.11}$$

and

$$p = \cos(\pi x)\cos(\pi y)\cos(\pi z)\sin(t) + \sin(\pi)^3 \sin(t). \tag{6.2.12}$$

The following numerical results are presented:

1. relative discretization rate with respect to the L^2 norms and H^1 seminorms on hexahedral (see Table 6.14) and tetrahedral (see Table 6.15) meshes at time $t = 0.5s$,
2. number of AMG iterations (see Table 6.16) and generalized minimal residual method (GMRES) iterations (see Table 6.17) with AMG preconditioning on four levels with a relative residual error reduction by a factor of 10^{-10} at each time step,

As expected, from the numerical results, we observe the $O(h^2)$ convergence rate with respect to the L^2 norms for both pressure and velocity, and $O(h)$ convergence rate with respect to the H^1 seminorms for the velocity. The iteration numbers using AMG and PGMRES (with AMG preconditioning) methods are independent of mesh levels.

Table 6.14: Relative discretization errors at time $t = 0.5s$ on hexahedral meshes (see Figure 6.3(b))

Time step size	Mesh size	$\frac{\|p-p_h\|_{L^2}}{\|p\|_{L^2}}$	$\frac{\|u-u_h\|_{L^2}}{\|u\|_{L^2}}$	$\frac{\|u-u_h\|_{H^1}}{\|u\|_{H^1}}$
$\delta t = 5.0e-01$	$h = 1/4$	0.0755348	0.0617607	0.297971
$\delta t = 1.25e-01$	$h = 1/8$	0.00981048	0.0156307	0.139052
$\delta t = 3.125e-02$	$h = 1/16$	0.00150186	0.00385891	0.0683943
$\delta t = 7.8125e-03$	$h = 1/32$	0.000292456	0.000960585	0.0340371

Table 6.15: Relative discretization errors at time $t = 0.5s$ on tetrahedral meshes (see Figure 6.3(a))

Time step size	Mesh size	$\frac{\|p-p_h\|_{L^2}}{\|p\|_{L^2}}$	$\frac{\|u-u_h\|_{L^2}}{\|u\|_{L^2}}$	$\frac{\|u-u_h\|_{H^1}}{\|u\|_{H^1}}$
$\delta t = 5.0e-01$	$h = 1/4$	0.757312	0.1243	0.632039
$\delta t = 1.25e-01$	$h = 1/8$	0.171197	0.0269533	0.269872
$\delta t = 3.125e-02$	$h = 1/16$	0.0436205	0.00617559	0.128315
$\delta t = 7.8125e-03$	$h = 1/32$	0.012951	0.00147254	0.0630016

Table 6.16: Number of AMG iterations on meshes of four levels (see Table 6.1) at each time step

Mesh size	Hexahedral mesh	Tetrahedral mesh
$h = 1/4$	5	7
$h = 1/8$	6	7
$h = 1/16$	5	7
$h = 1/32$	4	6

Table 6.17: Number of PGMRES iterations on meshes of four levels (see Table 6.1) at each time step

Mesh size	Hexahedral mesh	Tetrahedral mesh
$h = 1/4$	5	6
$h = 1/8$	6	6
$h = 1/16$	5	6
$h = 1/32$	5	6

Note that we are using the first-order implict Euler method for the time discretization. From one level to the next level, for measuring the H^1 seminorms and the L^2 norms at the same time, we reduce the time step size by fourth. By this choice, we observe the expected reduction of the discretization errors by a factor of $1/4$ and $1/2$ in space, with respect to the L^2 norms and H^1 seminorms.

6.2.4 Numerical study for a generalized Oseen model problem

The example for the *generalized Oseen equations* is given in the following setting:

$$\begin{cases} \lambda u + \rho_{\mathrm{f}}(w \cdot \nabla)u - 2\mu \mathrm{div}\epsilon(u) + \nabla p = f_{\mathrm{f}} & \text{in } \Omega, \\ \mathrm{div} u = 0 & \text{in } \Omega, \end{cases} \tag{6.2.13}$$

with the right hand side given by

$$\begin{aligned} f_{\mathrm{f}} = & \begin{pmatrix} 3\mu\pi^2 \sin(x)\cos(y)\cos(z) - \pi \sin(x)\cos(y)\cos(z) \\ -6.0\mu\pi^2 \cos(x)\sin(y)\cos(z) - \pi \cos(x)\sin(y)\cos(z) \\ 3.0\mu\pi^2 \cos(x)\cos(y)\sin(z) - \pi \cos(x)\cos(y)\sin(z) \end{pmatrix} \\ & + \lambda \begin{pmatrix} \sin(\pi x)\cos(\pi y)\cos(\pi z) \\ -2\cos(\pi x)\sin(\pi y)\cos(\pi z) \\ \cos(\pi x)\cos(\pi y)\sin(\pi z) \end{pmatrix} \\ & + \mu\pi \begin{pmatrix} \sin(\pi x)\cos(\pi x)\left(\cos^2(\pi y)\cos^2(\pi z) + 2\sin^2(\pi y)\cos^2(\pi z) - \sin(\pi z)\cos(\pi z)\right) \\ \sin(\pi y)\cos(\pi y)\left(2\sin^2(\pi x)\cos^2(\pi z) + 4\cos^2(\pi x)\cos^2(\pi z) + 2cos^2(\pi x)\sin^2(\pi z)\right) \\ cos(\pi z)\sin(\pi z)\left(-\sin^2(\pi x)\cos^2(\pi y) + 2\cos^2(\pi x)\sin^2(\pi y) + \cos^2(\pi x)\cos^2(\pi y)\right) \end{pmatrix}, \end{aligned} \tag{6.2.14}$$

$$w = \begin{pmatrix} \sin(\pi x)\cos(\pi y)\cos(\pi z) \\ -2\cos(\pi x)\sin(\pi y)\cos(\pi z) \\ \cos(\pi x)\cos(\pi y)\sin(\pi z) \end{pmatrix}, \tag{6.2.15}$$

and λ, a chosen parameter (e.g. $\lambda = \rho_{\mathrm{f}}/\delta t$ corresponds to the nonstationary *Oseen equations* and $\lambda = 0$ corresponds to stationary *Oseen equations*).

This model problem is complemented with the same prescribed boundary condtions (6.2.4) and (6.2.5).

The analytical solution for this model problem is given by

$$u = \begin{pmatrix} \sin(\pi x)\cos(\pi y)\cos(\pi z) \\ -2\cos(\pi x)\sin(\pi y)\cos(\pi z) \\ \cos(\pi x)\cos(\pi y)\sin(\pi z) \end{pmatrix}, \quad p = \cos(\pi x)\cos(\pi y)\cos(\pi z) + \sin(\pi)^3. \tag{6.2.16}$$

The following results will be presented:

1. relative discretization error with respect to L^2 norms and with respect to H^1 seminorms on hexahedral (see Table 6.18) and tetrahedral (see Table 6.19) meshes with different sizes of viscosity terms,

2. number of AMG iterations (see Table 6.20) and generalized minimal residual method (GMRES) iterations (see Table 6.21) with AMG preconditioning on four levels with a relative residual error reduction by a factor of 10^{-10},

As expected, from the numerical results, for the relatively large viscosity terms, we observe the $O(h^2)$ convergence rate with respect to the L^2 norms for both pressure and velocity, and $O(h)$ convergence rate with respect to the H^1 seminorms for the velocity. The PGMRES method show its robustness compared to AMG method, especially for small viscosity terms.

Table 6.18: Relative discretization errors on hexahedral meshes (see Figure 6.3(b))

Viscosity	Mesh size	$\frac{\|p-p_h\|_{L^2}}{\|p\|_{L^2}}$	$\frac{\|u-u_h\|_{L^2}}{\|u\|_{L^2}}$	$\frac{\|u-u_h\|_{H^1}}{\|u\|_{H^1}}$
$\mu = 3.5$	$h = 1/4$	0.118882	0.00173792	0.294596
	$h = 1/8$	0.0317309	0.000476144	0.138659
	$h = 1/16$	0.00648642	0.000120043	0.0682973
	$h = 1/32$	0.00368371	0.0000281371	0.0340215
$\mu = 0.35$	$h = 1/4$	0.119562	0.000244628	0.29511
	$h = 1/8$	0.0336418	0.0000810976	0.138737
	$h = 1/16$	0.00769119	0.0000225371	0.0683078
	$h = 1/32$	0.00156878	0.00000410757	0.0340228
$\mu = 0.035$	$h = 1/4$	0.119645	0.000127106	0.295167
	$h = 1/8$	0.0339258	0.0000474041	0.138747
	$h = 1/16$	0.00808537	0.000014156	0.0683092
	$h = 1/32$	0.00167265	0.00000234046	0.034023
$\mu = 0.0035$	$h = 1/4$	0.119654	0.000121127	0.295173
	$h = 1/8$	0.0339557	0.000045149	0.138748
	$h = 1/16$	0.00813128	0.00001357	0.0683094
	$h = 1/32$	0.00169354	0.00000226907	0.034023

Table 6.19: Relative discretization errors on tetrahedral meshes (see Figure 6.3(a))

Viscosity	Mesh size	$\frac{\|p-p_h\|_{L^2}}{\|p\|_{L^2}}$	$\frac{\|u-u_h\|_{L^2}}{\|u\|_{L^2}}$	$\frac{\|u-u_h\|_{H^1}}{\|u\|_{H^1}}$
$\mu = 3.5$	$h = 1/4$	90.8855	0.0603745	0.616655
	$h = 1/8$	22.3735	0.0196835	0.275424
	$h = 1/16$	5.49279	0.00538489	0.129941
	$h = 1/32$	1.2714	0.00125962	0.0630224
$\mu = 0.35$	$h = 1/4$	90.0086	0.072174	0.651908
	$h = 1/8$	22.0859	0.0264684	0.314006
	$h = 1/16$	5.27848	0.00816906	0.153951
	$h = 1/32$	1.14615	0.00175229	0.0680598
$\mu = 0.035$	$h = 1/4$	89.9198	0.0738805	0.658144
	$h = 1/8$	22.0496	0.0281785	0.333278
	$h = 1/16$	5.21858	0.00952963	0.182958
	$h = 1/32$	1.09502	0.00213492	0.0807443
$\mu = 0.0035$	$h = 1/4$	89.9109	0.0740605	0.658835
	$h = 1/8$	22.0457	0.0283963	0.336854
	$h = 1/16$	5.21048	0.00978133	0.197043
	$h = 1/32$	1.08599	0.00225214	0.092291

Table 6.20: Number of AMG iterations on mesh of four levels (see Figure 6.1)

Viscosity	Mesh size	Hexahedral mesh	Tetrahedral mesh
$\mu = 3.5$	$h = 1/4$	15	17
	$h = 1/8$	8	8
	$h = 1/16$	5	9
	$h = 1/32$	6	14
$\mu = 0.35$	$h = 1/4$	26	20
	$h = 1/8$	28	19
	$h = 1/16$	14	10
	$h = 1/32$	6	7
$\mu = 0.035$	$h = 1/4$	26	24
	$h = 1/8$	69	50
	$h = 1/16$	58	39
	$h = 1/32$	26	18
$\mu = 0.0035$	$h = 1/4$	29	24
	$h = 1/8$	70	64
	$h = 1/16$	>80	>80
	$h = 1/32$	>80	60

Table 6.21: Number of PGMRES iterations on mesh of four levels (see Figure 6.1)

Viscosity	Mesh size	Hexahedral mesh	Tetrahedral mesh
$\mu = 3.5$	$h = 1/4$	14	11
	$h = 1/8$	11	11
	$h = 1/16$	8	10
	$h = 1/32$	9	14
$\mu = 0.35$	$h = 1/4$	16	13
	$h = 1/8$	22	16
	$h = 1/16$	15	13
	$h = 1/32$	10	10
$\mu = 0.035$	$h = 1/4$	16	12
	$h = 1/8$	29	23
	$h = 1/16$	31	24
	$h = 1/32$	22	18
$\mu = 0.0035$	$h = 1/4$	17	12
	$h = 1/8$	30	24
	$h = 1/16$	44	38
	$h = 1/32$	43	36

6.3 Numerical results for the FSI problem

We firstly set material parameters. The Lamé constants are set by $\mu^l = 1.15 \times 10^6$ and $\lambda^l = 1.73 \times 10^6$, and the structure density by $\rho^s = 1.2$. The fluid viscosity is set by $\mu = 0.035$, and the fluid density by $\rho^{\mathrm{f}} = 1.0$.

The structure is considered linear and clamped at both inlet and outlet. The fluid and structure are initially at rest. For the fluid, we set the Neumann data of $(1.332 \times 10^4, 1.332 \times 10^4, 1.332 \times 10^4)^T$ on the inlet for a time period of 3ms and $(0, 0, 0)^T$ after 3ms, and the Neumann data of $(0, 0, 0)^T$ for the outlet for all time.

Two meshes (see Figure 6.4.) are used for testing the FSI simulation.

We use a time step size of $\delta t = 1$ms and run the simulation until time $t = 25$ms. For visualization purpose the deformation is amplified by a factor of 12.

The iteration numbers are reported in the following:

1: a relative error reduction by a factor of 10^{-5} is achieved in 2-3 Newton iterations in each time step,
2: each of these iterations requires 6-8 preconditioned GMRES iterations for a relative error reduction by a factor of 10^{-5} (in addition, it also requires applying the AMG solvers for the structure and fluid sub-problems once for updating the residual),
3: for each preconditioned GMRES iteration, we need to apply the AMG solver for the structure sub-problem twice (because we use the preconditioner from the structure part for each GMRES iteration) and the AMG solver for the fluid sub-problem once,
4: for solving the structure sub-problem, about 10 preconditioned conjugate gradient iterations with the AMG preconditioner are needed for a relative error reduction by a factor of 10^{-8},
5: for the fluid sub-problem about 5 AMG iterations for a relative error reduction by a factor of 10^{-8}.

Almost the same numbers of iterations are observed for the coarse and fine meshes.

For the doubled time step size $\delta t = 2$ms the number of inner AMG iterations for the fluid problem increased to about 10.

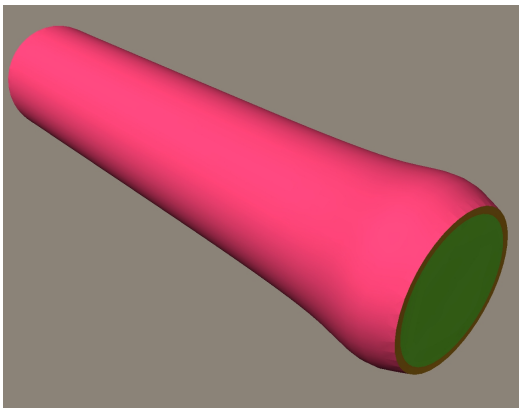
The pressure wave propagation on the fine mesh (see Figure 6.4(b)) at different time levels is visualized in Figure 6.8.

6.4 FSI simulations on grid computing

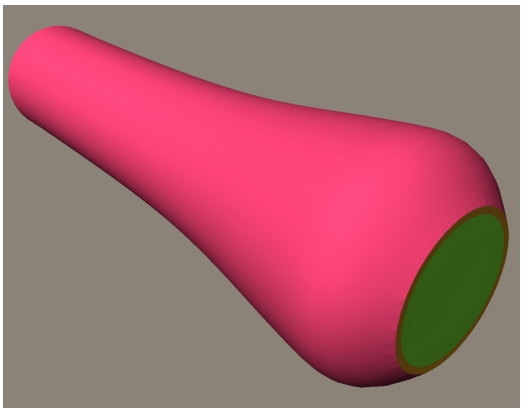
As we might see today, using the grid computing resources (see [58, 51, 57, 41, 125, 112, 160, 28]), the large scale numerical simulations can be performed in a more efficient way because of its huge computational and memory storage resources, e.g. see [173, 102, 98] for our previous work on computational fluid dynamics (CFD), and see [97, 99] on fluid-structure interaction problem. In this section, we develop a framework for designing a grid-enabled solver using the numerical method presented in Chapter 3 for solving the FSI problem.

This numerical method requires efficient, robust and fast solvers for the sub-problems (fluid and structure), which are the main costs of this type of algorithms and can be distributed and parallelized to many processors under the grid environment. Therefore it is well suitable for grid computing.

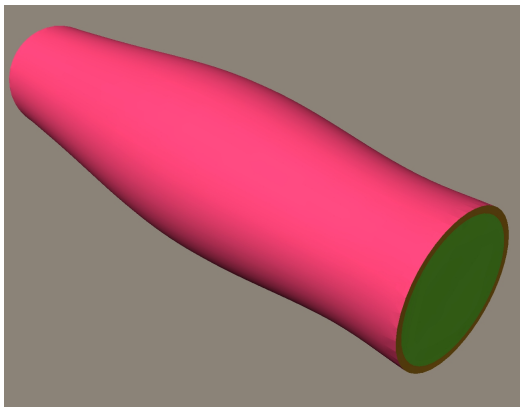
We employ three distinct grid nodes. The master node is responsible for gathering and redistributing data, and synchronizing the process in each nonlinear iteration, while the



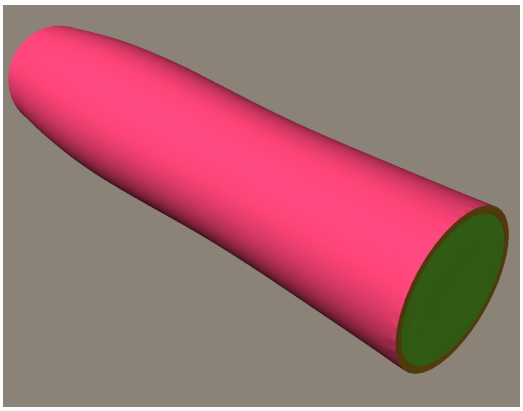
(a) $t = 1$ms



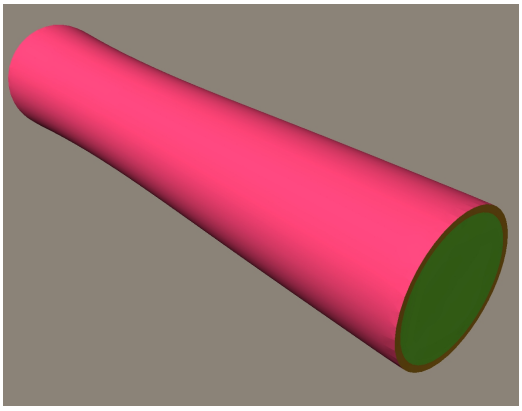
(b) $t = 5$ms



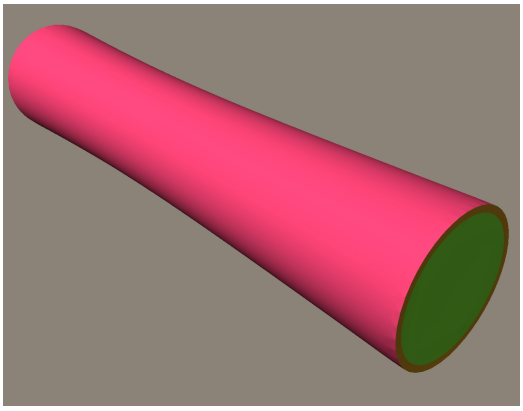
(c) $t = 10$ms



(d) $t = 15$ms



(e) $t = 20$ms



(f) $t = 25$ms

Figure 6.8: Simulation results at time $t = 1$ms (upper left), $t = 5$ms (upper right), $t = 10$ms (middle left) and $t = 15$ms (middle right), $t = 20$ms (lower left) and $t = 25$ms (lower right).

other two slave nodes will be responsible for solving the structure and fluid sub-problems respectively, see Figure 6.9 for an illustration.

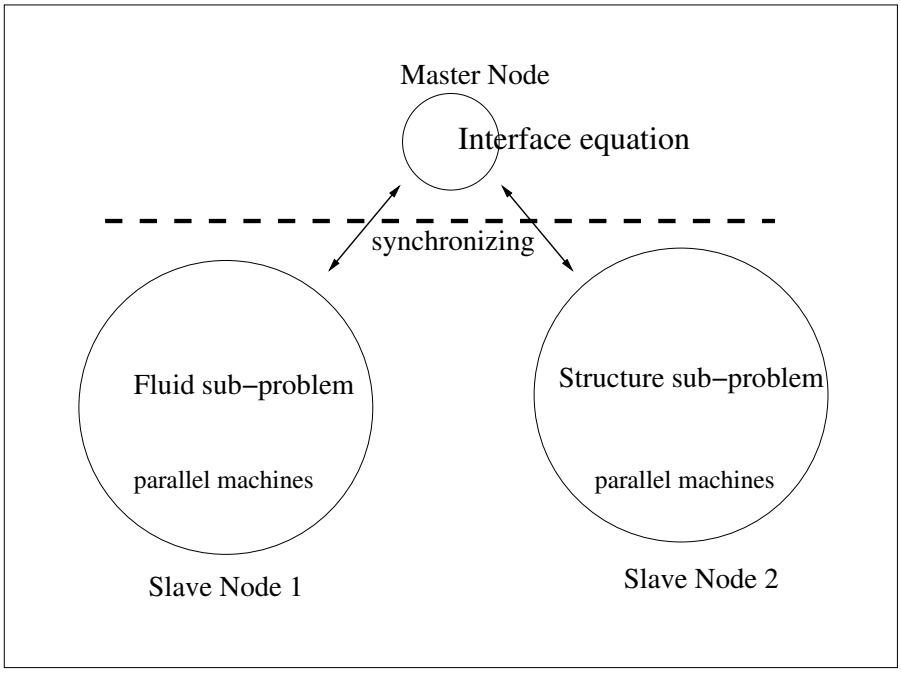


Figure 6.9: A grid-enabled solver model.

By adapting the Newton algorithm (see Algorithm 3.4) to the grid computing environment, only a small amount of data (the updated displacement at the interface and the normal stresses at the interface) will be transferred among the master and slave grid nodes during the process. Hence most of the simulation time is spent in solving the structure and the fluid sub-problems on corresponding slave grid nodes. In principle, each of these sub-problems can be solved in parallel on each node.

The grid-enabled fluid-structure interaction solver is realized by extending the idea of constructing a grid-enabled Client/Server (CS) model described in [173, 102] where Globus_IO secure channels (see [53]) are employed for creating a flexible and secure data transferring interface on the memory level for each grid node.

In the following sections, we will present details concerning how to construct the CS model under the grid computing environment, which is applied to the fluid-structure interaction simulation. Finally, we will report some test results concerning the computational complexity of the master and slave nodes, and the comparison of the computational and communicational cost.

However, this is only the first draft implementation which does not want to achieve optimal performance and thus serves as a proof of concept. Starting from this general framework, we can go further. For instance, although we distribute the structure and the fluid sub-problems on two different grid nodes such that both can start their own job in parallel, each sub-problem itself is not parallelized, i.e. we do not fully utilize the computing and memory resources on slave nodes. Future work will concentrate on improving the

performance and investigating the scalability of the method, see [19, 47].

6.4.1 The secure grid environment

One important point concerning grid computing is how to realize secure data transfer among client and server nodes through Internet/Intranet. The Globus Toolkit 4.0.4 includes the open source software MyProxy 3.7 for managing security credentials (certificates and private keys, see [4]). One highlight of this package is to combine an online credential repository with an online certificate authority which allow users to obtain credentials when and where needed. Under the Austrian grid environment, the user would use *myproxy-init* command to upload a credential to the myproxy-server *hydra.gup.uni-linz.ac.at* for later retrievals by CS nodes, e.g., under the Austrian Grid environment, we employ *altix1.jku.austriangrid.at* and *Schafberg.coma.sbg.ac.at* as client and server nodes, respectively. The credential is then delegated to the myproxy-server and stored with the given MyProxy passphrase. Proxy credentials with default lifetime of 12 hours can then be retrieved via *myproxy-get-delegation* with the MyProxy passphrase. Once the CS nodes obtain the proxy credentials, the authentication and authorization on the CS are done. The secure mode is verified via setting Globus_IO secure mode parameters as input arguments of the functions: *globus_io_attr_set_secure_authentication/channel_mode*. See Figure 6.10 for an illustration.

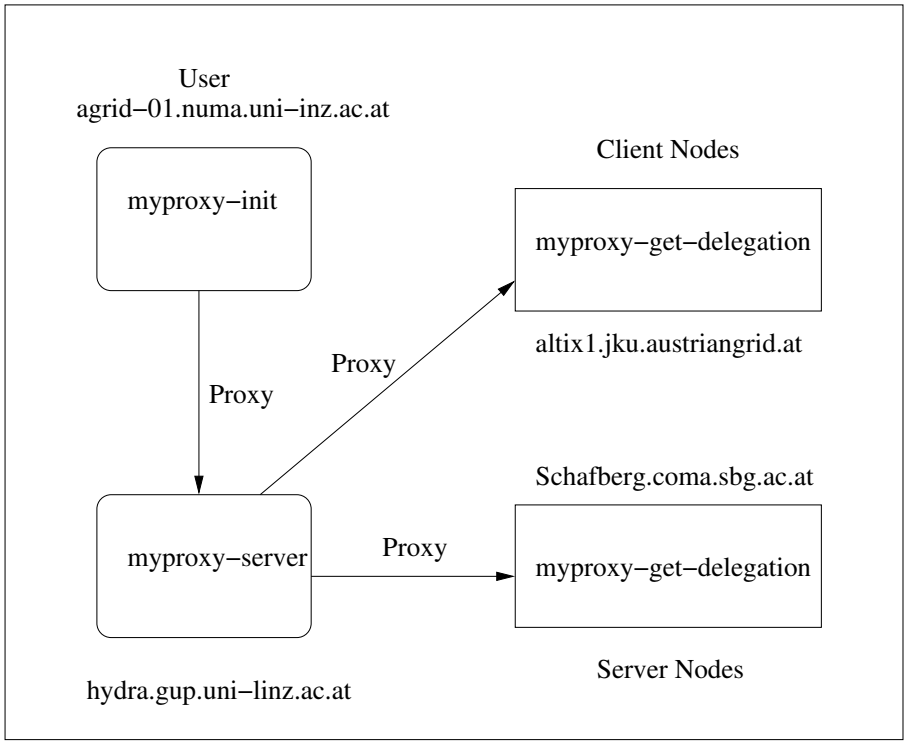


Figure 6.10: MyProxy process on the Austrian grid

Using the previous authentication and authorization, a secure channel connecting Client and Server nodes through the Internet/Intranet is provided by using the Globus_IO secure channel. It provides high-performance I/O with integrated security and a socket-like interface for users (see [53]). Normal users holding no powerful machines can also succeed in doing

such numerical simulations. As shown in Figure 6.11, once their identities are certified by a Certification Authority and recognized by the requested resources, users can submit the job to nodes and control the data flow between nodes. For instance, *globus-url-copy* (see

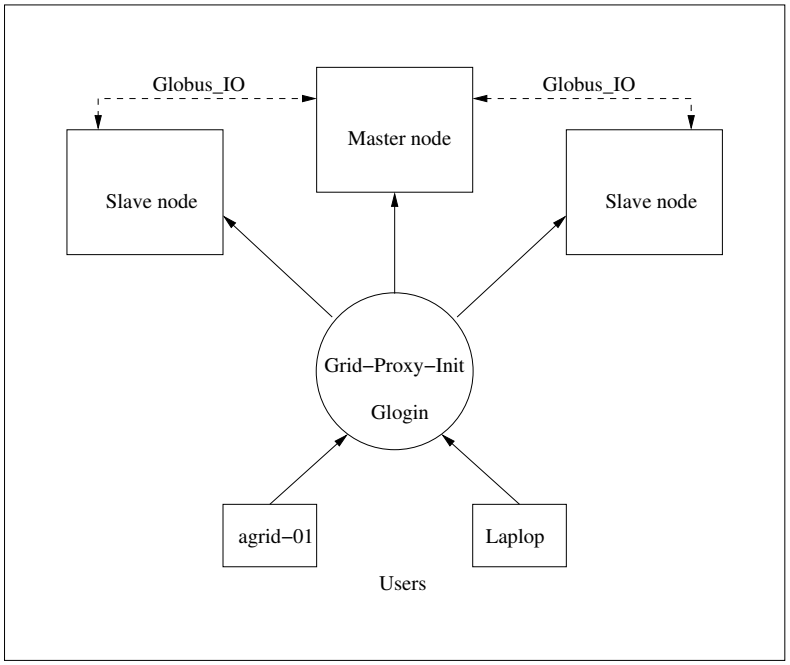


Figure 6.11: A secure channel in the Internet/Intranet

[2]), can realize the data files transfer among nodes, and by RSL (resource specification language), see [2], users can control the job running schedule on the nodes. Under the Austrian grid environment, using *glogin*, the identified user can realize the interactive usage of grid resources (see [134]). Via the Globus_IO secure and efficient channel mode, we can distribute our task to different nodes such that they are able to cooperate with each other. The communication between nodes is guaranteed in this channel created by calling *globus_io_tcp_connect*.

6.4.2 The grid-enabled Client/Server model

An additional feature compared to usual CS models on TCP/IP protocols is the authentication part on both client and server nodes by employing Globus_IO operations. The TCP connecting is mainly implemented via functions of *globus_io_tcp_listen*, *globus_io_tcp_connect* and *globus_io_tcp_accept*. The master node firstly creates two TCP listeners at two ports and keeps listening at these ports. Once it gets a notification from the two slave nodes, it will try to establish TCP connections. If this connecting is successful, they will continue executing the next jobs. Otherwise, the master node still listens at the opening ports until it gets next notifications from the slave nodes. The data transfer and redistribution among nodes are hidden in the box of FSI solver for each time step in Figure 6.12. Since this is a time dependent problem, this FSI solver has to be called at each time step. When the iteration ends at the final time step, all I/O operations and TCP connections on the nodes are closed.

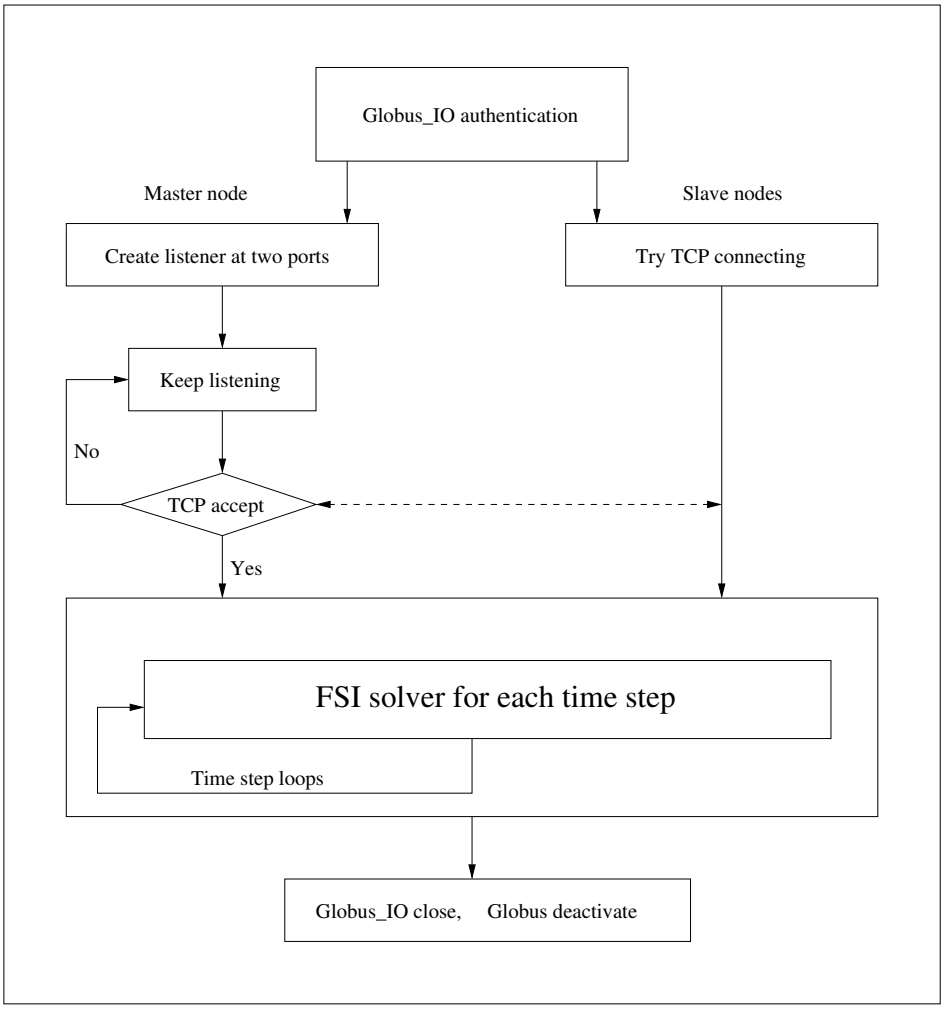


Figure 6.12: Grid-enabled CS model for the FSI simulation

For each time step, we will apply several Newton iterations (Algorithm 3.4.1) as described in Section 3.4. For convenience of reading, we rewrite it here:

From these steps in Algorithm 6.4.1, it is easy to realize a grid-enabled version of the Newton Algorithm (see Algorithm 6.4.2). Note that if a preconditioner, e.g. S_{s}^{-1}, is applied to the linearized problem in Step 2 of Algorithm 6.4.2 (preconditioned GMRES (PreGMRES)), some additional communication among nodes are needed. We also mention that in Step 2 of Algorithm 6.4.2, the GMRES iteration needs the operations of $S_{\mathrm{s}}'(\lambda^k)\delta\lambda^k$ and $S_{\mathrm{f}}'(\lambda^k)\delta\lambda^k$ which are the main cost of the GMRES iteration and done on the slave nodes.

6.4.3 Shared data transferring interface

The communication between master and slave nodes are mainly focus on delivering a small amount of data (vector values of the displacement and the stress at the interface). A secure, stable and efficient data transferring interface has to be constructed on both master and slave nodes. As we mentioned before, the Globus_IO offers such desirable operations. By calling

Algorithm 6.4.1 Newton iterations

For $k \geq 0$,

1: update the residual $S_{\mathrm{s}}(\lambda^k) + S_{\mathrm{f}}(\lambda^k)$ by solving the structure and fluid sub-problems,

2: solve the linear problem $\left(S'_{\mathrm{s}}(\lambda^k) + S'_{\mathrm{f}}(\lambda^k)\right)\delta\lambda^k = -\left(S_{\mathrm{s}}(\lambda^k) + S_{\mathrm{f}}(\lambda^k)\right)$ via GMRES method,

3: update the displacement $\lambda^{k+1} = \lambda^k + \delta\lambda^k$, if not accurate enough, go to step 1.

Algorithm 6.4.2 Grid-enabled Newton iterations

1: distribute displacement λ^k at the interface from the master node to slave nodes, update the residual $S_{\mathrm{s}}(\lambda^k)$ and $S_{\mathrm{f}}(\lambda^k)$ by solving the structure and fluid sub-problems independently on corresponding slave nodes, send back the results to the master node, and calculate $S_{\mathrm{s}}(\lambda^k) + S_{\mathrm{f}}(\lambda^k)$ on the master node,

2: solve the linearized problem $\left(S'_{\mathrm{s}}(\lambda^k) + S'_{\mathrm{f}}(\lambda^k)\right)\delta\lambda^k = -\left(S_{\mathrm{s}}(\lambda^k) + S_{\mathrm{f}}(\lambda^k)\right)$ via GMRES method on the master node, i.e. update $S'_{\mathrm{s}}(\lambda^k)\delta\lambda$ and $S'_{\mathrm{f}}(\lambda^k)\delta\lambda$ independently on corresponding slave nodes, send them back to the master node, and update $\left(S'_{\mathrm{s}}(\lambda^k) + S'_{\mathrm{f}}(\lambda^k)\right)\delta\lambda^k$ for each GMRES iteration,

3: update the displacement $\lambda^{k+1} = \lambda^k + \delta\lambda^k$ on the master node, and go to step 1 if not accurate enough

the *globus_io_write/read* pair, one can realize the data sending and receiving through the established Globus_IO channel. Usually, hierarchical data structures of vectors are used for storing the vector data. The vectors could also contain simple data structures. The size of these hierarchical data structures has to be measured carefully since the *globus_io_write/read* calling needs to know the exact block size of the message the nodes will send and receive. The *globus_io_write/read* callings are encapsulated inside such that the synchronizing process will be guaranteed for each sending and receiving pair.

6.4.4 Client/Server configuration files

Using the high-performance and secure data transferring protocol GridFTP (see [2]), the executable binary files and necessary configuration files are transferred from a user machine (*agrid-01*) to client and server grid nodes (*altix1.uibk.ac.at*, *alex.jku.austriangrid.at*, *lilli.edvz.uni-linz.ac.at*). Once the binary files are installed on the grid nodes, one can specify the grid node roles with the help of configuration files. See Figure 6.13.

For the configuration on the server node (*alex.jku.austriangrid.at*), we need to specify the grid node type (*GRIDTYPE Gserver*), the name of the grid node (*GRIDNODE alex.jku.austriangrid.at*), and the port numbers opening for two client nodes (*GRIDPORT* $44106, 44400$). These parameters tell the node it should play the server role. Some additional parameters have to be specified in order to distribute and gather data from client nodes.

Two client nodes are used in this model. For the fluid part, the grid node *lilli.edvz.uni-linz.ac.at* is chosen as a client node. We set the grid node type (*GRIDTYPE GClientFluid*), the server node which we want to connect to (*GRIDNODE alex.jku.austriangrid.at*), and the port number as one of port numbers specified on the server node (*GRIDPORT 44400*). In analogous way, for the structure part, the grid node (*altix1.uibk.ac.at*) is chosen as a client

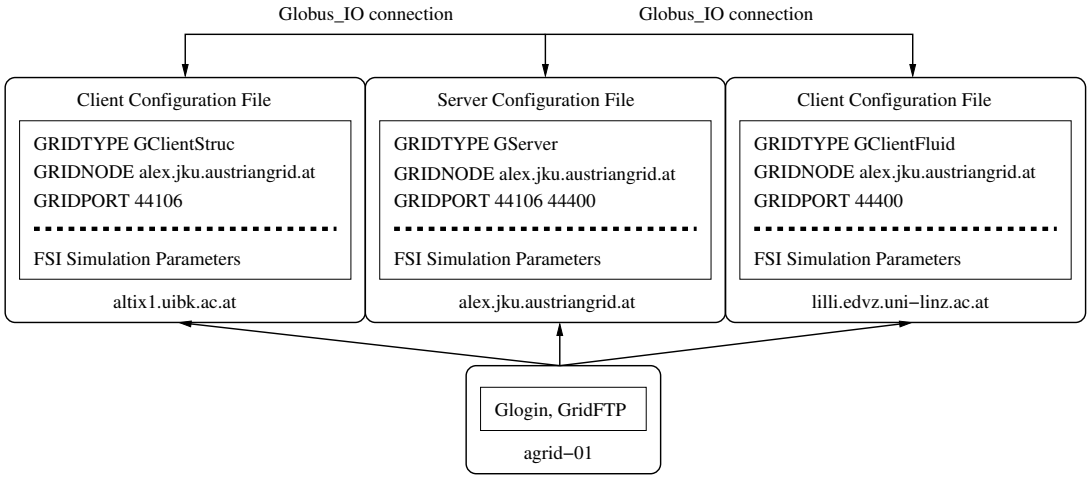


Figure 6.13: CS configuration files

node. We set the grid node type (*GRIDTYPE GClientStruc*), the server node which we want to connect to (*GRIDNODE alex.jku.austriangrid.at*), and the port number as one of port numbers specified on the server node (*GRIDPORT 44106*). For both client nodes, we need to set some FSI simulation parameters such that each of them will be responsible for its own job, i.e. solving the structure and the fluid sub-problems, respectively.

These configuration files will be transferred to corresponding grid nodes, and act as running input parameters when starting the jobs on three grid nodes.

6.4.5 Experiments on the Austrian Grid

Testing environment

We use three different nodes for testing the algorithm:

1: the Server (master) node for synchronizing the whole process, *alex.jku.austriangrid.at* in Linz, which is a cluster with a total of 384 Xeon (nehalem) cores connected via gigabit,
2: the Client (slave) node for running the structure solver, *altix1.uibk.ac.at* in Innsbruck, which is a cluster of four 16-way SGI Altix 350 systems interconnected by an Infiniband fabric,
3: the Client (slave) node for running the fluid solver, *lilli.edvz.uni-linz.ac.at* in Linz, which is a shared-memory single node with 256 CPUs and 1 TB RAM, connected via gigabit.

See grid node information in Table 6.22 and [20].

Table 6.22: Grid Nodes Information

Site	Grid Node	Processors Types (#processors)	RAM
ALTIX-UIBK	altix1.uibk.ac.at	Intel Itanium-2 (16)	1.5 GB
JKU	lilli.edvz.uni-linz.ac.at	Intel Itanium-2 (256)	1.0 TB
JKU	alex.jku.austriangrid.at	Xeon (nehalem) Cores (768)	1.5 TB

In addition, the user machine *agrid-01* in Linz is a desktop with two AMD Opteron processors and 4 GB RAM, which is responsible for transferring data and binary files from the user to the grid nodes.

However, as mentioned before, for this moment, neither the structure nor the fluid solver is parallelized, so we only utilize one of processors from each grid node. In order to obtain good scalability and high performance, we may implement the structure and fluid solvers in parallel for the future plan, e.g. see the parallel technique in [47, 18].

Computational complexity

As we see in Algorithm 6.4.2, the main cost is solving the structure and the fluid sub-problems on client grid nodes.

We will report the cost (measured in second (s)) for one preconditioned GMRES iteration (PGMRES) which involves the cost for the structure solver (S-cost), for the fluid solver (F-cost), and for the preconditioning from the structure part (P-cost), see Table 6.23.

Table 6.23: Computational cost on the coarse and fine mesh (see Figure 6.4)

Mesh	PGMRES	S-cost (Solver)	F-cost (Solver)	P-Cost (Solver)
coarse	$36.45s$	$2.79s(2.75s)$	$10.53s(9.46s)$	$23.12s(23.05s)$
fine	$162.2s$	$10.32s(10.23s)$	$92.89s(68.49s)$	$59.00s(58.76s)$

In Table 6.23, we report the following cost:

1: PGMRES, the cost for each preconditioned GMRES iteration,
2: S-cost (Solver), the overall cost for solving the structure sub-problem and transferring the data from the structure slave grid node to the master grid node, and the cost for the PCG with AMG preconditioning (in the bracket),
3: F-cost (Solver), the overall cost for solving the fluid sub-problem and transferring the data from the fluid slave grid node to the master grid node, and the cost for the AMG solver (in the bracket),
4: P-cost (Solver), the overall cost for preconditioning and transferring the data from the structure slave grid node to the master grid node, and the cost for the PCG with AMG preconditioning (in the bracket).

The cost for each Newton iteration is obtained by multiplying the cost for each PGMRES with 6-8, and the cost for each time step is obtained by multiplying the cost for each Newton Iteration with 2-3.

Chapter 7

Conclusions and future work

One of the main contributions we have investigated in this thesis is the generalization of linear elements to extended elements on hybrid meshes for the structure and fluid sub-problems. It has been applied to the fluid-structure interaction problem which is solved by a Newton based solver on the Steklov-Poincaré operator. However, there are alternative strategies which can be considered in the future. For instance the discontinuous Galerkin method could be a good alternative for constructing the finite elements on such hybrid meshes.

We are able to apply AMG methods to the finite element equations for structure and fluid sub-problems of the fluid-structure interaction problem. Here we extend the AMG method for the saddle point problem arsing from the discretization of the MINI-element on hybrid meshes. The extension of AMG applications to other elements might be interesting for the future work. On the other hand, a robust AMG solver for relatively small viscosity terms are still on studying, in particular under the ALE framework.

For the time being, we are only using a simple linear elasticity model for the structure sub-problem. For the structure model in life science, for instance, blood flow simulation, the material and geometric nonlinearity will be considered in the future. Current research in this direction can be found, for instance in [78, 79].

Finally, we also developed a grid-enabled solver under the Austrian Grid environment by constructing a Client/Server model. It was shown that the concept is working. However, this is only the first draft implementation. The future work in this direction could concentrate on improving the performance and investigating the scalability of the method, see [30].

Bibliography

[1] Austrian Grid webside. http://www.austriangrid.at/.

[2] The Globus Toolkit webside. http://www.globus.org/toolkit/.

[3] Grid computing webside. http://en.wikipedia.org/wiki/Grid_computing/.

[4] The MyProxy webside. http://grid.ncsa.uiuc.edu/myproxy/.

[5] Polynomial approximation of functions in sobolev spaces. *Math. Comp.*, 34(150):441–463, 1980.

[6] M. F. Adams. Algebraic multigrid methods for constrained linear systems with applications to contact problems in solid mechanics. *Numer. Linear Algebra Appl.*, 00(1-6):1–13, 2000.

[7] R. A. Adams and J. J. F. Fournier. *Sobolev Spaces*, volume 140 of *Pure and Applied Mathematics*. Academic Press, Amsterdam, Boston, second edition, 2003.

[8] H. T. Ahn. *A New Incompressible Navier-Stokes Method with General Hybrid Meshes and Its Application to Flow/Structure Interactions*. PhD thesis, Offshore Technology Research Center, Texas A&M University, 2005.

[9] K. Arrow, L. Hurwicz, and H. Uzawa. *Studies in nonlinear programming*. Stanford University Press, Standford, CA, 1958.

[10] S. Artlich and W. Mackens. Newton-coupling of fixed point iterations. In *Numerical treatment of coupled systems, (ed. by W. Hackbush and G. Wittum), Notes on Numerical Fluid Mechanics, vol. 51*, pages 1–10, Wiesbaden, 1995. Vieweg.

[11] G. P. Astrachancev. An iterative method for solving elliptic net problems. *USSR Comput. Math. Math. Phys.*, 11(2):171–182, 1971.

[12] O. Axelsson. *Iterative solution methods*. Cambridge University Press, second edition, 1996.

[13] F. Baaijens. A fictitious domain/mortar element method for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 35:743–761, 2001.

[14] N. S. Bachvalov. On the convergence of a relaxation method with natural constraints on the elliptic operator. *USSR Comput. Math. Math. Phys.*, 6(5):101–135, 1966.

[15] S. Badia, F. Nobile, and C. Vergara. Fluid-structure partitioned procedures based on Robin transmission conditions. *Journal of Computational Physics*, 227:7027–7051, 2008.

[16] S. Badia, A. Quaini, and A. Quarteroni. Splitting methods based on algebraic factorization for fluid-structure interaction. *SIAM Journal of Scientific Computing*, 30(4):1778–1805, 2008.

[17] A. T. Barker and X-C. Cai. Scalable parallel methods for monolithic coupling in fluid-structure interaction with application to blood flow modeling. *Preprint submitted to Journal of Computational Physics*, 2009.

[18] P. Bastian. *Parallele adaptive Mehrgitterverfahren.* PhD thesis, Universität Heidelberg, 1994.

[19] P. Bastian. *Parallele adaptive Mehrgitterverfahren.* Teubner Skripten zur Numerik. Teubner-Verlag, 1996.

[20] M. Baumgartner, C. Glasner, and J. Volkert. An Overview of the Austrian Grid Infrastructure. In J. Volkert, T. Fahringer, D. Kranzlmüller, and W. Schreiner, editors, *Proceedings of* 1^{st} *Austrian Grid Symposium*, pages 277–286, 2006.

[21] M. Benzi and G. H. Golub. Numerical solution of saddle point problems. *Acta Numerica*, pages 1–137, 2005.

[22] S. Bertling. Ein algebraisches mehrgitterverfahren zur lösung der stokes.gleichungen. Master's thesis, RWTH, Aachen, 2002.

[23] M. Braack, E. Burman, V. John, and G. Lube. Stabilized finite element methods for the generalized Oseen problem. *Comput. Method Appl. Mech. Engrg.*, 196:853–866, 2007.

[24] D. Braess. Towards algebraic multigrid for elliptic problems of second order. *Computing*, 55:379–393, 1995.

[25] D. Braess. *Finite Elements. Theory, fast solvers, and applications in solid mechanics.* Springer, Berlin, Heidelberg, New York, 2nd edition, 2001.

[26] D. Braess and W. Hackbusch. A new convergence proof for the multigrid method including the V-cycle. *SIAM. J. Numer. Anal.*, 20:967–975, 1983.

[27] D. Braess and R. Sarazin. An efficient smoother for the Stokes problem. *Appl. Numer. Math.*, 23:3–20, 1997.

[28] G. Brajesh and L. Shilpa. *Grid Revolution: An Introduction to Enterprise Grid Computing.* McGraw-Hill/Osborne Media, 2006.

[29] J. H. Bramble. *Multigrid methods.* Longman Scientific and Technical, Harlow, 1993.

[30] D. Brands, A. Klawoon, O. Rheinbach, and J. Schröder. Modelling and convergence in arterial wall simulations using a parallel FETI solution strategy. *Computer Methods in Biomechanics and Biomedical Engineering*, 11(5):569–583, 2008.

[31] A. Brandt. Algebraic multigrid theory: The symmetric case. *Appl. Math. Comput.*, 19:23–56, 1986.

[32] A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In *Sparsity and its applications*, pages 257–284. Cambridge University Press, Cambridge, 1984.

[33] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 2nd edition, 2002.

[34] M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteufel, S. F. McCormick, and J. W. Ruge. Algebraic multigrid based on element interpolation (AMGe). *SIAM J. Sci. Comput.*, 22(5):1570–1592, 2000.

[35] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer series in computational mathematics. Springer, New York, 1991.

[36] A. N. Brooks and T. J. R. Hughes. Streamline upwind Petrov-Galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32:199–259, 1982.

[37] M. Buchberger. *Biomechanical Modelling of the Human Eye*. PhD thesis, Johannes Kepler University of Linz, Institute for Applied Knowledge Processing (FAW), Linz, Austria, 2004.

[38] P. Causin, J. F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithm for fluid-structure problems. *Comp. Meth. Appl. Mech. Eng.*, 194:4506–4527, 2005.

[39] M. Cervera, R. Codina, and M. Galindo. On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems. *Engrg. Comput.*, 13(6):4–30, 1996.

[40] P. G. Ciarlet. *The finite element method for elliptic problems*, volume 40 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, reprint of the 1978 original edition, 2002.

[41] F. Craig and J. Joshy. *Grid Computing*. IBM Press, 2003.

[42] S. Deparis. *Numerical analysis of axisymmetric flows and methods for fluid-structure interaction arising in blood flow simulation*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2004.

[43] S. Deparis, M. Discacciati, G. Fourestey, and A. Quarteroni. Fluid-structure algorithms based on Steklov-Poincaré operators. *Comput. Methods Appl. Mech. Engrg.*, 195:5797–5812, 2006.

[44] S. Deparis, M. Fernández, and L. Formaggia. Acceleration of a fixed point algorithm for fluid-structure interation using transpiration conditions. *M2AN*, 37(4):606–616, 2003.

[45] H. A. Van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear system. *SIAM J. Sci. Stat. Comput.*, 13(2):631–644, 1992.

[46] J. Donea. An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *Comput. Methods. Appl. Mech. Eng.*, 33:680–723, 1982.

[47] C. Douglas, G. Haase, and U. Langer. *A Tutorial on Elliptic PDE Solvers and their Parallelization*, volume 16 of *SIAM Series on Software, Environments, and Tool.* Philadelphi, PA, 2003.

[48] 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. *Comput. Methods. Appl. Mech. Eng.*, 157(1-2):95–114, 1998.

[49] C. Farhat, K. G. v. d. Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Comput. Methods Appl. Mech. Engrg.*, 195:1973–2001, 2006.

[50] R. P. Feforenko. A relaxation method for solving elliptic difference equations. *USSR Comput. Math. math. Phys.*, 1(5):1092–1096, 1961.

[51] C. Fellenstein and J. Joseph. *Grid Computing.* Prentice Hall Ptr, 2003.

[52] M. A. Fernández and M.Moubachir. A Newton method using exact Jacobians for solving fluid-structure coupling. *Comput. and Struct.*, 83(2-3):127–142, 2005.

[53] L. Ferreira, V. Berstis, and J. Armstrong. *Introduction to Grid Computing with Globus.* IBM Corp, 2003.

[54] C. A. Figueroa, I. E. V. Clementel, K. E. Jansen, T. J. R. Hughes, and C. A. Taylor. A coupled momentum method for modeling blood flow in three-dimensional deformable arteries. *Computer Methods in Applied Mechanics and Engineering*, in press.

[55] L. Formaggia and F. Nobile. A stability analysis for the arbitrary Lagrangian Eulerian formulation with finite elements. *East-West Journal of Numerical Mathematics*, 7:105–132, 1999.

[56] C. Förster. *Robust methods for fluid-structure interaction with stabilised finite elements.* PhD thesis, University Stuttgart, 2007.

[57] I. Foster and C. Kesselman. *Grid 2 Blueprint for a New Computing*, volume 13 of *Elsevier Series in Grid Computing.* Morgan Kaufmann Pub, 2003.

[58] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *High Performance Computing Applications*, 15(3):200–222, 2001.

[59] G. Fourestey and S. Piperno. A second-order time-accurate ALE Lagrange-Galerkin method applied to wind engineering and control of bridge profiles. *Comput. Methods. Appl. Mech. Engrg.*, 190:3147–3170, 2001.

[60] L. P. Franca, T. J. R. Hughes, and R. Stenberg. Stabilized finite element methods for the Stokes problem. In M. Gunzburger and R. A. Nicolaides, editors, *Incompressible Computational Fluid Dynamics*, chapter 4, pages 87–107. Cambridge University Press, 1993.

[61] L. P. Franca and R. Stenberg. Error analysis of some Galerkin least squares methods for the elasticity equations. *SIAM. J. Numer. Anal.*, 28(6):1680–1697, 1991.

[62] K. O. Friedrichs. Differential forms on Riemannian manifolds. *Comm. Pure Appl. Math.*, 8:551–590, 1955.

[63] Z. M. Gao, Y. C. Ma, and H. W. Zhang. Optimal shape design for the time-dependent Navier-Stokes flow. *Int. J. Numer. Meth. Fluids*, 57:1505–1526, 2008.

[64] Z. M. Gao, Y. C. Ma, and H. W. Zhang. Shape optimization for Navier-Stokes flow. *Inverse Problems in Science and Engineering*, 16(5):583–616, 2008.

[65] J. Gerbeau and M. Vidrascu. A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *M2AN*, 37(4):663–680, 2003.

[66] J. F. Gerbeau, M. Vidrascu, and P. Frey. Fluid-structure interaction in blood flows on geometries coming from medical imaging. Technical-Report 5052, INRIA, 2003.

[67] V. Girault and P. A. Raviart. *Finite Element Methods for Navier-Stokes Equations.* Springer, Berlin, Heidelberg, 1986.

[68] M. Griebel, T. Neunhoeffer, and H. Regler. Algebraic multigrid methods for the solutions for the Navier-Stokes equations in complicated geometries. *Int. J. Numer. Methods Fluids*, 26:1–137, 1998.

[69] M. Griebel, D. Oeltz, and M. A. Schweitzer. An algebraic multigrid method for linear elasticity. *SIAM. J. Sci. Comput.*, 25(2):385–407, 2003.

[70] M. E. Gurtin. *An introduction to continuum mechanics*, volume 158 of *Mathematics in Science and Engineering.* Academic Press Inc., New York, 1981.

[71] G. Haase and U. Langer. Multigrid methods: from geometrical to algebraic versions. In *Modern Methods in Scientific Computing and Applications (ed. by A. Bourlioux and M. J. Gander)*, pages 103–153, Netherlands, 2002. Kluwer Academic Publishers.

[72] W. Hackbusch. A fast iterative method for solving Poisson's equation in a general domain. In R. Bulirsch, R. D. Grigorieff, and J. Schröder, editors, *Numerical Treatment of Differential Equations*, pages 51–62, Berlin-Heidelberg-New York, 1976. Springer-Verlag.

[73] W. Hackbusch. *Multi-grid methods and applications.* Springer, Berlin, Heidelberg, New York, Tokyo, 1985.

[74] W. Hackbusch. *Iterative solution of large sparse systems of equations.* Springer Verlag, Berlin, Heidelberg, New York, 1993.

[75] J. De Hart, GWM. Peters, PJG. Schreurs, and FPT. Baaijens. A three-dimensional computational analysis of fluid-structure interaction in the aortic valve. *Journal of Biomechanics*, 36:103–112, 2003.

[76] M. Heil. An efficient solver for the fully coupled solution if large-displacement fluid-structure interaction problems. *Comput. Methods Appl. Mech. Engrg.*, 193(1-2):1–23, 2004.

[77] V. E. Henson and P. S. Vassilevski. Element-free AMGe: General algorithms for computing interpolation weights in AMG. *SIAM. J. Sci. Comput.*, 23(2):629–650, 2001.

[78] G. A. Holzapfel. Arterial tissue in health and disease: experimental data, collagen-based modeling and simulation, including aortic dissection. In G.A. Holzapfel and R.W. Ogden, editors, *Biomechanical Modelling at the Molecular, Cellular and Tissue Levels*, volume 606 of *CISM Courses and Lectures No. 508*, Wien, New York, 2009. Springer.

[79] G. A. Holzapfel and R. W. Ogden. *Biomechanical Modeling at the Molecular, Cellular and Tissue Levels.* CISM Courses and Lectures No. 508. Springer, Wien, New York, 2009.

[80] T. J. R. Hughes and L. P. Franca. A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: symmetric formulations that converge for all velocity/pressure spaces. *Comp. Meths. Appl. Mech. Engrg.*, 65:85–96, 1987.

[81] Y. Ito, A. M. Shih, and B. K. Soni. Hybrid mesh generation with embedded surfaces using a multiple marching direction approach. *Int. J. Numer. Meth. Fluid*, 2008. in press.

[82] C. Johnson and J. Saranen. Streamline diffusion methods for the incompressible Euler and Navier-Stokes equations. *Mathematics of computation*, 47(175):1–18, 1986.

[83] J. E. Jones and P. S. Vassilevski. AMGe based on element agglomeration. *SIAM J. Sci. Comput.*, 23(1):109–133, 2001.

[84] M. Jung and U. Langer. Application of multilevel methods to practical problems. *Surveys on Mathematics for Industry*, 1:217–257, 1991.

[85] M. Jung, U. Langer, A. Meyer, W. Queck, and M. Schneider. Multigrid preconditioners and their applications. In G. Telschow, editor, *Third Multigrid Seminar, Biesenthal 1988*, number Report R–MATH–03/89, pages 11–52, Berlin, 1989. Karl–Weierstrass–Institut.

[86] Y. Kallinderis. Hybrid grids and their applications. In *Handbook of Grid Generation*, Boca Raton, FL, 1999. CRC Pess.

[87] Y. Kallinderis and . T. Ahn. Incompressible Navier-Stokes method with general hybrid meshes. *Journal of Computational Physics*, 210:75–108, 2005.

[88] F. Kickinger. Algebraic multigrid for discrete elliptic second-order problems – a program description. Technical Report 96-5, Department of Mathematics, Johannes Kepler University of Linz, 1996.

[89] F. Kickinger. Automatic microscaling mesh generation. SFB-Report 525, Department of Mathematics, Johannes Kepler University Linz, 1997.

[90] F. Kickinger. Algebraic multigrid for discrete elliptic second-order problems. In *Multigrid Methods V. Proceedings of the 5th European Multigrid conference (ed. by W. Hackbush), Lecture Notes in Computational Sciences and Engineering, vol. 3*, pages 157–172. Springer, 1998.

[91] F. Kickinger and U. Langer. A note on the global extraction element-by-element method. *ZAMM*, (78):965–966, 1998.

[92] V. G. Korneev. *Finite element schemes of higher order of accuracy (in Russian).* Leningrad University Press, Leningrad, 1977.

[93] J. Kraus. Algebraic multigrid based on computational molecules, 1: Scalar elliptic problems. *Computing*, 77:57–75, 2006.

[94] J. Kraus. Algebraic multigrid based on computational molecules, 2: Linear elasticity problems. *SIAM J. Sci. Comput.*, 30(1):505–524, 2008.

[95] J. Kraus and S. Margenov. *Robust Algebraic Multilevel Methods and Algorithms*, volume 5 of *Radon Series Comp. Appl. Math.* de Gruyter, Berlin, New York, 2009.

[96] U. Langer. On the choice of iterative parameters in the relaxation method on a sequence of meshes. *USSR Comput. Math. Math. Phys.*, 22(5):98–114, 1982.

[97] U. Langer, T. Odaker, H. Yang, and W. Zulehner. Fluid-structure Interaction (FSI) Simulation Under the Grid Environment. Poster at the supercomputing conference '08, Austin, Texas, USA, November 15-21 2008.

[98] U. Langer and H. Yang. A parallel solver for the 3D incompressible Navier-Stokes equations on the austrian grid. SFB-Report 06-12, SFB "Numerical and Symbolic Scientific Computing" F013, Johannes Kepler University of Linz, 2006.

[99] U. Langer, H. Yang, and W. Zulehner. A grid-enabled solver for the fluid-structure interaction (FSI) problem. Technical Report 2009-07, Institute of Computational Mathematics, Johannes Kepler University of Linz, 2009. submitted.

[100] U. Langer, H. Yang, and W. Zulehner. A grid-enabled solver for the fluid-structure interaction (FSI) problem. In J. Volkert, W. Schreiner, and T. Fahringer, editors, *3. Austrian Grid Symposium*, RISC-Linz Report Serier No. 09-14, Johannes Kepler University Linz, Austria, 2009.

[101] U. Langer, H. Yang, and W. Zulehner. Numerical simulation of fluid-structure interaction problems on the grid environment. Technical Report 2009-07, Institute of Computational Mathematics, Johannes Kepler University of Linz, 2009. submitted.

[102] U. Langer, W. Zulehner, H. Yang, and M. Baumgartner. GStokes: A Grid-enabled solver for the 3D Stokes/Navier-Stokes system on hybrid meshes. *Parallel and Distributed Computing, Sixth International Symposium on Parallel and Distributed Computing (ISPDC'07)*, 0:377–382, 2007.

[103] G. Lube and G. Rapin. Residual-based higher-order FEM for a generalized Oseen problem. *Math. Model. Meth. Appl. Sci.*, 16(7):949–966, 2006.

[104] J. F. Gerbeau M. A. Fernández and C. Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *Int. J. Numer. Meth. Engng*, 69:794–821, 2007.

[105] M. Marwan and J. P. Zolesio. *Moving Shape Analysis and Control Applications to Fluid Structure Interaction*. Pure and Applied Mathematics. Chapman and Hall/CRC, Boca Raton, London, New York, 2006.

[106] I. Z. Matkó. *Grid-aware Database Support for Medical Software*. PhD thesis, International School for Informatics, Hagenberg, Austria, 2008.

[107] H. Matthies and J. Steindorf. Numerical efficiency of different partitioned methods for fluid-structure interaction. *Z. Angew. Math. Mech.*, 2(80):557–558, 2000.

[108] H. Matthies and J. Steindorf. Partitioned strong coupling algorithms of fluid-structure interaction. *Comput. Struct.*, 81:805–812, 2003.

[109] H. G. Matthies and J. Steindorf. Partitioned but strongly coupled iteration schemes for nonlinear fluid-structure interaction. *Comput. Struct*, 80:1991–1999, 2002.

[110] S. F. McCormick. An algebraic interpolation of multigrid methods. *SIAM. J. Numer. Anal.*, 19(3):548–560, 1982.

[111] G. Meurant. *Computer solution of large linear systems*, volume 28 of *Studies in Mathematics and its Applications*. Elsevier, 1999.

[112] D. S. Michael. *Distributed Data Management for Grid Computing*. Wiley-Interscience, 2005.

[113] D. Mittendorfer. *Grid-Capable Persistence Based on a Metamodel for Medical Decision Support*. PhD thesis, University of Applied Sciences, Dept. of Software Engineering, Hagenberg, Austria, 2005.

[114] D. P. Mok and W. A. Wall. Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. In *Trends in Computational Structural Mechanics (ed. by K. Schweizerhof, W. Wall)*, Barcelona, 2001. K. U. Bletzinger, CIMNE.

[115] D. P. Mok, W. A. Wall, and E. Ramm. Accelerated iterative substructing schemes for instationary fluid-structure interaction. In *Computational fluid and solid mechanics, vol. 1,2 (Cambridge, MA, 2001)*, pages 1325–1328, Amsterdam, 2001. Elsevier.

[116] H. Morand and R. Ohayon. *Numerical Mathematics*. John Wiley and Sons, Chichester, 1995.

[117] W. W. Nichols and M. F. O'Rouke. *McDonald's Blood Flow in Arteries. Theoretical, experimental and clinical principles.* Arnold, London, 1998.

[118] F. Nobile. *Numerical approximation of fluid-structure interaction problems with application to haemodynamics.* PhD thesis, École Polytechnique Fédérale de Lausanne, 2001.

[119] S. Patankar. *Numerical heat transfer and fluid flow.* Computational Methods in Mechanics and Thermal Sciences. McGraw-Hill, New York, 1980.

[120] S. Patankar and D. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transfer*, 15:1787–1806, 1972.

[121] C. Pechstein. *Finite and Boundary Element Tearing and Interconnecting Methods for Multiscale Elliptic Partial Differential Equations.* PhD thesis, Johannes Kepler University Linz, 2008.

[122] S. Piperno. Explicit/implicit fluid-structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations. *Int. J. Numer. Methods Fluids*, 25:1207–1226, 1997.

[123] S. Piperno. Numerical simulation for civil engineering: aeroelastic instabilities of elementary bridge decks. In *Computational Methods for Fluid-Structure Interaction (ed. by T. Kvamsdal et al.)*, pages 41–50, Tapir, 1999.

[124] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems-Part II: Energy transfer analysis and three-dimensional applications. *Comput. Methods Apply. Mech.*, 190:3147–3170, 2001.

[125] P. Plaszczak and R. Wellner. *Grid Computing.* Morgan Kaufmann Pub, 2005.

[126] A. Quaini. *Algorithms for fluid-structure interaction problems arising in hemodynamics.* PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2009.

[127] A. Quarteroni, M. Tuveri, and A. Veneziani. Computational vascular fluid dynamics: problems, models and methods. *Comput. Vis. Sci.*, 2:163–197, 2000.

[128] B. Quatember and M. Mayr. GRID Software Solution for the Segmentation of the Co ronary Artery Tree in Biplane Angiograms. In *EUROCAST 2007*, pages 457–464.

[129] B. Quatember, M. Mayr, and W. Recheis. Patient-specific modelling and simulation of coronary haemodynamics. In *SpringSim 2008*, 573-580.

[130] B. Quatember and F. Veit. Simulation Model of Coronary Artery Flow Dynamics and Its Applicability in the Area of Coronary Surgery. In *EUROSIM 1995*, 945-950.

[131] C. Reisinger and M. Wabro. Analysis and numerical solution of the k-ϵ turbulence model with non-standard boundary conditions. Master's thesis, Johannes Kepler University of Linz, Institute of Computational Mathematics, Linz, 1999.

[132] S. Reitzinger. *Algebraic Multigrid Methods for Large Scale Finite Element Equations.* PhD thesis, Johannes Kepler University Linz, 2001.

[133] T. J. R.Hughes, L. P. Franca, and M. A. Ballestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the BabuškaBrezzi condition: a stable PetrovGalerkin formulation of the Stokes problem accommodating equal order interpolations. *Comput. Method Appl. Mech. Engrg.*, 59:85–99, 1986.

[134] H. Rosmanith and D. Kranzlmüller. glogin-A Multifunctional, Interactive Tunnel into the Grid. In *Proceedings of Grid 2004, 5th IEEE/ACM Intl.Workshop on Grid Computing*, pages 266–272, Pittsburgh, PA, USA, 2004. IEEE Computer Society.

[135] J. W. Ruge and K. Stüben. Algebraic multigrid (AMG). In *Multigrid Methods*, volume 5 of *Fontiers in Applied Mathematics*, pages 73–130. SIAM, Philadelphia, 1986.

[136] S. Rugonyi and K. Bathe. On finite element analysis of fluid flows coupled with structural interaction . *Computer Modeling in Engineering and Sciences (CMES)*, 2(2):195–212, 2001.

[137] A. Russo. Bubble stabilization of finite element methods for the linearized incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg*, 132:335–343, 1995.

[138] A. Russo. A posteriori error estimators for the Stokes problem. *Appl. Math. Lett.*, 8(2):1–4, 1995.

[139] Y. Saad. *Iterative methods for sparse linear systems.* SIAM, Philadelphia, second edition, 2003.

[140] Y. Saad and M. H. Schultz. Gmres : A generalized minimal residual algorithm for solving nonsymmetric linear system. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[141] J. Schöberl and W. Zulehner. On Schwarz-type smoothers for saddle point problems. *Numer. Math.*, 95(2):377–399, 2003.

[142] D. J. Silvester and A. J. Wathen. Fast iterative solution of stabilized stokes systems. II. Using general block preconditioners. *SIAM J. Num. Anal.*, 31:1352–1367, 1994.

[143] R. Simon. *Multigrid Solvers for Saddle Point Problems in PDE-Constrained Optimization.* PhD thesis, Johannes Kepler University, 2008.

[144] J. Sokolowski and J.P. Zolesio. *Introduction to Shape Optimization*, volume 16 of *Springer series in computational mathematics.* Springer, New York, Berlin, Heidelberg, 1992.

[145] K. Stüben. An introduction to algebraic multigrid. In *Multigrid*, pages 413–532. Academic Press, New York, London, 2001.

[146] K. Stüben. A review of algebraic multigrid. *Computational and Applied Mathematics*, 128:281–309, 2001.

[147] H. Sundar, R. S. Sampath, and G. Biros. Bottom up construction and 2:1 balance refinement of linear octree in parallel. *SIAM Journal on Computing*, 30(5):2675–2708, 2008.

[148] P. L. Tallec and J. Mouro. Fluid structure interaction with large structural displacement. *Comput. Methods Appl. Mech. Engrg.*, 190:3039–3067, 2001.

[149] T. E. Tezduyar. Finite element methods for fluid dynamics with moving boundaries and interfaces. *Arch. Comput. Methods Engrg.*, 8:83–130, 2001.

[150] T. E. Tezduyar, M. Behr, S. Mittal, and J. Liou. A new stategy for finite element computational involving moving boundaries and interfaces–the deforming- spatial-domain/space-time procedure. ii. *Computation of free-surface flows, two-liquid flows, and flows with drifting cylingers*, 94(3):353–371, 1992.

[151] L. Tobiska and G. Lube. A modified streamline diffusion method for solving the stationary Navier-Stokes equations. *Numer. Math.*, 59:13–29, 1991.

[152] L. Tobiska and R. Verfürth. Analysis of a streamline diffusion finite element method for the Stokes and Navier-Stokes equations. *SIAM J. Numer. Anal.*, 33(1):107–127, 1996.

[153] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, New York, London, 2001.

[154] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56:179–196, 1996.

[155] S. Vanka. Block-implicit multigrid calculation of two-dimensional recirculating flows. *Comp. Meth. Appl. Mech. Eng.*, 59(1):29–48, 1986.

[156] P. S. Vassilevski. *Multilevel Block Factorization Preconditioners*. Springer, 2008.

[157] R. Verfürth. Error estimates for a mixed finite element approximation of the Stokes equations. *R.A.I.R.O. Num. Anal.*, 18(2):175–182, 1984.

[158] R. Verfürth. A multilevel algorithm for mixed problems. *SIAM J. Numer. Anal.*, 21:264–271, 1984.

[159] R. Verfürth. A *posteriori* error estimators for the Stokes equations. *Numer. Math.*, 55:309–325, 1989.

[160] S. Vladimir. *Grid Computing for Developers*. Charles River Media, 2005.

[161] M. Wabro. *Algebraic Multigrid Methods for the Numerical Solution of the Incompressible Navier-Stokes Equations*. PhD thesis, Johannes Kepler University, 2003.

[162] M. Wabro. Coupled algebraic multigrid methods for the oseen problem. *Computing and Visualization in Science*, 7(3-4):141–151, 2004.

[163] M. Wabro. AMGe - Coarsening strategies and application to the Oseen equations. *SIAM J. Sci. Comput.*, pages 2077–2097, 2006.

[164] C. Wagner. On the algebraic construction of multilevel transfer operators. *Computing*, 65:73–95, 2000.

[165] W. A. Wall. *Fluid-Structure Interaction with Stabilized Finite Elements*. PhD thesis, Institute of Structural Mechanics, University of Stuttgart, 1999.

[166] J. Watzl. *Investigations on Improving the SEE-GRID Optimization Algorithm*. PhD thesis, Johannes Kepler University of Linz, Research Institute for Symbolic Computation (RISC), Linz, Austria, 2008.

[167] R. Webster. An algebraic multigrid solver for Navier-Stokes problems. *Int. J. Numer. Meth. Fluids*, 18:761–780, 1994.

[168] G. Wittum. Multi-grid methods for Stokes and Navier-Stokes equations. *Numer. Math.*, 54:543–563, 1989.

[169] G. Wittum. On the convergence of multi-grid methods with transforming smoothers. *Numer. Math.*, 57:15–38, 1990.

[170] H. Yang and W. Zulehner. A Newton based fluid-structure interaction solver with algebraic multigrid methods on hybrid meshes. Technical Report 2009-10, Institute of Computational Mathematics, Johannes Kepler University of Linz, 2009. submitted.

[171] H. Yang and W. Zulehner. Numerical simulations of fluid-structure interaction problems on hybrid meshes with algebraic multigrid methods. *Preprint submitted to Journal of Computational and Applied Mathematics*, 2009.

[172] H. Yang and W. Zulehner. Numerical simulations of fluid-structure interaction problems on hybrid meshes with algebraic multigrid methods. In I. Lirkov, S. Margenov, and J. Wasniewski, editors, *Large-Scale Scientific Computing*, pages 116–123, Berlin, Heidelberg, 2010. Springer.

[173] H. Yang, W. Zulehner, U. Langer, and M. Baumgartner. A robust PDE solver for the 3D Stokes/Navier-Stokes systems on the grid environment. In *GRID '07: Proceedings of the 8th IEEE/ACM International Conference on Grid Computing*, pages 145–152, Washington, DC, USA, 2007. IEEE Computer Society.

[174] K. Yosida. *Functional analysis*. Springer Classics in Mathematics. Springer, Berlin-Heidelberg-New York, 6th edition, 1980.

[175] H. Zhang, X. Zhang, S. Ji, G. Guo, Y. Ledezma, N. Elabbasi, and H. deCougny. Recent development of fluid-structure interaction capabilities in the adina system. *Computers & Structures*, 81(8-11):1071–1085, 2003.

[176] W. Zulehner. A class of smoothers for saddle point problems. *Computing*, 65(3):227–246, 2000.

[177] W. Zulehner. Analysis of iterative methods for saddle point problems: A unified approach. *Math. Comput*, 71(238):479–505, 2002.

[178] W. Zulehner. Uzawa-type methods for block-structured indefinite linear systems. SFB-Report 05-05, Johannes Kepler University Linz, SFB 'Numerical and Symbolic Scientific Computing', 2005.

Eidesstattliche Erklärung

Ich, Dipl.-Math. Huidong Yang, erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Linz, December 2009

_____

Huidong Yang