Coupling Different Discretizations for Fluid Structure Interaction in a Monolithic Approach

Doctoral Dissertation submitted to the

Faculty of Informatics of the Università della Svizzera Italiana

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

presented by

Johannes Steiner

under the supervision of

Prof. Rolf Krause

October 2014

Dissertation Committee

Prof. Rolf Krause	Università della Svizzera italiana
Prof. Igor Pivkin	Università della Svizzera italiana
Prof. Olaf Schenk	Università della Svizzera italiana
Prof. Axel Klawonn	Universität zu Köln, Germany
Prof. Arnold Reusken	RWTH Aachen, Germany

Dissertation accepted on 14. October 2014

Research Advisor

Prof. Rolf Krause

PhD Program Director

Prof. Igor Pivkin, Prof. Stefan Wolf ***pro tempore***

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Johannes Steiner
Lugano, 14. October 2014

Abstract

In this thesis we present a monolithic coupling approach for the simulation of phenomena involving interacting fluid and structure using different discretizations for the subproblems. For many applications in fluid dynamics, the Finite Volume method is the first choice in simulation science. Likewise, for the simulation of structural mechanics the Finite Element method is one of the most, if not the most, popular discretization method. However, despite the advantages of these discretizations in their respective application domains, monolithic coupling schemes have so far been restricted to a single discretization for both subproblems. We present a fluid structure coupling scheme based on a mixed Finite Volume/Finite Element method that combines the benefits of these discretizations. An important challenge in coupling fluid and structure is the transfer of forces and velocities at the fluid structure interface in a stable and efficient way. In our approach this is achieved by means of a fully implicit formulation, i.e., the transfer of forces and displacements is carried out in a common set of equations for fluid and structure. We assemble the two different discretizations for the fluid and structure subproblems as well as the coupling conditions for forces and displacements into a single large algebraic system. Since we simulate real world problems, as a consequence of the complexity of the considered geometries, we end up with algebraic systems with a large number of degrees of freedom. This necessitates the use of parallel solution techniques.

Our work covers the design and implementation of the proposed heterogeneous monolithic coupling approach as well as the efficient solution of the arising large nonlinear systems on distributed memory supercomputers. We apply Newton's method to linearize the fully implicit coupled nonlinear fluid structure interaction problem. The resulting linear system is solved with a Krylov subspace correction method. For the preconditioning of the iterative solver we propose the use of multilevel methods. Specifically, we study a multigrid as well as a two-level restricted additive Schwarz method. We illustrate the performance of our method on a benchmark example and compare the afore mentioned different preconditioning strategies for the parallel solution of the monolithic coupled system.

Acknowledgements

I want to thank my advisor Prof. Rolf Krause for the support over the last years and the possibility to present my work at different conferences. I also want to express my gratitude to the other members of my dissertation committee, Prof. Igor Pivkin, Prof. Olaf Schenk, Prof. Axel Klawonn and Prof. Arnold Reusken, for their time and interest. Thanks to Dr. Daniel Rupprecht for proofreading parts of this work. A special thanks goes to Dr. Dorian Krause for many discussions and for proofreading this thesis.

Contents

1 Introduction

The numerical simulation of the interaction of fluids and structures is an important challenge in life-science and engineering. The areas of application range from, e.g., blood flow [WRS$^+$05, Hro07, TYW$^+$02, DHBPS03, KBB09, KNE$^+$08, WSKH04, FOT$^+$06, CLL06, PRK07], parachutes [TO01, SBTP01, SBC$^+$07], naval architectures [Oha01], aerospace [KS04] to hydraulic engines [SL04]. All of these applications have in common that a fluid interacts with the surface of a structure. In this interaction the structure is usually deformed and this deformation leads to a changed fluid domain. In this interplay of fluid and structure both subproblems interact dynamically. From a mathematical point of view, this means, that neither the fluid nor the structure subproblem can be solved without considering the other subproblem. Both subproblems are already difficult to solve, since each of them can be nonlinear, depending on the employed model. The handling of the coupled problem is even more challenging, because an additional global non-linear problem is created at the fluid structure interface. The main challenges of fluid structure interaction problems are:

- Formulation of the subproblems (modeling)
- Coupling of the subproblems
- Discretization of the coupled problem
- Design of an efficient and robust solver for the coupled problem.

These main challenges also interact in the sense that a design decision in one part directly influences the others. In the first step one has to choose models for the fluid and the structure subproblems. This can be done separately for each field and typically results in a set of differential equations for both subproblems. In the next step these equation have to be coupled. For fluid structure interaction problems the coupling takes place at a common interface of fluid and structure. Here two physically motivated coupling conditions can be derived: The continuity of velocities and the balance of forces. In a typical fluid structure interaction problem the structure is deformed in the course of time. The natural way to describe a structure in motion is a Lagrangian formulation, i.e., the deformation of the material also deforms the computational domain. On the other side of the fluid structure interface we face a different situation. For a fluid the Eulerian formulation is the natural choice, i.e., the computational domain is kept fixed and the material points move. In a coupled simulation the fluid domain is deformed by the structure displacements at the common interface. At the interface the Lagrangian and the Eulerian formulation clash. Here usually one of the formulations is (in parts) reformulated to obtain a matching formulation at the interface. Often an Arbitrary Lagrangian Eulerian formulation is used for a fluid in a moving domain. Therein the fluid is reformulated and can be Eulerian in a sub-domain of the fluid domain and Lagrangian in some others. Between the Eulerian and the Lagrangian parts it is a mixture. The motion of the fluid domain can be formulated as a third problem in addition to the fluid and the structure subproblems. Therefore the coupled fluid structure problem can be considered as a three field problem consisting of:

- Fluid subproblem
- Structure subproblem
- Geometry subproblem.

The discretization of the subproblems and coupling conditions at the interface are typically approached in one of two ways. Either

- separate and usually different fluid and structure discretization methods for the subproblems or
- the same discretization method for the subproblems

are used. From this point of view one often distinguishes analogously between two main coupling concepts:

- Partitioned approaches
- Monolithic approaches.

In the partitioned coupled approach, the two subproblems are solved in a staggered fashion with an unidirectional information transfer at the interface. In the each solution step the information is taken as a boundary condition for the respective subproblem. However, partitioned approaches have serious drawbacks, as they usually result in non-matching meshes at the interface during the staggered solution process and their convergence properties are unclear. On the other hand they profit from their modularity, where the discretization or the solver for one subproblem can be replaced without major changes to the other subproblem. The main advantage of partitioned scheme is that existing, highly optimized, solver can be reused. In contrast, monolithic approaches handle the coupling implicitly. Here, the governing equations of fluid and structure are discretized and solved simultaneously using a single nonlinear solution scheme. The main advantage of this solving strategy is that monolithic coupled schemes are more robust.

In this work we present a novel coupling approach that combines the advantages of the Finite Volume method for the discretization of the Navier Stokes equations with the benefits of a Finite Element discretization for the structure problem on general body-fitted unstructured meshes. The coupling of different discretization methods is appealing since we can use specially tailored methods for each subproblem. In fact, for many applications in fluid dynamics, the Finite Volume method is the first choice and likewise, for the simulation of structural mechanics, the Finite Element method is the most popular discretization method. The main contribution of the presented work is the design of a heterogeneous monolithic coupling scheme as well as the study of efficient and scalable solution techniques for the arising large-scale nonlinear equations. We model the fluid by means of the incompressible Navier Stokes equations in an Arbitrary Eulerian Lagrangian formulation, while we use a pure Lagrangian formulation for the structure. This leads to an additional quantity to be coupled, a domain velocity for the fluid domain, which we can handle implicitly or explicitly in the inner of the fluid domain in our coupling scheme. For the discretization in time we use different implicit time stepping schemes. We illustrate our approach along two applications. First, we present a patient specific simulation based on geometries derived from clinical data. Second, we discuss an application in the field of microelectromechanical sensor design. As a consequence of the complexity of the considered geometries, we end up with algebraic systems with a large number of degrees of freedom, which makes the use of parallel solvers mandatory. Here, a good choice for an efficient preconditioning strategy is important. Our solver is based on the application of Newton's method to linearize the fully implicit coupled nonlinear fluid structure problem. The resulting linear system is solved with a Krylov subspace correction method. We use different preconditioning strategies for the Krylov method, namely a geometric multigrid method and a restricted two level additive Schwarz method based on a hierarchy of unstructured meshes. We illustrate the performance of our method on a benchmark example and compare different preconditioning strategies for the parallel solution of the monolithic coupled system.

2 Outline

In this thesis we present the monolithic coupling of a Finite Element method and a Finite Volume method for fluid-structure interaction. The text is organized as follows:

- In Chapter three we describe the underlying subproblems. We start with an introduction to different choices of the frame of reference. For the structure subproblem we introduce linear elasticity in a Lagrangian framework. For the fluid subproblem we introduce the Navier Stokes equations in an Eulerian formulation. In preparation for the coupled description, the Navier Stokes equations are then reformulated in an Arbitrary Lagrangian Eulerian formulation.

- In Chapter four we present different coupling schemes for fluid structure interaction. Here, we mainly focus on partitioned and monolithic coupled schemes.

- In Chapter five we discuss the coupling conditions for fluid structure interaction. We start with the physically motivated coupling conditions for velocities and forces. Then we discuss the additional geometric coupling condition that is required because of the Arbitrary Lagrangian Eulerian formulation of the fluid. Finally we write the coupled problem as a coupled geometry, fluid, and structure problem.

- Chapter six is concerned with geometric conservation laws. The Arbitrary Lagrangian Eulerian formulation for the fluid introduces an additional velocity field. This velocity is an artificial fluid domain velocity, i.e, there is no direct physical interpretation for it. It needs special attention since conservation laws for this velocity can be derived. The goal is to achieve mass conservation for a fluid in a moving domain.

- In Chapter seven we present a weak formulation for fluid and structure. Here we start with the structure subproblem. We use a Finite Element method for the structure subproblem. For the fluid we use piecewise constant test functions, resulting in a Finite Volume formulation. In this Chapter we also discuss the weak formulation of the force coupling.

- In Chapter eight we present the time discretization. In this thesis we use time stepping schemes based on Runge Kutta methods.

- Chapter nine is concerned with the spatial discretization for structure, fluid and the coupling conditions. The discretization of the structure is done using Finite Elements. For the fluid we use a Finite Volume method. In this Chapter we describe the construction of the dual mesh for the Finite Volume method and the used stabilization. Finally we present the discretization of the coupling conditions.

- In Chapter ten we describe two different strategies for the treatment of the mesh motion of the interior of the fluid domain. In the first strategy, we solve the implicit coupled fluid structure problem by using an auxiliary equation for the fluid domain motion. In the second strategy, a geometric explicit method, we split the full implicit coupled fluid structure problem into two subproblems. Here, we have to solve a second problem in order to compute a smooth distribution for the fluid mesh.
- In Chapter eleven we present the Jacobian matrix of the coupled system and how the coupling takes place in there.
- Chapter twelve is about solver and preconditioning. We start with an overview about different iterative solver. We also present the used multigrid and additive Schwarz method.
- Chapter thirteen highlights two applications of our simulation code to real-world problems, namely the simulation of a fluid metering device and a simulation of an aneurysm are presented. For the second application we also present the setup of the workflow from the handling of medical data to the simulation of blood flow on a realistic geometry.
- In Chapter fourteen we present a benchmark example and the scaling behavior of the used preconditioning strategies.

3 Problem description

The equations for fluid and structure are defined in different frameworks. The fluid equations are usually written in an Eulerian formulation. In contrast to the fluid, the structure equations are usually formulated from a Lagrangian perspective. In the interplay of fluid and structure, the structure is deformed at the fluid structure interface, which leads to the fact that the fluid domain is also deformed in the course of time. Therefore the Eulerian points at the interface and potentially in the interior of the fluid domain move. A way to deal with this situation is to reformulate the fluid in an Arbitrary Lagrangian Eulerian formulation. In this chapter we first formulate the structure in a pure Lagrangian and the fluid in an Eulerian formulation. Then we write the fluid problem in an Arbitrary Lagrangian Eulerian formulation for the coupled problem. For a more comprehensive introduction to fluid and structure mechanics in the context of fluid structure interaction we refer to [Dep04, Nob01, Qua09]

Let Ω_{t_0} be a bounded open reference domain in $\mathbb{R}^3$ and its interior be filled with a continuous substance. We define a smooth mapping

$$\varphi : \Omega_{t_0} \times (t_0, T) \to \Omega_t \subset \mathbb{R}^d,$$

which we call a motion.

3.1 Lagrangian framework

In the Lagrangian framework spatial points coincide with material points. The deformation of the material can be formulated canonically with the change of the computational domain. This framework can be seen as the natural choice for the description of deformations of a structure since a deformation can easily be written as

$$d(X,t) = \varphi(X,t) - X$$

for a given X in the reference computation domain. For the Lagrangian framework we define a family of mappings

$$\mathcal{L} : \Omega_{t_0} \to \Omega_t, \quad \varphi(X,t) = \mathcal{L}_t(X),$$

that keeps track of the moving domain in the course of time.

3.2 Eulerian framework

In the Eulerian framework spatial and material points $x \in \Omega_t$ do not coincide. Therefore this formulation can be seen as the self-evident formulation for a fluid. With Gauss' theorem, mass conservation laws can easily be described by fluxes over boundaries of (control) volumes. Since the spacial points do not move we do not have to define a family of mappings for the Eulerian formulation to keep track of the computational domain.In this section we will use superscripts to distinguish between a function in a Lagrangian and Eulerian framework. Given a function f^L in Lagrangian framework it can be expressed in the Eulerian framework f^E using

$$f^L(X,t) = f^E \circ \varphi(X,t)$$

We have already defined a deformation in the Lagrangian framework and it is easy to derive a velocity, that is the change of displacement in time

$$v(X,t) = \frac{\partial \varphi(X,t)}{\partial t} = \frac{\partial d(X,t)}{\partial t}.$$

The time derivative

$$\frac{D}{Dt}$$

with respect to a moving coordinate system is called Lagrangian or material derivative. The material derivative can be seen as a link between Eulerian and Lagrangian descriptions. It describes the time rate of change of a scalar or a vector field of fluid parcels, or more general a material element, under the impact of a time and space dependent velocity field. It can be derived by using the chain rule for $f^E \circ \varphi$

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + v \cdot \nabla f.$$

The Reynolds transport theorem gives a similar link between Eulerian and Lagrangian descriptions for the computation of the material derivative of integrals. It can be seen as a higher dimensional Leibniz integral rule. In the Reynolds transport theorem a time-dependent material volume or (control) volume $V(t)$ is used. The control volume may change in time and we denote the outer normal vector of the boundary with n. The Reynolds transport theorem then reads

$$\frac{d}{dt} \int_{V(t)} f \, dV = \int_{V(t)} \frac{\partial f}{\partial t} \, dV + \int_{\partial V(t)} f \, (v \cdot n) \, dS$$

where $\partial V(t)$ is the surface of $V(t)$.

3.3 Conservation of mass

For the derivation of the principle of mass conservation we use the material volume $V(t)$, which is the amount of all material points which are in the volume $V_0 = V(t_0)$ at time t_0. We introduce a density $\rho(x,t) > 0$ and define the mass of a material volume

$$m(V) := \int_V \rho(x,t)\; dV.$$

In the following we simplify the notation by omitting (x,t). The assumption of mass conservation states that mass is neither generated nor destroyed, i.e,

$$\frac{d}{dt}m(V) = \frac{d}{dt}\int_V \rho\; dV = 0.$$

Applying the Reynolds transport theorem we get

$$\int_V \frac{d}{dt}\rho\; dV + \int_{\partial V} \rho(v \cdot n)\; dS = 0.$$

The assumption of an incompressible fluid states that, even for high pressure, the volume $V(t)$ of the fluid remains constant. Therefore the condition

$$\frac{d}{dt}\int_V \rho\; dV = 0$$

is reduced for constant ρ to

$$\frac{d}{dt}\int_V \; dV = 0.$$

3.4 Conservation of momentum

Conservation of momentum states that in a closed system the total momentum is constant. In continuum mechanics we have to deal with body forces in V and surface forces on ∂V. A Body forces is given by

$$F_B(t) = \int_V \rho(x) f\; dV$$

here $f(x,t)$ denotes a force density within the volume. The surface forces are described via the force density σ acting on a surface element $\partial V = \partial V(t)$

$$F_S(t) = \int_{\partial V} n\sigma\; dS,$$

here n denotes the unit normal vector of the surface element. The stress tensor σ contains already the properties of the material. In the respective section about fluid and structure we present the used constitutive equations. Thus the total force equal

$$F(t) = \int_V \rho f\; dV + \int_{\partial V} n\sigma\; dS = \int_V \rho f + \nabla \cdot \sigma\; dV.$$

The momentum, which is the product of the mass and velocity, is given by

$$M(t) = \int_V \rho u \, dV.$$

Newton's second law states that the change of momentum is directly proportional to the resultant force applied to an object

$$\frac{d}{dt}\int_V \rho u \, dV = \int_V \rho f + \nabla \cdot \sigma \, dV.$$

We apply Reynolds transport and Gauss' theorem for $f = \rho u$ component wise and get

$$\frac{d}{dt}\int_V \rho u_i \, dV = \int_V (\frac{\partial(\rho u_i)}{\partial t} + \frac{(\partial \rho u_j u_i)}{\partial x_j}) \, dV.$$

Using this we can write the momentum equation under the assumption of a constant ρ as

$$\rho \frac{\partial}{\partial t} u + \rho u \cdot \nabla u = \rho f + \nabla \cdot \sigma. \tag{3.1}$$

3.5 Structure mechanics

For a given mapping φ we define a deformation gradient for each material point in Ω_{t_0} as

$$F = \nabla \varphi = I + \nabla d^s.$$

Its determinant

$$J = \det F$$

has to be is strictly positive. For the structure we use a Lagrangian framework and get

$$\rho^s \frac{\partial}{\partial t} v^s - \nabla \cdot \sigma_P^s = \rho^s f^s.$$

For the structure we use v^s to denote the velocity field. In the following we usually use for the structure a superscript s while for the fluid we use f. The stress tensor σ^s is written in the current deformed configuration. However we want to formulate the stresses in the reference configuration, therefore we introduce the first Piola-Kirchhoff stress tensor

$$P := \det(\nabla \varphi) \sigma_P^s (\nabla \varphi)^{-T}.$$

Since the first Piola-Kirchhoff stress tensor is not symmetric, we moreover introduce the second Piola-Kirchhoff stress tensor

$$\sigma^s := \det(\nabla \varphi)(\nabla \varphi)^{-1} P (\nabla \varphi)^{-T} = F^{-1} P.$$

With this symmetric tensor a constitutive relation can be easily formulated. Under the assumption of an isotropic linear elastic material the constitutive relation can be written as

$$\sigma^s = \lambda^s(\mathrm{tr}(\varepsilon^s))I + 2\mu^s\varepsilon^s,$$

where ε is the Green-Lagrange strain tensor and λ^s and μ^s the Lamé constants. These constants describe a certain material behavior. Often material properties are specified in terms of the Young modulus E^s and the Poisson number ν^s. The Young modulus and the Poisson number can be written in terms of the Lamé constants and vice versa

$$E^s = \frac{\mu^s(2\nu^s + 3\lambda^s)}{\mu^s + \nu^s},$$

$$\nu^s = \frac{\lambda^s}{2(\mu^s + \lambda^s)},$$

$$\mu^s = \frac{E^s}{2(1+\mu^s)},$$

$$\lambda^s = \frac{\mu^s E^s}{(1+\nu^s)(1-2\nu^s)}.$$

The Green-Lagrange strain tensor ε^s is defined as

$$\varepsilon^s := \frac{1}{2}(C - I) = \frac{1}{2}(F^T F - I),$$

where C is the right Cauchy-Green-Tensor. The Green-Lagrange strain tensor can be written as

$$\varepsilon^s = \frac{1}{2}(\nabla d^s + (\nabla d^s)^T + (\nabla d^s)^T \nabla d^s).$$

However, for small displacements d we can ignore the higher order terms and get the following strain-displacement relation

$$\varepsilon^s = \frac{1}{2}(\nabla d^s + (\nabla d^s)^T)$$

which is called the linearized strain tensor. So finally we can write the structure problem as

$$\frac{\partial^2}{\partial t^2} d^s - \frac{1}{\rho_s}\nabla \cdot \sigma^s \quad = \quad f^s,$$

with additional boundary conditions. Here we will focus on two different types, namely Dirichlet and Neumann conditions. The first describe displacements on a specific part of the boundary Γ_t^D

$$d^s = g_D \text{ on } \Gamma_t^D.$$

The Neumann boundary conditions are surface stresses applied on a part of the boundary Γ_t^N

$$\sigma^s(d^s) \cdot n = g_N \text{ on } \Gamma_t^N.$$

3.6 Fluid mechanics

The fluid subproblem can be described with different models depending on the application. Characteristic features are, e.g., compressibility and viscosity. Air for example has a lower viscosity than blood and liquid fluids are often modeled as incompressible while air is usually assumed to be compressible. For modeling air the Euler equations are often chosen because the viscous properties of this fluid can usually be neglected. For blood or water, where viscous effects can not be neglected, the Navier Stokes equations are a more appropriate choice than the Euler equation. For the derivation of the fluid equations we will assume a fixed domain in time and we will derive the fluid in the Eulerian framework. In a second step we will formulate the Navier Stokes equations in an Arbitrary Lagrangian Eulerian formulation.

The main difference in the material behavior for the two subproblems is given by the different stress tensors. Therefore we have to specify the stress tensors in the conservation of momentum equation also for the fluid. This tensor models the frictional forces inside the fluid: For a Newtonian fluid, σ^f is given in a linear model by

$$\sigma^f(u^f, p^f) = -p^f I + 2\mu^f \varepsilon^f(u^f),$$

where $p^f = p^f(x,t)$ is the pressure. The second part $\mu^f \varepsilon^f(u)$ is the viscous part of the stress tensor and μ^f is the dynamic viscosity. Sometimes we use $\nu^f = \frac{\mu^f}{\rho^f}$ the kinematic viscosity and assume without loss of generality $\rho^f = 1$. We use the following strain rate tensor

$$\varepsilon^f(u^f) = \frac{1}{2}(\nabla u^f + (\nabla u^f)^T). \tag{3.2}$$

If we insert equation (3.2) in equation (3.1) we get the Navier Stokes equations. We have used the Eulerian frame to write this equation. For a time independent domain Ω we can write this as a system with unknowns u^f and p^f [DR06, Gal94, FMRT08]

$$\frac{\partial}{\partial t} u^f + u_f \nabla u^f - \frac{1}{\rho^f} \nabla \cdot \sigma^f \quad = \quad f^f.$$

In addition, we need to prescribe an initial state for the fluid, i.e. a given fluid velocity

$$u_0^f(x).$$

Finally boundary conditions have to be prescribed. These can be, e.g., inflow, outflow, stress and wall conditions. For the incompressible Navier Stokes equations the density stays constant for all pressures and mass conservation is given by

$$\frac{d}{dt} \int_V dV = 0.$$

The incompressibility condition therefore reads

$$\nabla \cdot u^f \quad = \quad 0.$$

Finally we get the incompressible Navier Stokes equations on a fixed domain

$$\begin{aligned} \frac{\partial}{\partial t} u^f + u^f \nabla u^f - \frac{1}{\rho^f} \nabla \cdot \sigma^f &= f^f \\ \nabla \cdot u^f &= 0. \end{aligned}$$

3.7 Arbitrary Lagrangian Eulerian formulation

The structure is usually formulated in a Lagrangian way, while for the fluid an Eulerian formulation is used. In this section we will derive the concept of the Arbitrary Lagrangian Eulerian formulation. We are going to couple an elastic body with a fluid, which means in our case that we have to couple an Eulerian formulation with a Lagrangian formulation. A very common approach to couple the formulations is to reformulate the fluid in an Arbitrary Lagrangian Eulerian formulation.

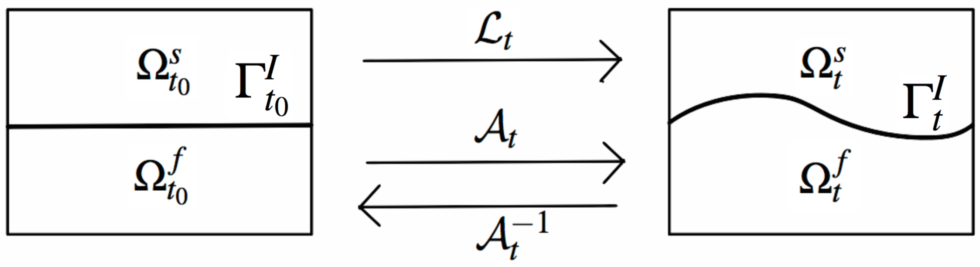


Figure 3.1. *Fluid structure interaction problem. The structure remains Lagrangian, while the fluid is reformulated in an Arbitrary Lagrangian Eulerian formulation. We denote the common interface between fluid and structure by* Γ^I_t.

We will enforce that the fluid domain follows the deformation of the structure at the fluid structure interface. In this case the structure can remain in a Lagrangian formulation, while in the inner of the fluid domain we have to reformulate the configuration setting of the fluid. For a sketch of the situation we refer to Figure 3.1. On a non moving part of the fluid domain the fluid remains in the Eulerian formulation and on the moving part it is both, in the interior, Arbitrary Lagrangian Eulerian and at the interface fully Lagrangian. The aim of this section is to write the Navier Stokes equations in an Arbitrary Lagrangian Eulerian formulation. The details on the following description can be found in [Nob01, Cro11, Pen09]. We introduce a family of mappings $(\mathcal{A}_t)$, and $\mathcal{A}_t$ maps a point Y from the reference configuration Ω_{t_0} to a point x in the moved domain Ω_t

$$\mathcal{A}_t : \Omega_{t_0} \to \Omega_t, \quad x(Y,t) = \mathcal{A}_t(Y), \quad \Omega_t \times (t_0, T).$$

The notation $\Omega_t \times (t_0, T)$ is actually an abuse of notation since the space-time computational domain has a no tensor structure if the spatial domain is time dependent. For the sake of clarity we however use the notation $\Omega_t \times (t_0, T)$ instead of $\bigcup_{t\in(t_0,T)} \Omega_t \times \{t\}$. We want to remark that we have chosen without loss of generality the domain Ω_{t_0} with $t_0 = 0$ as the reference configuration. The mapping is assumed to be a continuous bijection of the closure of Ω_0 onto the closure of Ω_t with a continuous inverse. We also require that the mapping $t \to \mathcal{A}_t(Y)$ is differentiable almost everywhere in $(0,T)$. We now take a function in the Eulerian frame

$$f : \Omega_t \times (0,T) \to \mathbb{R}.$$

We can use the Arbitrary Lagrangian Eulerian map to define the function f in the Arbitrary Lagrangian Eulerian frame

$$f_{\mathcal{A}} : \Omega_{t_0} \times (0,T) \to \mathbb{R} \quad f_{\mathcal{A}}(Y,t) = f(\mathcal{A}_t(Y),t).$$

We introduce the Arbitrary Lagrangian Eulerian derivatives

$$\frac{\partial f}{\partial t}(x,t) = \frac{\partial f_A}{\partial t}(\mathcal{A}_t^{-1}(x),t) = \left.\frac{\partial f}{\partial t}\right|_x + \left.\frac{\partial x}{\partial t}\right|_Y \cdot \nabla_x f(x,t),$$

and we call

$$w(x,t) = \left.\frac{\partial x}{\partial t}\right|_{Y=\mathcal{A}_t^{-1}(x)}$$

the fluid domain velocity.

For special choices of $\mathcal{A}_t$ we receive the Eulerian and Lagrangian formulation. For $\mathcal{A} = I$, which is equivalent to $w = 0$, we get a pure Eulerian formulation. For $\mathcal{A} = \mathcal{L}$, which is $w = u$ (the domain velocity is the same as the material velocity), we get a Lagrangian formulation. Usually all of these three cases occur in an fluid structure interaction simulation; for an illustration of this we refer to Figure 3.2.

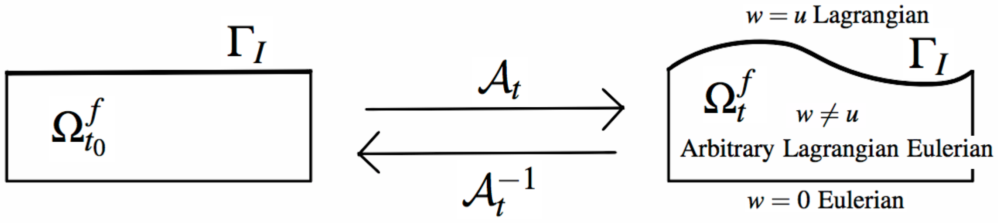


Figure 3.2. *Fluid field in a moving domain with a domain velocity w. For $w = 0$ the fluid remains in an Eulerian formulation, for $w = u$ the fluid is in a Lagrangian formulation and for $w \neq u$ is is in an Arbitrary Lagrangian Eulerian formulation.*

3.8 Navier Stokes equations in an Arbitrary Lagrangian Eulerian formulation

When dealing with fluid-structure interaction problems a common choice is an Arbitrary Lagrangian Eulerian formulation for the fluid [KAS08, LY04, BW00, Ngu10] and a purely Lagrangian framework for the structure. Usually, this is achieved by some auxiliary coordinate transformation for the fluid domain. Within the fluid domain, the Arbitrary Lagrangian Eulerian Navier Stokes equations are solved on the deforming fluid domain. At the fluid structure interface the motion of the fluid domain is driven by the displacements of the structure. For the discretization of the Arbitrary Lagrangian Eulerian formulation of the Navier Stokes equations we have to consider some geometric conservation laws for the deforming fluid domain; for details we refer to Chapter 6.

However a new equation for the fluid domain motion is required, and its dependence on the solution of the fluid-structure interaction problem introduces an additional nonlinearity with respect to the motion of the fluid domain. Using the Arbitrary Lagrangian Eulerian derivative we can write the incompressible Navier Stokes equations on moving domains as

$$\frac{\partial}{\partial t}u^f + (u^f - w^f)\cdot\nabla u^f - \frac{1}{\rho_f}\nabla\cdot\sigma^f \quad = \quad f^f \text{ in } \Omega_t \times (0,T), \tag{3.3}$$

$$\nabla\cdot u^f \quad = \quad 0 \text{ in } \Omega_t \times (0,T). \tag{3.4}$$

Here we already assume some properties for the fluid domain velocity w^f. For detail we refer to Chapter 6.

4 Coupling schemes

In the last two decades a large number of numerical methods have been developed for the simulation of fluid structure interaction. In this chapter we sketch an overview of these methods, without any claim of completeness. The main difficulty for fluid-structure interaction problems from the numerical point of view is that:

- each of the subproblems is possibly nonlinear,
- the coupling conditions at the interface introduce additional nonlinearities.

Depending of the treatment of the coupling conditions one may distinguish between different methods, e.g:

- monolithic and partitioned,
- strong and weak coupled,
- implicit, semi-implicit and explicit,
- using conforming and non-conforming meshes.
- ...

Since there is no agreement in a systematical categorization for fluid structure interaction coupling schemes, we give an overview in which we will distinguish between two general concepts:

- monolithic coupled schemes,
- partitioned coupled schemes.

In the partitioned approach [BNV09, DHA$^+$10, DBHV10, BvZ03, BQQ08] each field is separately defined, discretized and solved. The coupling is carried out by transmission and synchronization of coupled state variables at the interface. Usually the coupled system is solved at every time step using an iterative method for solving the nonlinear geometry, structure and fluid problems. Highly specialized codes and solvers for each problem can be used. However, partitioned approaches have serious drawbacks, like the added-mass effect [För07, CGN05, BQQ08].

The alleged superiority of the partitioned approach is generally attributed to the fact that smaller and better conditioned sub-systems are solved instead of one large problem. Partitioned methods

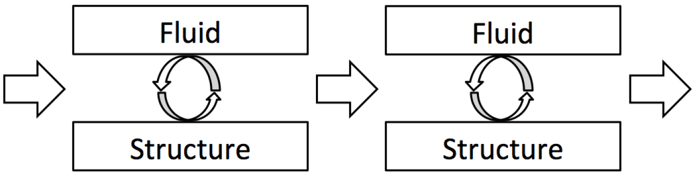


Figure 4.1. *Partitioned approach.*

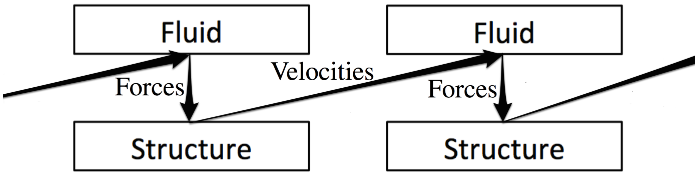


Figure 4.2. *Weak coupled partitioned approach.*

can be strong or weak coupled. In a weak coupled method the transfer of the coupling quantities is usually just done once per time step. In a strong coupled method the transfer is done in an iterative scheme until some convergence criteria is fulfilled, the two schemes can be found in Figure 4.2 and Figure 4.3.

We distinguish between two different approaches; Weak and strong coupled. In a weak coupled partitioned approach the fluid subproblem is solved, then the structure problem is solved using the forces at the fluid structure interface computed with the fluid solver as boundary conditions. The structure solver computes interface velocities that are taken as boundary conditions for the fluid solver in the next time step. These weak coupled schemes tend to be unstable, since there is no convergence control for exchanged quantities. This scheme is shown in Figure 4.2. In a strong coupled partitioned approach the exchange of coupling variables is done within an iterative method. In one time step velocities and forces are exchanged several times until a convergence criterium is fulfilled. Several relaxation methods have been developed for this kind of coupling, that is shown in Figure 4.3. In contrast, monolithic approaches [HWD04, KTZ09, WKHD05, MvBdB05, RRIOn10] treat both domains simultaneously, i.e. these solution methods lead to a single set of algebraic differential equations. This approach seems to be advantageous in many situations, since the interface conditions become implicit and there are less restrictions on feasible time steps. However, the monolithic

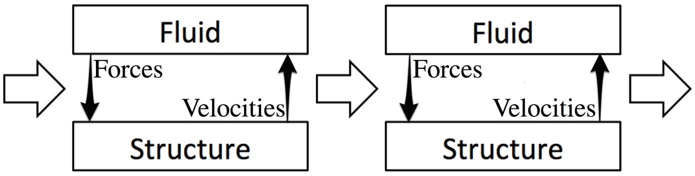


Figure 4.3. *Strong coupled partitioned approach.*

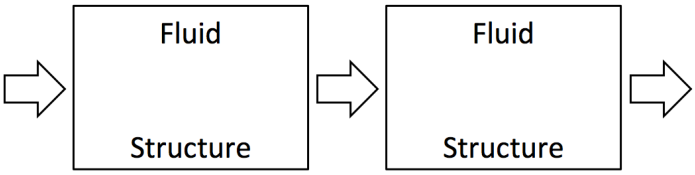


Figure 4.4. *Monolithic approach.*

approach leads to an ill conditioned Jacobian matrix. Therefore, for solving the monolithic system obtained after discretization and linearization, one has to set up a good preconditioning strategy. Monolithic algorithms are also appealing because the transmission conditions are satisfied exactly by construction and convergence results can be proven. But they are also often dismissed because it is believed that they are too expensive in terms of memory and computing time. Since nowadays increased computing performance is achieved by adding more processors, the solution to this problem is the development of parallel and scalable solution methods for fluid structure interaction problems to which the presented work contributes.

5 Coupling conditions

The coupling takes place at the interface between fluid and structure. Beside the physically motivated coupling conditions in forces and velocities, we have to formulate a geometric coupling condition, which arises from the Arbitrary Lagrangian Eulerian formulation. In this chapter we will first treat the physically motivated coupling conditions and then the so-called geometric coupling condition. The problem can be split into three problems, the fluid subproblem, the structure subproblem and a geometry subproblem.

5.1 Physical coupling conditions

We consider a structure surrounded by a fluid in a bounded volume. For a viscous fluid we can assume a no-slip condition for the common interface of the fluid and structure. A no-slip boundary condition states that at the boundary or - in our case - at the fluid structure interface the fluid has zero velocity relative to the boundary. In the model concept a particle of the fluid uttermost close to the boundary surface does not move due to molecular attraction. For a fixed boundary, i.e., the velocity for a fluid has to be zero, while for a moving boundary the fluid velocity and velocity of the boundary has to match. This leads to the following continuity of velocities (normal and tangential) condition

$$u^f \quad = \quad \frac{\partial}{\partial t} d^s \text{ on } \Gamma_t^I \times (0,T).$$

This condition describes the impact from the structure to the fluid. On the other hand the impact of the fluid to the structure is given by the force on the surface at the interface. The force acting on the structure is caused by the impulse of the stream of the fluid. For a viscous fluid this force can be expressed by the normal components of the stress tensor

$$\sigma^f \cdot n^f.$$

The equilibrium condition of the balance of forces is given by

$$\sigma^s \cdot n^s + \sigma^f \cdot n^f \quad = \quad 0 \text{ on } \Gamma_t^I \times (0,T).$$

We conclude that the physically motivated coupling conditions for a general fluid structure problem are given by means of two transmission conditions, a kinematic and a dynamic one

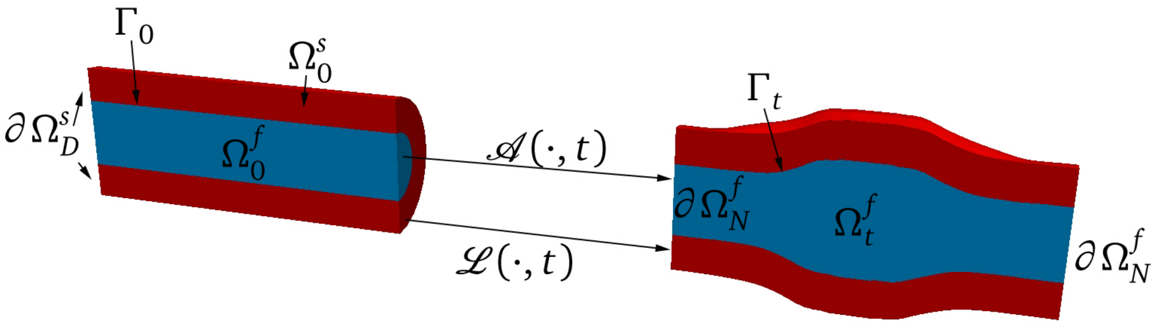


Figure 5.1. *Lagrangian and Arbitrary Lagrangian Eulerian description.*

- the continuity of velocities,
- and the balance of forces.

5.2 Geometric coupling condition

Besides the kinematic and dynamic coupling conditions the continuity of displacements must also hold on the common interface. This coupling condition can be seen as an artificial coupling condition arising for the Arbitrary Lagrangian Eulerian formulation of the fluid.

The map $\mathcal{L}_t = \mathcal{L}(\cdot,t)$, representing the deformation of the structure, and $\mathcal{A}_t = \mathcal{A}(\cdot,t)$, describing the evolution of the fluid domain must coincide on the interface Γ_t^I,

$$\mathcal{L}_t \quad = \quad \mathcal{A}_t \text{ on } \Gamma_t^I$$

for all t. Apart from satisfying the interface constraints, the fluid domain mapping $\mathcal{A}_t$ can otherwise be chosen arbitrarily. Since the evolution in time of the structure domain is described by a deformation it is also useful to describe the evolution of the fluid domain in terms of a displacement field

$$d^f(x_0,t) = \mathcal{A}(x_0,t) - x_0,$$

for any $x_0 \in \Omega_{t_0}^f$. Therefore, the geometric coupling condition is given by

$$d^f = d^s \text{ on } \Gamma_t^I,$$

since $d^s(x_0,t) = \mathcal{L}_t(x_0,t) - x_0$ for any $x_0 \in \Omega_{t_0}^s$. It gives an adhesion condition for the two domains at the interface. For a viscous flow the adhesion condition is already incorporated in the dynamic coupling condition for velocities. We want to point out again that the displacements of Ω^f at the interface are determined by the fluid structure interface, while the displacements of the mesh inside of Ω^f can be chosen arbitrarily. A common choice for the extension $d^f = \text{Ext}(d_s, \Gamma_t^I)$, is an harmonic one e.g.

$$\begin{aligned} -\Delta\, d^f &= 0 \text{ in } \Omega^f \\ d^f &= d^s \text{ on } \Gamma_t^I \end{aligned}$$

We give an overview of different possible extensions and the smoothing schemes that are used in this thesis in Chapter 10.

5.3 Coupled fluid structure problem

The coupled problem can be written as a set of three subproblems with coupling conditions. All of the subproblems themselves may be nonlinear depending on the constitutive equations. The strong form of the fluid-structure problem in the actual domain reads as follows:

1. Geometry subproblem: Find the fluid domain displacement such that

$$\begin{aligned} x_t(x_0) &= x_0 + d^f(x_0,t), \\ w^f &= \frac{\partial}{\partial t} d^f, \\ \Omega_t^f &= \mathcal{A}_t(\Omega_0^f). \end{aligned}$$

2. Fluid subproblem: Find velocities and pressure such that

$$\begin{aligned} \frac{\partial}{\partial t} u^f + (u^f - w^f) \cdot \nabla u^f - \frac{1}{\rho^f} \nabla \cdot \sigma^f &= f^f \text{ in } \Omega_t^f \times (0,T), \\ \nabla \cdot u^f &= 0 \text{ in } \Omega_t^f \times (0,T). \end{aligned} \tag{5.1}$$

3. Structure subproblem: Find displacement such that

$$\frac{\partial^2}{\partial t^2} d^s - \frac{1}{\rho^s} \nabla \cdot \sigma^s = f^s \text{ in } \Omega_t^s . \times (0,T), \tag{5.2}$$

4. Coupling conditions:

$$\begin{aligned} u^f &= \frac{\partial}{\partial t} d^s \text{ on } \Gamma_t^I \times (0,T), \\ \sigma^s \cdot n^s + \sigma^f \cdot n^f &= 0 \text{ on } \Gamma_t^I \times (0,T), \\ d^f &= d^s \text{ on } \Gamma_t^I \times (0,T), \end{aligned}$$

   together with boundary conditions on the non-interaction boundaries of the subproblems.

Additionally to the possible nonlinearities of the subproblems, the coupled problem introduces different kinds of nonlinearities, e.g.:

1. the dependence of fluid displacements to the structure displacements,
2. the nonlinearity of the domain velocity of the fluid field formulation $w^f \cdot \nabla u^f$
3. the nonlinearity of structure displacements in dependence of the fluid velocities and pressure at the interface,

4. the dependence of the fluid velocities and the structure velocities at the common interface.

In a standard strong coupled monolithic approach all non-linearities are treated implicitly. This, often called fully implicit [Bar09, DP07, CDFQ11, KGF$^+$10, GV03, LTM01, BCHZ08, HHB08, Hro07, TSS06] approach, leads to a robust scheme in terms of time-step size but it is also the most expensive one. However some of the nonlinearities can be handled explicitly. In [BQQ08] a Geometry-Convective Explicit scheme is suggested. We also use a similar strategy for an explicit treatment for the inner fluid domain displacements. The used method is described in Chapter 10.

6 Geometric Conservation Laws

The Arbitrary Lagrangian Eulerian formulation introduces a domain velocity term in the convective term of the Navier Stokes equations. The order of the time discretization of the moving computational domain of the fluid can only be preserved if one considers additional conservation laws to compute the additional term. The Geometric Conservation Law was first introduced by Thomas and Lombard [TL79]. The main idea is that the way the fluid domain velocity for a moving mesh is computed should reproduce a constant fluid velocity solution or, less strictly, at least a resting fluid on a moving mesh. The meaning of the Geometric Conservation Law, that is also sometimes referred to as discrete Geometric Conservation Law, is still topic of ongoing research. It has been show by Guillard and Farhat [GF00] and Farhat et al. [FGG01] that a moving domain can lead to a less stable and accurate numerical scheme compared to the respective fixed frame approach. Formaggia and Nobile [FN04] and Boffi et al. [BG04] have proven several results on the relation of the stability and the Geometric Conservation Law.

The conservation of mass condition for an incompressible fluid in an Arbitrary Lagrangian Eulerian formulation is given by

$$\frac{d}{dt}\int_V dV + \int_{\partial V}(u-w)\cdot n\, dS = 0,$$

for an arbitrary volume V of the fluid. The volume conservation law states that the domain velocity w has to fulfill an additional conservation law. This conservation law can be directly derived using the conservation of mass in Arbitrary Lagrangian Eulerian formulation with a zero fluid velocity

$$\frac{d}{dt}\int_V dV - \int_{\partial V} w\cdot n\, dS = 0.$$

A second conservation law, the so called surface conservation law, can be derived [ZRTC93] using a constant fluid field in a an arbitrary direction a

$$\int_{\partial V} a\cdot n\, dS = 0.$$

For details we refer to [ZRTC93, Nob01, FP99], but we want to remark that this second conservation law gives some restrictions for the computation of the normals of the moving surfaces of the volume V surfaces. A violation of the geometric conservation laws may lead to an error in the mass balance of the fluid computation. A simple example shown in Figure 6.1. Here one can see a sub-control

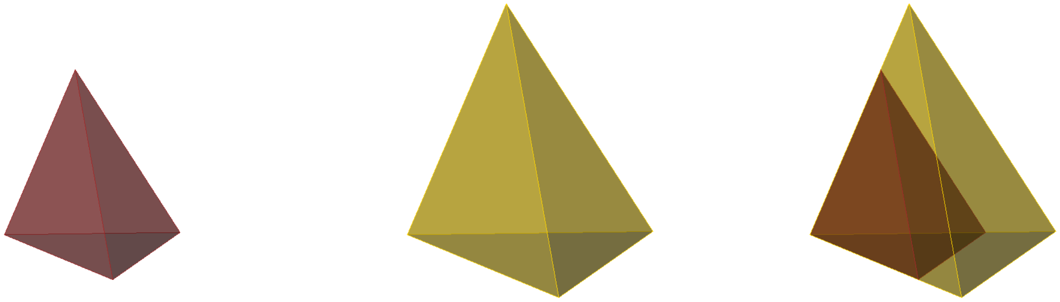


Figure 6.1. *Arbitrary volume, where one surface is moved in normal direction. The yellow part on the right indicates the volume that was added, due to the moving surface.*

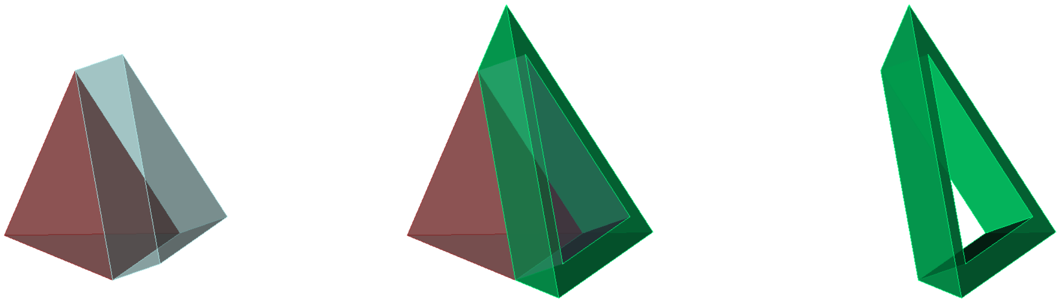


Figure 6.2. *Volume with a moving surface. Using the old configuration to compute the swept volume would lead to an divergence error (green).*

volume where one surface is swept equally in normal direction of the surface. One can imagine more advanced examples, where the mesh nodes of a surface are moved in arbitrary directions. If one simply uses the nodal grid velocities of the fluid domain to calculate the mass flux over the volume surface, conservation of mass is not necessarily ensured [FP99].

The use of the surface and the normal of the surface in the start configuration to compute

$$\int_{\partial V} w \cdot n \, dS$$

result in a divergence error, that is illustrated by the green volume in Figure 6.2. In the discrete setting we can achieve mass conservation, if the method to compute

$$\int_{\partial V} w \cdot n \, dS$$

fulfills a discrete version of the Geometric Conservation Laws. Among several different other strategies the computation of averaged quantities can resolve this problems [FG04, GF00]. The averaged configurations have to be adjusted to the used time discretization. For an incompressible fluid the

Geometric Conservation Law has an additional consequence, namely the preservation of the original mass conservation condition. To see this we apply the volume conservation law

$$\frac{d}{dt}\int_V dV - \int_{\partial V} w \cdot n \, dS = 0$$

in the mass conservation and here the first terms of the mass conservation in ALE form

$$\frac{d}{dt}\int_V dV - \int_{\partial V} w \cdot n \, dS + \int_{\partial V} u \cdot n \, dS = 0$$

cancel each other and the usual mass conservation equation remains

$$\int_{\partial V} u \cdot n \, dS = 0.$$

7 Variational formulation

In the next chapters we use triangulations build out of tetrahedron or hexahedron. An admissible triangulation fulfills the following properties: We use K tetrahedron or hexahedron with the properties that

- $\overline{\Omega} = \bigcup_{k=1}^{K} \overline{\Omega}_k$
- $\Omega_i \cap \Omega_j = \emptyset$ for $i \neq j$
- $\overline{\Omega}_i \cap \overline{\Omega}_j = \{$a common node, edge or side of T_i and $T_j\}$ for $i \neq j$.

In the following we introduce the weak form for both subproblems, and we write the coupled problem in variational formulation.

7.1 Structure subproblem

We multiply equation (5.2) by $v^s \in V^s = \{v^s \in H^1(\Omega)\}$. For sake of simplicity we ignore for now any modifications for Dirichlet values. We use the standard definitions for Hilbert, Sobolev and Lesbesgue spaces, for details we refer to [Bra07]. After integration by parts together with appropriate boundary conditions and splitting up the right hand side we we obtain the following variational problem: Find displacements d^s such that,

$$\int_{\Omega_t^s} \rho^s \frac{\partial^2}{\partial t^2} d^s \cdot v^s dV + \int_{\Omega_t^s} \sigma^s : \nabla v^s dV = \int_{\Omega_t^s} \rho^s f^s \cdot v^s dV + \int_{\partial\Omega_t^s} \rho^s t^s \cdot v^s dS. \tag{7.1}$$

7.2 Fluid subproblem

For the weak formulation of the fluid we use test and ansatz spaces on the current deformed domain. Therefore both are driven in time by the ALE mapping $\mathcal{A}_t$. We start again with multiplying equations (5.1) by $v^f \in V^f = \{v^f = \hat{v} \circ \mathcal{A}_t^{-1} | \hat{v} \in H^1(\Omega_0^f)\}$. As for the structure subproblem we omit any modifications for Dirichlet boundaries. After integration by parts we get

$$\frac{\partial}{\partial t}\int_{\Omega_t^f} u^f \cdot v^f \, dV + \int_{\partial\Omega_t^f} \{-\nu^f(\nabla u^f + (\nabla u^f)^T)n + (u^f - w^f)(u^f)^T n + p^f n\} \cdot v^f \, dS$$
$$- \int_{\Omega_t^f} -\nu^f((\nabla u^f) + (\nabla u^f)^T) + (u^f - w^f)(u^f)^T + p^f \cdot \nabla v^f \, dV - \int_{\Omega_t^f} f^f \cdot v^f \, dV = 0 \qquad (7.2)$$

and for the continuity equation

$$\int_{\partial\Omega_t^f} u^f \cdot n \cdot v^f \, dS - \int_{\Omega_t^f} (u^f)^T \cdot \nabla v^f \, dV = 0. \qquad (7.3)$$

For different choices of the test function space we obtain either a Finite Element discretization for the fluid or a Finite Volume discretization. For a H^1 continuous test function with compact support and $v(x) = 0$ for all $x \in \partial\Omega_t^f$ the boundary integral terms vanish and we get a Finite Element method, where we have to find a solution for u, p such that.

$$\frac{\partial}{\partial t}\int_{\Omega_t^f} u^f \cdot v^f \, dV - \int_{\Omega_t^f} -\nu^f(\nabla u^f + (\nabla u^f)^T) + (u^f - w^f)(u^f)^T + p^f \cdot \nabla v^f \, dV - \int_{\Omega_t^f} f^f \cdot v^f \, dV = 0$$

and

$$- \int_{\Omega_t^f} (u^f)^T \cdot \nabla v^f \, dV = 0.$$

In contrast, one can see the Finite Volume method as a Petrov-Galerkin Finite Element method with constant test functions $V^f := \{v^f = \hat{v} \circ \mathcal{A}_t^{-1} | \hat{v} \in L^2(\Omega_t^f) | \hat{v}_{|\Omega} \in \mathcal{P}_0(\Omega_t^f)\}$ [Bey98, Hau10]. Without loss of generality we chose $\mathcal{P}_0 = \{1\}$.

For this choice equation (7.2) simplifies as follows

$$\sum_k \frac{\partial}{\partial t}\int_{\Omega_k^f} u^f \cdot v^f dV + \sum_k \int_{\partial\Omega_k^f} -\nu^f(\nabla u^f + (\nabla u^f)^T)n + (u^f - w^f)(u^f)^T n + p^f n \, dS$$
$$- \int_{\Omega_k^f} f^f \, dV = 0 \qquad (7.4)$$

and for equation (7.3)

$$\sum_k \int_{\partial\Omega^f} u^f \cdot n \, dS = 0 \qquad (7.5)$$

and therefore for every k holds

$$\frac{\partial}{\partial t}\int_{\Omega_k^f} u^f dV + \int_{\partial\Omega_k^f} -\nu^f(\nabla u^f + (\nabla u^f)^T)n + (u^f - w^f)(u^f)^T n + p^f n \, dS - \int_{\Omega_k^f} f^f \, dV = 0$$

$$\int_{\partial\Omega_k} u^f \cdot n \, dS = 0.$$

7.3 Variational formulation of the coupled FSI Problem

In the following we will give a variational formulation for the coupled FSI problem. In particular we will apply the continuity of the stress coupling condition in the variational formulation to enforce a strong coupling for this condition. We have already introduced the weak structure subproblem

$$\int_{\Omega_t^s} \rho^s \frac{\partial^2}{\partial t^2} d^s \cdot v^s dV + \int_{\Omega_t^s} \sigma^s : \nabla v^s dV = \int_{\Omega_t^s} \rho^s f^s \cdot v^s dV + \int_{\partial \Omega_t^s} \rho^s t^s \cdot v^s dS,$$

as well as the weak fluid subproblem

$$\frac{\partial}{\partial t} \int_{\Omega_k^f} u^f dV + \int_{\partial \Omega_k^f} -\nu^f (\nabla u^f + (\nabla u^f)^T) n + (u^f - w^f)(u^f)^T n + p^f n \, dS - \int_{\Omega_k^f} f^f \, dV = 0$$

$$\int_{\partial \Omega_k} u^f \cdot n \, dS = 0.$$

Formally we will use the continuity of the stress coupling condition as a Neumann boundary condition for the structure and we apply the kinematic coupling condition as a Dirichlet boundary condition for the fluid as done in [Bar09]. The concept is motivated by the fact that the fluid acts as a force on the structure while the structure provides the deformed configuration of the fluid boundaries. However we want to stress that these conditions are not boundary conditions, but coupling conditions at the interface.

The sum of the two terms $\nu^f(\nabla u^f + (\nabla u^f)^T)n$ and $p^f n$ at the fluid structure interface has the physical interpretation of a force acting on the structure, i.e.,

$$\int_{\Gamma_t^I} \nu^f (\nabla u^f + (\nabla u^f)^T) n - p^f n \, dS = \int_{\Gamma_t^I} t^f dS.$$

In a partitioned coupling scheme it is possible to compute the fluid interface load as a residual of the fluid variational formulation. In the context of our monolithic approach we directly assemble a force from the fluid in the structure equation. The strong coupling of the force may lead to an unstable numerical coupling [FLLT98, LTM01] depending on the used discretizations.

A numerically stable coupling can be achieved choosing matching test and shape functions at the fluid structure interface [Nob01, Cro11, Ast10] We replace $\int_{\Gamma_t^I} t \cdot v^s dS$ in the equation for the structure problem and use the fact that the forces acting on the fluid structure interface cancel since $\int_{\Gamma_t^I} t^s dS = \int_{\Gamma_t^I} t^f dS$. We then obtain the following

$$\int_{\Omega^s} \rho^s \frac{\partial^2}{\partial t^2} d^s \cdot v^s dV + \int_{\Omega^s} \sigma^s : \nabla v^s dV = \int_{\Omega^s} \rho^s f \cdot v^s dV + \int_{\partial \Omega^s \setminus \Gamma_I} \rho^s t^s \cdot v^s dS + \int_{\Gamma_t^I} \sigma^f n^f \cdot v^s dS. \quad (7.6)$$

Since we use different discretizations for fluid and structure, we want to draw attention to the fact that in equation (7.6) the structure test function have to be used in the last addend. The force coming from the fluid acting on the structure can then be applied by using $\sigma^f(u^f, p^f) = -p^f I + 2\nu^f \varepsilon^f(u^f)$, which in three dimensions is given by

$$\sigma^f(u^f,p^f) = \begin{pmatrix} -p^f & 0 & 0 \\ 0 & -p^f & 0 \\ 0 & 0 & -p^f \end{pmatrix} + \nu^f \begin{pmatrix} 2\frac{\partial u_1^f}{\partial x} & \frac{\partial u_2^f}{\partial x} + \frac{\partial u_1^f}{\partial y} & \frac{\partial u_3^f}{\partial x} + \frac{\partial u_1^f}{\partial z} \\ \frac{\partial u_1^f}{\partial y} + \frac{\partial u_2^f}{\partial x} & 2\frac{\partial u_2^f}{\partial y} & \frac{\partial u_3^f}{\partial y} + \frac{\partial u_2^f}{\partial z} \\ \frac{\partial u_1^f}{\partial z} + \frac{\partial u_3^f}{\partial x} & \frac{\partial u_2^f}{\partial z} + \frac{\partial u_3^f}{\partial y} & 2\frac{\partial u_3^f}{\partial z} \end{pmatrix}.$$

The velocity coupling is mainly driven by the assumption of a no-slip condition. In the case of a stationary interface this condition reduces to the standard no-slip condition. In case of a moving interface the condition is given the velocities of fluid and structure have to match at the interface

$$u^f = \frac{\partial}{\partial t} d^s.$$

This coupling condition is enforced node by node as a Dirichlet like condition in the fluid. Again, however we remark that since this is an interface condition it depends on the respective velocity variable of the structure. We have used different frameworks for fluid and structure and as a consequence we have to deal with an additional coupling condition, that is a no penetration condition at the interface for a possible auxiliary problem

$$d^f = d^s.$$

This coupling condition is also enforced node by node as a Dirichlet like condition for the auxiliary fluid domain problem. As for the velocity coupling, this interface condition depends on the respective deformation variable of the structure.

8 Time discretization

In Chapter 7 we have introduced the weak formulation of the coupled fluid structure interaction problem. From a mathematical point of view, this problem is a set of algebraic differential equations for an unknown u. In its general form we can write this problem as

$$\begin{aligned} \frac{\partial u}{\partial t} + N(u) &= f \text{ on } \Omega_t \times (0,T) \\ A(u) &= 0. \end{aligned}$$

N is a non-linear operator and u a vector. We use reduction of order for the structure equation, therefore we have two first order differential equations for the structure. In the following we ignore the term $A(u) = 0$, for simplicity of notation, since this constrained can be easily incorporated for an implicit time discretization. In this chapter we will introduce different time integration schemes that have been implemented for the coupled fluid structure interaction problem. Furthermore we will discuss restrictions to some of these time stepping schemes for both, in a moving and a fixed framework.

In computational fluid dynamics and structure simulations many different time schemes are used, from one-step methods to multistep methods. The first decision for a time discretization scheme is the choice of an explicit or implicit method. An explicit method has the benefit that it is less expensive per time step than an implicit method, where a system of equation has to be solved. For fluid problems, one can show that the maximum stable time step is bounded by a Courant-Friedrichs-Lewy (CFL) Condition. For the incompressible Navier Stokes equations the CFL condition in the one dimensional case reads as the following

$$\frac{u\tau}{h} \leq 1$$

where h denotes the mesh size and τ the time step size. In an implicit and A-stable method the time step size is not bounded by the CFL condition. However it is well-know that also an implicit scheme can be restricted in the time-step size. The Crank-Nicolson Scheme for example, an implicit second order scheme, is restricted in the time-step size on a fixed domain [HR90, For73]

$$\tau \leq ch^{2/3}$$

with a c constant c. For higher order schemes on moving domains instabilities due to the Arbitrary Lagrangian Eulerian formulation have been observed [FGG01]. In Wick [Wic13] one can find a result on the stability for the incompressible Navier Stokes equations in Arbitrary Lagrangian Eulerian formulation using the Crank-Nicolson scheme.

In the following we give a short introduction to the numerics of ordinary differential equations, with an emphasis on Runge Kutta methods. An ordinary differential equation for an initial value problem is given by

$$y'(t) = f(y,t)$$

with some initial values $y(0) = y_0$. Let $\tau > 0$ be the time step size and let $t_i = t_0 + i\tau$ be the i-th time step. For the sake of simplicity we use equidistant time stepping, but of course adaptive time stepping is also possible. Starting with

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} y'(t)dt$$

we use a quadrature rules with nodes $c_1,...,c_2$ and weights $b_1,...,b_s$

$$y(t_{n+1}) = y(t_n) + \tau \sum_{i=1}^{s} b_i y'(t_n + c_i \tau).$$

Inserting the ordinary differential equation $y'(t) = f(y,t)$ we get

$$y_{n+1} = y_n + \tau \sum_{i=1}^{s} b_i f(y_{n,i}, t_{n,i})$$

Here we set $t_{n,i} = t_n + c_i \tau$ and $y_{n,i} = y(t_{n,i})$. For each step in the Runge Kutta scheme one has to compute

$$y_{n,i} = y_n + \int_{t_n}^{t_{n,i}} y'(t)dt$$

using quadrature rules with nodes $c_1,...,c_2$ and weights a_{ij}

$$y_n + \int_{t_n}^{t_{n,i}} y'(t)dt = y_n + \tau \sum_{j=1}^{s} a_{ij} f(y_{n,j}, t_{n,j})$$

Now we can write the Runge Kutta method as

$$y_{n+1} = y_n + \tau \sum_{i=1}^{s} b_i f(y_{n,i}, t_{n,i}),$$

$$y_{n,i} = y_n + \tau \sum_{j=1}^{s} a_{ij} f(y_{n,j}, t_{n,j}).$$

The coefficient for Runge Kutta method can be written in a Butcher array

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array}$$

For example for $s = 1$ we get the backward Euler method

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

or written as an equation:

$$y_{n+1} - y_n = \tau f(y_{n+1}, t_{n+1}).$$

For $s = 2$ we get the Crank-Nicolson method with the following choice for the coefficients

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

or written as an equation

$$y_{n+1} - y_n = \frac{\tau}{2}(f(y_n, t_n) + f(y_{n+1}, t_{n+1})).$$

For a different choice of coefficients we can get another second order scheme, which is a so called DIRK (Diagonal Implicit Runge Kutta scheme).

$$\begin{array}{c|cc} \alpha & \alpha & 0 \\ 1 & 1-\alpha & \alpha \\ \hline & 1-\alpha & \alpha \end{array}$$

with $\alpha = 1 - \frac{1}{2}\sqrt{2}$. This scheme is computed in two separate sub-time steps. In this thesis we have implemented the implicit coupled fluid structure problem for the three presented time discretization schemes. Special care is need for the discretization of the fluid domain velocity. For details we refer to Chapter 6.

9 Spatial discretization

As mentioned in the last chapter, we deal with a non-linear partial differential equation of the form

$$\frac{\partial u}{\partial t} + N(u) = f \text{ on } \Omega_t \times (0,T).$$

In this chapter we discretize the non-linear operator $N(u)$. The most popular methods are the Finite Difference, the Finite Element and the Finite Volume method. They all have in common that the computational domain is subdivided by a grid. The Finite Difference method is the oldest discretization method of those mentioned above. Here usually the space is equally decomposed and the operator $N(u)$ is approximated with finite differences. The solution vector is usually collocated at the grid nodes. For complex geometries however this approach is not very suitable, since for unstructured or non-uniform grids assembling the finite difference operator is rather complicated. However, when a structured grid can be employed, high order Finite Difference methods are still often used. Finite Element and Finite Volume methods are often applied in combination with unstructured grids. For Finite Elements an extensive mathematical theory was developed during the second half of the last century. The Finite Elements method, just as the Finite Volume method, is based on a weak or variational formulation. As already discussed in Chapter 7 the operator is multiplied with a test function. After integration by parts and using Greens formula the special properties of the test functions are used. For the Finite Element formulation we get volume terms, for the Finite Volume discretization we get boundary terms for the operator $N(u)$. These boundary terms are usually approximated by the computation of fluxes over control volumes. Since the Finite Volume method can be also seen as a Petrov-Galerkin Finite Element method it shares many of the features of Galerkin Finite Element methods [Ode91]. One of the most considerable advantages of the Finite Volume method compared to the Finite Element method is the (local) conservation property of the numerical fluxes, which is, that the numerical flux is conserved between neighboring control volumes [Bey98]. Therefore this method is used in many fields where conservation of physical quantities is needed, such as heat and mass transfer as well as fluid mechanics. Standard Galerkin methods usually do not have a local conservation property [Hir07]. For convection dominated problems with steep gradients Finite Elements and Finite Volumes tend to produce numerical solutions with oscillations [Joh09, QV08, RST08]. So-called upwind stabilization methods for convection dominated problems have been developed [Joh09, RST08, GR05] to avoid these oscillations. Many of them can be used for both Finite Element an Finite Volume methods. However, for Finite Vol-

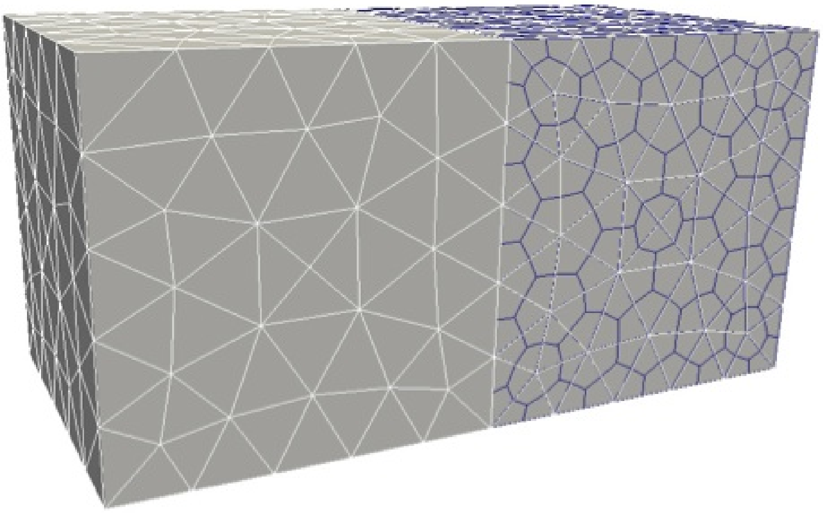


Figure 9.1. *Finite Element mesh (left) and Finite Volume mesh (right) with the constructed dual mesh on the right hand side.*

umes schemes specially tailored schemes have been also developed [LDV96] using e.g artificial diffusion and a variation of standard upwind methods. For both the Finite Element and the Finite Volume method the discretization of the Navier Stokes equations result in a non symmetric stiffness matrix. The fully coupled fluid structure interaction system is also non symmetric. We use different preconditioning strategies, a multigrid and a two-level additive Schwarz method. The development of robust solver for this kind of problems is challenging [BW97, Pfl99, HP97]. For details on the solver strategies used in this work we refer to Chapter 12. In this work we use different discretization schemes for the structure and fluid. On the fluid domain we use the Finite Volume method while on the structure domain we apply a Finite Element discretization. In this chapter we discuss the discretization of the structure equation and the fluid equations using their respective discretization technique. For the fluid we also describe the employed stabilization. Finally we describe the discretization of the coupling terms for velocities and forces.

For both, the Finite Element method and the Finite Volume method, we use the same ansatz functions. Since the structure is formulated in Lagrange coordinates, the shape functions do not change in time. Due to the Arbitrary Lagrangian Eulerian formulation of the fluid the shape functions on the fluid sub-domain are driven by the Arbitrary Lagrangian Eulerian mapping. However, for a fixed time t they meet the same properties as in the non moving case. Therefor and for simplicity we use the same notation in this chapter for the time independent and time dependent shape functions. We use linear independent basis functions $\phi_1, \phi_2, ..., \phi_n$ to define V_h

$$V_h := \text{span}\{\phi_i\}_{i=1}^{n_p}$$

here n_p is the number of grid nodes. These piecewise linear or trilinear functions, depending on the element type have the following properties:

- ϕ_i has compact support, i.e. it is non zero only in the neighboring elements of the node x_i,
- ϕ_i is continuous and bounded,
- $\phi_i(x_j) = \begin{cases} 1, & \text{for } i = j \\ 0, & \text{else.} \end{cases}$

For the sake of simplicity we do not use the superscripts s and f in this chapter whenever it is clear to which subproblem the variables belong.

9.1 Structure - Finite Elements

On the structure sub-domain we use a Finite Element discretization. Finite Element methods are the discretization techniques often used in structural mechanics. For a detailed description of the Finite Element method we refer to [Cia02]. In contrast to, e.g., a Finite Difference method, a Finite Element method is suitable for complex geometries with curved boundaries. The basic idea in a Finite Element method is to use a weak formulation for the problem and one looks for an approximation of the solution in a finite dimensional space. We start from the variational formulation introduced in Chapter 7.1

$$\int_{\Omega^s} \rho^s \frac{\partial^2}{\partial t^2} d^s \cdot v^s dV + \int_{\Omega^s} \sigma^f : \nabla v^s dV = \int_{\Omega^s} \rho^s f \cdot v^s dV + \int_{\partial\Omega^s} \rho^s t^s \cdot v^s dS.$$

First, we split the boundary force term into two terms according to a decomposition of the boundary into the fluid structure interface and its complement. The first term is the usual structure boundary where the externally applied force acts. The second is at the fluid structure interface, where a force from the fluid acts on the structure

$$\int_{\partial\Omega^s} \rho^s t^s \cdot v^s dS = \int_{\partial\Omega^s \setminus \Gamma_I} \rho^s t^s \cdot v^s dS + \int_{\Gamma_I} \rho^s t^s \cdot v^s dS.$$

In Chapter 7 we replace the last term by the acting force of the fluid on the structure

$$\int_{\partial\Omega^s} \rho^s t^s \cdot v^s dS = \int_{\partial\Omega^s \setminus \Gamma_I} \rho^s t^s \cdot v^s dS + \int_{\Gamma_I} \sigma^f n^f \cdot v^s dS.$$

For the discretization of the last term we refer to the force coupling section of this chapter.

We define

$$K^s(d^s, v^s) := \int_{\Omega^s} \sigma^s : \nabla v^s dV,$$

and

$$M^s(d^s, v^s) := \rho^s \int_{\Omega^s} d^s \cdot v^s dV.$$

For the external forces we define the linear form

$$f^s(v^s) := \int_{\Omega^s} \rho^s f^s \cdot v^s dV + \int_{\partial\Omega^s} \rho^s t^s \cdot v^s dS.$$

With this notation we can write the whole system in matrix form

$$M^s \ddot{d}_h + K^s d_h = f^s_h,$$

using the representation

$$d_h = \sum_{i=1}^{n_p} d_i \phi_i$$

in terms of the basis functions $\{\phi_i\}_j$.

For hexahedron e we can write the local nodal displacement vector

$$d_e = \begin{pmatrix} d_{e_1} \\ d_{e_2} \\ d_{e_3} \\ d_{e_4} \\ d_{e_5} \\ d_{e_6} \\ d_{e_7} \\ d_{e_8} \end{pmatrix}$$

where d_{e_i} is the local displacement vector at node i and therefore each d_{e_i} contains three components. We define the element shape function matrix as

$$N = \begin{pmatrix} N_1 & N_2 & N_3 & N_4 & N_5 & N_6 & N_7 & N_8 \end{pmatrix},$$

where each N_i is given by local shape functions

$$N_i = \begin{pmatrix} \phi_i & 0 & 0 \\ 0 & \phi_i & 0 \\ 0 & 0 & \phi_i \end{pmatrix}.$$

The element mass matrix is computed using the element shape function matrix

$$M_e = \int_{\Omega_e} \rho^s N^T N \, dV.$$

In a similar way we define a strain matrix

$$B = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 & B_8 \end{pmatrix}.$$

The element stiffness matrix is computed using the strain-displacement matrix

$$B_i = \begin{pmatrix} \frac{\partial \phi_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial \phi_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial \phi_i}{\partial z} \\ \frac{\partial \phi_i}{\partial y} & \frac{\partial \phi_i}{\partial x} & 0 \\ 0 & \frac{\partial \phi_i}{\partial z} & \frac{\partial \phi_i}{\partial y} \\ \frac{\partial \phi_i}{\partial z} & 0 & \frac{\partial \phi_i}{\partial x} \end{pmatrix}.$$

The element stiffness matrix contribution is then given by

$$K_e = \int_{\Omega_e} B^T E B \, dV$$

where E is the elasticity matrix, containing the Lamé constants

$$E = \begin{pmatrix} \lambda^s + 2\mu^s & \lambda^s & \lambda^s & 0 & 0 & 0 \\ \lambda^s & \lambda^s + 2\mu^s & \lambda^s & 0 & 0 & 0 \\ \lambda^s & \lambda^s & \lambda^s + 2\mu^s & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu^s & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu^s & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu^s \end{pmatrix}.$$

For the element force vector holds

$$f_e = \int_{\partial \Omega_e} N^T f dS.$$

9.2 Fluid - Finite Volume

The Finite Volume approach is the natural choice when dealing with conservation laws of physics, e.g. the conservation of mass, energy or momentum. We chose a collocated arrangement for velocities and pressure. Without stabilization this usually results in an (unphysical) checkerboard distribution for the pressure, therefore we use a stabilization proposed by Schneider and Raw [Raw85, SR87] which was later extended by Karimian and Schneider [KS95]. The description of the incompressible Navier Stokes equations using the stabilization of Schneider and Raw with the stabilization extensions used in this thesis is based on the description used by[Näg04, RR96] . In the literature about Finite Volume methods on can find several different variants

- Vertex-centered Finite Volume methods,
- Cell-centered Finite Volume methods,
- Cell-Vertex centered Finite Volume methods.

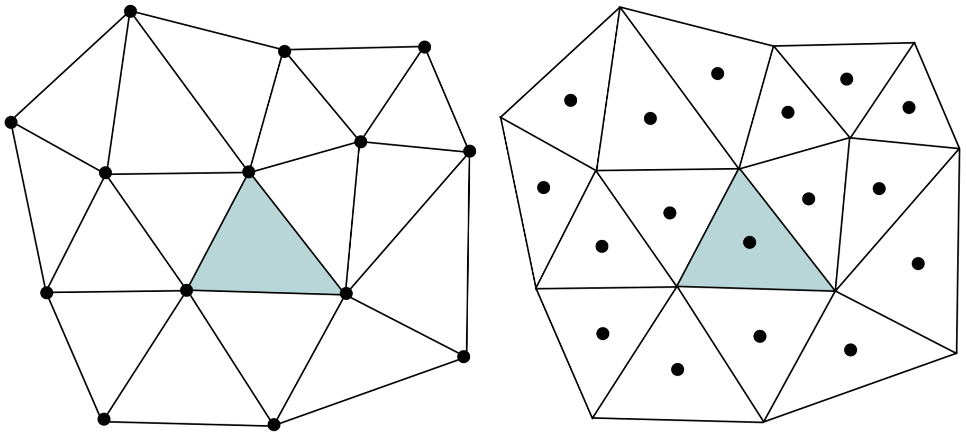


Figure 9.2. *Construction of control volumes for the Finite Volume method. Left: cell-centered. Right: vertex-centered.*

The main difference between these variants lies in their way of constructing control volumes, for a sketch we refer to Figure 9.2 and Figure 9.3. In the following we give a summary of the discretization of the collocated arrangement of unknowns in our vertex centered Finite Volume method and the used stabilization. In this discretization all degrees of freedom are located in the nodes of the standard Finite Element grid. The basic idea is to decompose the domain Ω^f into k sub-domains, called control volumes, each containing a node of the grid. There are different ways to construct such a tessellation. In two dimensions we can use the normals of the edges of the original grid to construct a dual box-grid. This method is restricted to the case where all interior angle are smaller or equal to $\frac{\pi}{2}$ for details we refer to [Bey98]. In an alternative approach we can use the center of gravity and the midpoint of the edges of an element. This method works for all interior angles and can be applied in three dimensions. In this scheme a triangle is split up into three commensurate parts.

The integration over a control volume is written as the sum over the belonging sub-control volumes (SCV). Accordingly the integral over the boundary of a control volume is the sum over all sub-control volume surfaces ($SCVF$) of the control volume. Figure 9.3 shows the used quantities for the Finite Volume discretization. The so-called integration points ip are located on the center of mass at the sub-contol volume surfaces. We denote a node of the triangulation by x. We want to remark that the following terms depend on values at x and ip. Later we will give a short overview how these terms can be modified with upwind strategies.

In the following for the sake of simplicity a normal n_j on a sub-control volume surface is scaled with the area of the sub-control volume surface and n_{SCV} denotes the number of sub control volumes belonging to a specific node. Although we write all terms as sums over the nodes, the implementation of the assembling routines is carried out over the elements. This means we compute the contribution to a node for each element and its contribution to the global matrix. In the following we approximate each term of equation 7.4 and equation 7.5 for details on this we refer to [Näg04].

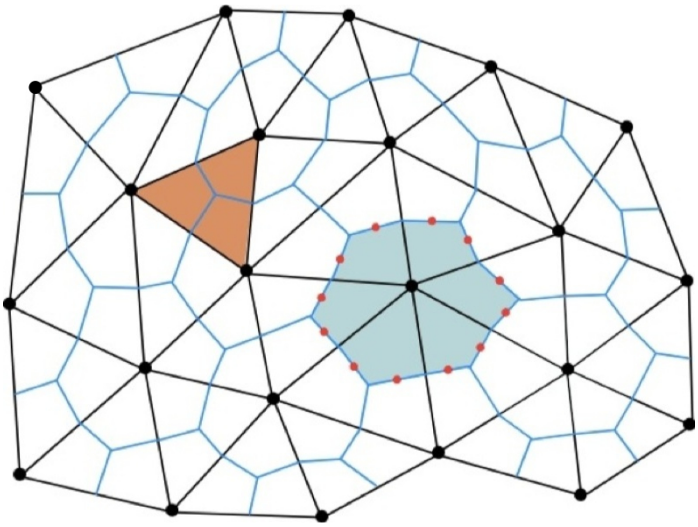


Figure 9.3. *We use a vertex centered Finite Volume scheme with control volumes based on dual boxes. We generate for each node sub control volumes that are combined to a control volume (blue) for a node p . We introduce on the sub control volume boundary faces additional integration points (red). The assembling process is carried out on the underlying Finite Element grid. On each element we compute a contribution to each node, that is added to the global mass and stiffness matrices.*

- The first fluid term which is approximated by computing the sum of the sub control volumes,

$$\frac{\partial}{\partial t} \sum_{j=1}^{n_{SCV(k)}} \int_{SCV_j} u \, dV \approx \frac{\partial}{\partial t} \sum_{j=1}^{n_{SCV(k)}} |SCV_j| u(x_i).$$

- Diffusion

$$- \sum_{j=1}^{n_{SCVF(k)}} \int_{SCVF_j} -\nu(\nabla u + (\nabla u)^T) n \, dS \approx - \sum_{j=1}^{n_{SCVF(k)}} \sum_{i=1}^{n_p} \nu(\nabla \phi_i(ip_j) u(x_i) + (\nabla \phi_i(ip) u(x_i))^T) n_j.$$

- Pressure gradient

$$\sum_{j=1}^{n_{SCVF(k)}} \int_{SCVF_j} pn \, dS \approx \sum_{j=1}^{n_{SCVF(k)}} \sum_{j=i}^{n_p} \phi_i(ip_i) p(x_i) n_j.$$

- Source term

$$\sum_{j=1}^{n_{SCV(k)}} \int_{SCV_j} f \, dV \approx \sum_{j=1}^{n_{SCV(k)}} |SCV_j| f(x_i).$$

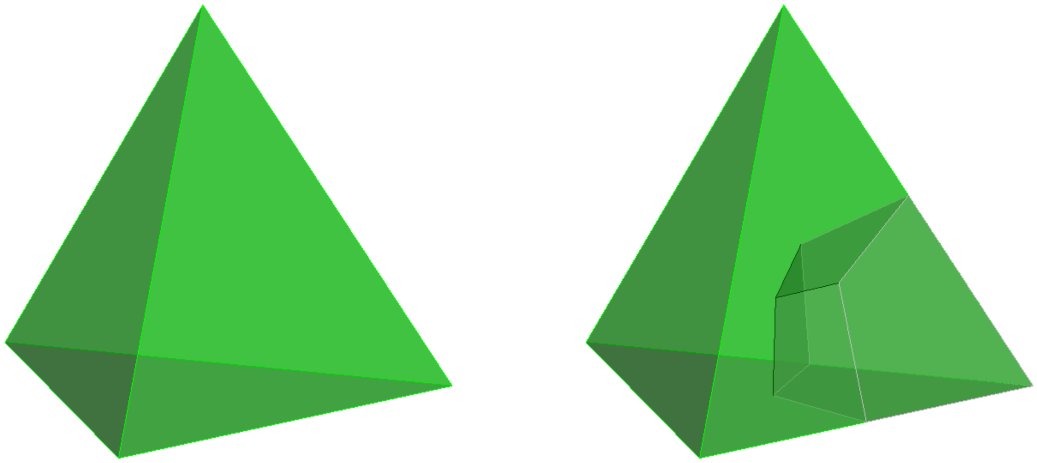


Figure 9.4. *Tetrahedral element and the corresponding sub-control volume for a node.*

- Continuity equation

$$\frac{\partial}{\partial t} \sum_{j=1}^{n_{SCVF(k)}} \int_{SCVF_j} u \cdot n \, dS \approx \sum_{j=1}^{n_{SCVF(k)}} u(ip_j) \cdot n_j.$$

- Convective term

$$\sum_{j=1}^{n_{SCVF(k)}} \int_{SCVF_j} (u-w)u^T n \, dS \approx \sum_{j=1}^{n_{SCVF(k)}} (u(ip) - w^f(ip))u(ip)^T n_j.$$

We use upwind techniques to formulate these terms depending on a value at the corresponding node. For details we refer to upcoming section about different upwinding schemes and the modifications for the convective term. There are several methods to compute a divergence free fluid domain velocity. The exact computation is expensive, therefore approximation strategies have to be used in practice because the computation of an approximation for the swept boundaries of the sub control volumes is often cheaper. Besides the exact computation of swept volumes we use an approach where we use averaged geometric quantities [KF99, FGG01, FG04] to construct a schemes which satisfies the geometric conservation laws. For the computation of the grid velocity and normal for the Crank-Nicolson scheme we use: First, we compute an averaged normal n_a

$$n_a = \frac{n(x^t) + n(x^{t+\tau})}{2}$$

then we compute $w \cdot n$ using the averaged normal

$$w \cdot n = \frac{x^{t+\tau} - x^t}{\tau} \cdot n_a$$

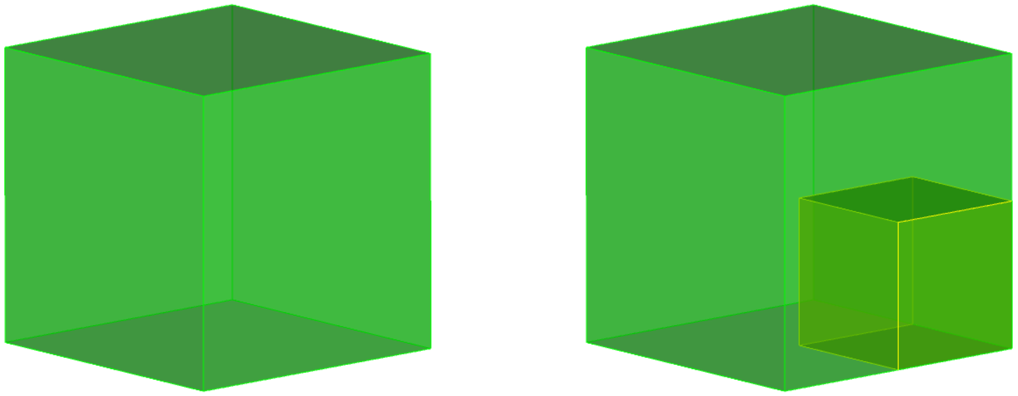


Figure 9.5. *Hexahedral element and the corresponding sub-control volume for a node.*

To compute the velocities at the integration points we use the standard shape functions. For details and schemes for different time discretizations we again refer to [KF99, FG04] and references therein.

9.2.1 Stabilization of the fluid discretization

We use a Finite Volume method where all unknowns are located at the nodes. This results in an unstable discretization for the fluid, if the values at the integration-points are only interpolated with shape functions. We use a stabilization that was developed by Schneider and Raw [SR87] and later extended by Schneider and Karimian [KS95]. Therein the interpolation is done using the dependence of velocity and pressure arising from the momentum equation. In the following we give a short summary of the used methods. Details and comparison of the methods of Schneider and Raw and Schneider and Karmian can be found in [NW07]. The code we use is based on an implementation by [RR96, Näg04]. In the following we give an overview of the possible stabilizations variants that are included in our fluid structure implementation. For the stabilization of the discrete system we use upwind methods. There are various approaches specially tailored for e.g Finite Elements [GR05, QV08] or Finite Volumes [BBFS90, LDV96, LeV02]. As suggested in Karminan and Schneider [KS95] we apply a local momentum equation to interpolate the values at the nodes to the integration points. With the modification from Schneider and Raw [SR87] we get

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} - \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + \frac{\partial p}{\partial x_i} = u_i \, div(u). \tag{9.1}$$

The crucial point is the development of an appropriate dependence of velocities and the pressure. The stabilization approach of Schneider and Raw consists of two parts. First a finite difference approach is used to discretize the momentum equation and then the linearization of the convective term using an upwind scheme. The right hand side of equation 9.1 can be written as

$$u_i\, div(u) = u_j \frac{\partial u_i}{\partial x_j} - \sum_{j=1, j\neq i}^{3} (u_j \frac{\partial u_i}{\partial x_j} - u_i \frac{\partial u_j}{\partial x_j}).$$

We follow [NW07, Hau10] and discretize and approximate on each node via shape functions to obtain

$$u_i\, div(u) \approx ||\tilde{u}|| \frac{u_i^{dn} - u_i^{up}}{L_d} - \sum_{j=1, j\neq i}^{3} \sum_{k}^{n_p} (\tilde{u}_j \frac{\partial \phi_k}{\partial x_j} u_i(x_k) - \tilde{u}_i \frac{\partial \phi_k}{\partial x_j} u_j(x_k)).$$

Here L_d is the distance between the upwind point introduce the upwind and downwind points y^{up} to the downwind point y^{dn} and $||\tilde{u}||$ is the absolute value of the solution of the velocity of the last time step. The velocity u^{up} denotes the velocity at the upwind point y^{up}, for a sketch of the used upwind points we refer to Figure 9.6. A downwind point is the equivalent to an upwind point in downwind direction. The local momentum equation now can be written as

$$\begin{aligned} \frac{u_i - u_i^{t-1}}{\tau} + ||\tilde{u}|| \frac{u_i - u_i^{up}}{L_c} - \nu \frac{\sum_{k=1}^{n_p} \phi_k u_i(x_k) - u_i}{L_d^2} + \sum_{k=1}^{n_p} \frac{\partial \phi_k}{\partial x_i} p(x_k) - ||\tilde{u}|| \frac{u_i^{dn} - u_{ip}^{up}}{L_d} \\ + \sum_{j=1, j\neq i}^{3} \sum_{k=1}^{n_p} (\tilde{u}_j \frac{\partial \phi_k}{\partial x_j} u_i(x_k) - \tilde{u}_i \frac{\partial \phi_k}{\partial x_j} u_j(x_k)) = 0. \end{aligned} \tag{9.2}$$

Here L_c and L_d denote the distance of the convective and diffusive up and down point and u_i^{t-1} is the solution of the velocity of the last time step.

9.2.2 Diffusion term

For the diffusion term

$$\nu \frac{\sum_{k}^{n_p} \phi_k u_i(x_k) - u_i}{L_d^2}$$

in equation 9.2 we use the stabilization that was introduced by [SR87] and modified by [Näg04] and [RR96] to compute L_d^2

$$\frac{1}{L_d^2} = \frac{2||n_{min}||^2}{||SCV_0||^2} + \frac{8}{3||n_{averg}||^2}.$$

Here, $||n_{min}||^2$ is the minimum of the norms of the normal that belong to an element and$||n_{averg}||^2 = \frac{1}{n_{co}} \sum_{k=1}^{n_c} ||n_k||^2$. Finally we end up in a formulation for the nodal quantities of the form

$$u_i(ip) = \sum_{k}^{n_p} (C_u(x_k) u_i(x_k) + C_p(x_k) \frac{\partial p}{\partial x}(ip) + C_t(x_k) u_i^{t-1}(x_k)),$$

where the constants C_{u_i}, C_p and C_t denote the dependencies of the integration point velocities on the nodal velocities, for details we refer to [Näg04].

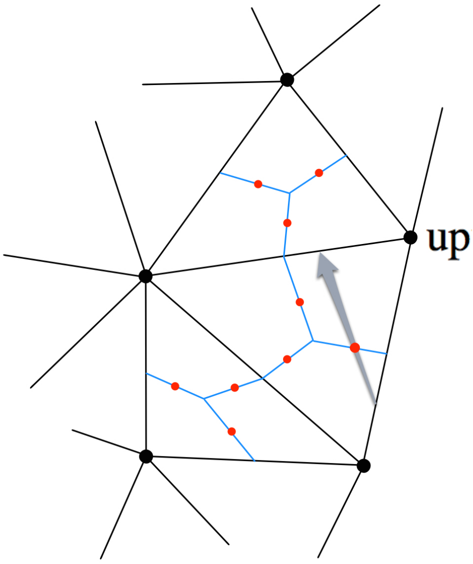


Figure 9.6. *Full upwind scheme (UDS).*

9.2.3 Upwind strategy

The discretization of the convective term may end up with some zero entries on the diagonal. If the time-step size is small enough this is compensated by entries of the mass matrix and the diffusive terms. However, for convective dominant flows this may lead to difficulties and to oscillations in the solution. When we compute the contributions at an integration point with bilinear shape functions, the sum is weighted equally in all directions. For convective problems this does not take into account the physical anisotropy. Using upwind techniques one can weight upstream nodes more than downstream nodes. One can derive upwind method using Tailors expansion

$$u_{ip} = u^{up} + (x^{up} - x_{ip}) \cdot \frac{\partial u}{\partial x}|^{up} + \mathcal{O}(||x^{up} - x_{ip}||^2)$$

In a full-upwind scheme one choses the point as an upwind point that belongs to the sub-control volume that is in upstream direction. This scheme is simple but introduces an artificial diffusion. We can construct second order schemes by adding a correction term to the upwind integration point, which is at least a first order approximation of the derivative of u

$$u_{ip} = u^{up} + (x^{up} - x_{ip}) \cdot \frac{\partial u}{\partial x}|^{up} = u^{up} + \Delta u_{ip}$$

In literature one can find several second order schemes, e.g. central difference scheme, where the derivation of u is approximated using a combination of a left and right finite difference scheme. For

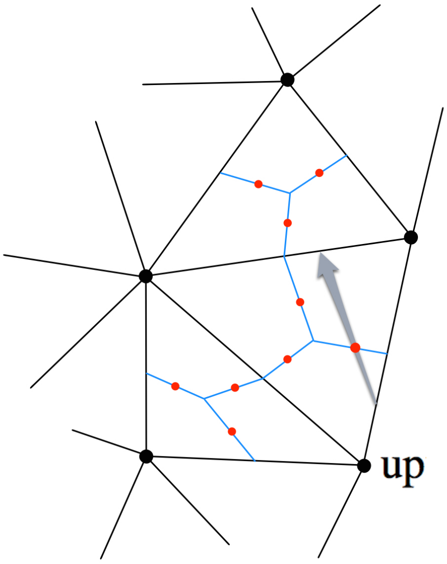


Figure 9.7. *Skewed Upwind scheme (SKP).*

a special choice of weights one results in the so-called Quick scheme. In upwind schemes the upstream point usually gets a bigger weight than the downstream point. For some choices this results in negative weights or in unphysical dependencies of downstream integration points. Especially for unstructured grids the implementation of a Quick scheme, that introduces dependencies to neighboring elements can be quite demanding. In the following we present the approaches used in this work.

- The first scheme we employ is a very simple one. The strategy here is to look for the upwind point constructing a line in downstream direction. This line intersects an element side. The closest node in upstream direction to this intersection is used as upwind point. This situation is shown in Figure 9.6. We call this scheme full upwind scheme (UDS).

- The second scheme used in this work is the Skewed Upwinding (SKP). The strategy here is to look for the upwind point constructing a line in upstream direction. This line intersects an element side, the closest node to this intersection is used as upwind point. A sketch is shown in Figure 9.7

- The third skew scheme is Linear Profile Skewed Upwinding (LPS). Here an intersection with the upstream element side is computed as in Figure 9.8. In a second step the surrounding node of the element side are weighted according to the values of the shape functions of the upwind point. Usually this results in a stable upwind scheme, however for some special cases this can introduce negative coefficients.

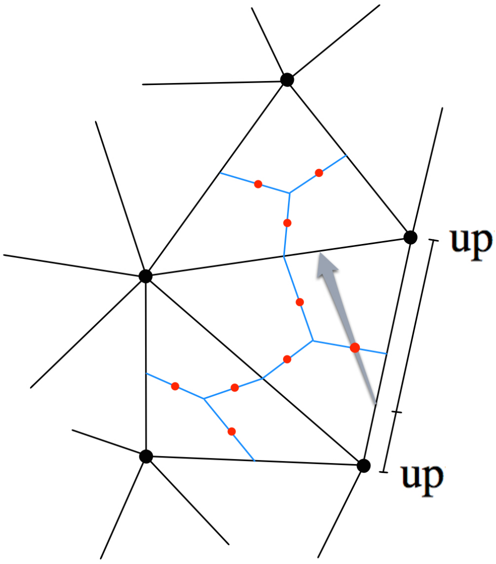


Figure 9.8. *Linear Profile Skewed Upwinding scheme (LPS).*

- The last upwind scheme in this work is the Positive Upwinding (POS). Here the mass-flux at each integration point is computed. The basic idea is that for an incompressible flow the outflow of a control volume is just driven by the inflow. For this upwinding scheme we can distinguish between three cases:
  - In the first case $m_{in} = 0$. Therefore the flow direction has to point from the node in the sub control volume. in that situation we choose this node to be the upwind point.
  - In the second case $0 < \frac{m_{in}}{m_{out}} < 1$ we have to weight between the nodes and the integration point.
  - In the last case $\frac{m_{in}}{m_{out}} \geq 1$ we put all upwind contribution to the inflow integration point.

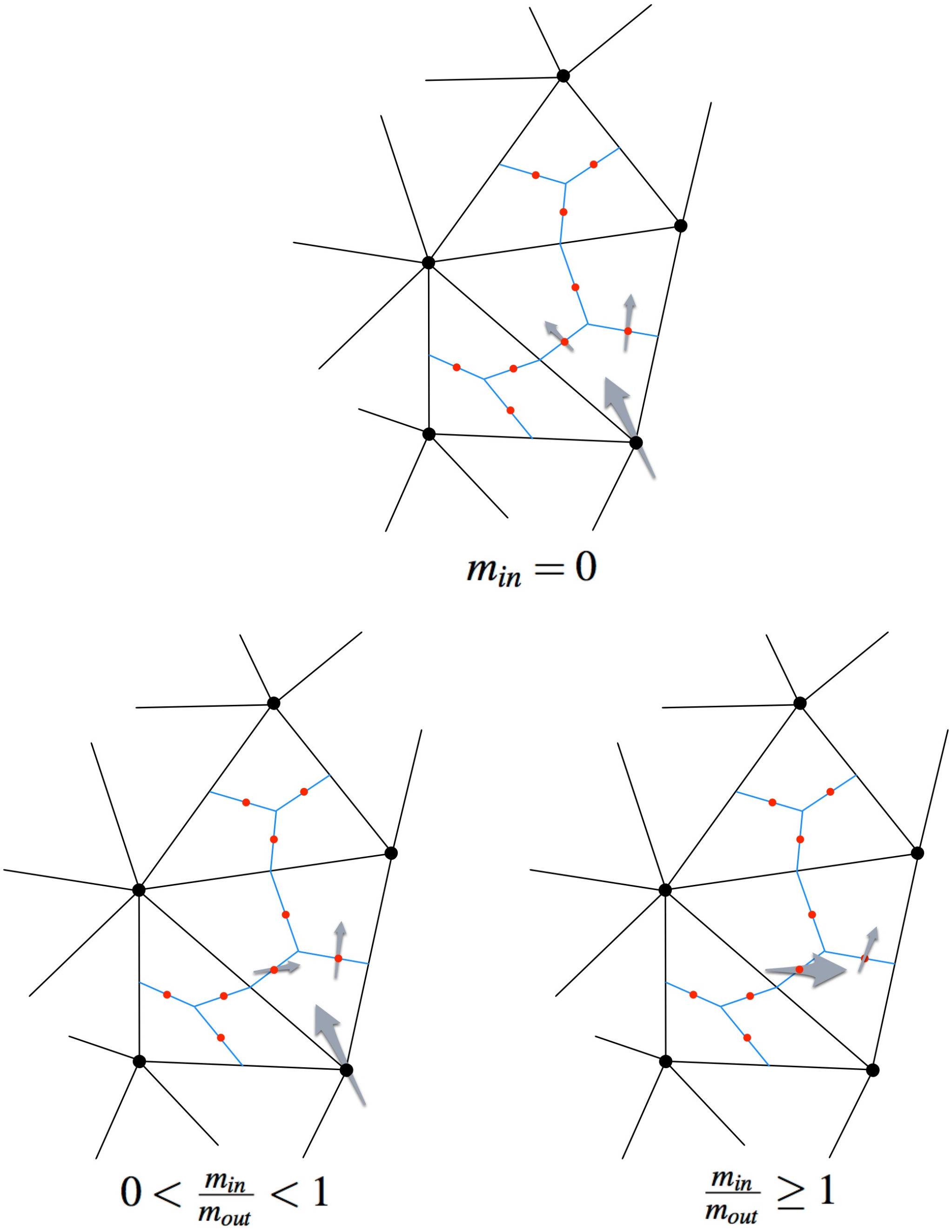
  for details and the meaning of m_{in} and m_{out} we refer to Figure 9.9.

9.2.4 Convective Term

The convective term without the fluid domain velocity is discretized by

$$\sum_{j=1}^{n_{SCVF(k)}} \int_{SCVF_j} uu^T n dS \approx \sum_{j=1}^{n_{SCVF(k)}} u(ip)u(ip)^T n_j.$$

We want to remark that the domain velocity of the fluid domain gives an additional contribution to the velocity u. The contribution from the domain velocity is computed as a nodal quantity and



$m_{in} = 0$

$0 < \frac{m_{in}}{m_{out}} < 1$

$\frac{m_{in}}{m_{out}} \geq 1$

Figure 9.9. *Positive Upwinding scheme (POS).*

then subtracted from the stabilized velocity using standard shape function for the interpolation to the integration point velocities. The fluid domain velocity gives an additional contribution to each integration point. As suggested in [Näg04, RR96] we use a physical advection correction (PAC) for the convective term. The PAC scheme is a scaled version of pure upwinding. Again the idea is to solve local Navier Stokes equations to couple the nodal values and the integration point values. We can rewrite the relation given by equation (9.2) as

$$u_i(ip) = (\frac{1}{\tau} + \frac{\nu}{L_d^2} + \frac{||\tilde{u}||}{L_c})^{-1} \frac{||\tilde{u}||}{L_c} u_i^{up} +$$
$$(\frac{1}{\tau} + \frac{\nu}{L_d^2} + \frac{||\tilde{u}||}{L_c})^{-1} (\frac{u_i^{t-1}}{\tau} + \frac{\nu}{L_d^2} \sum_{k=1}^{n_p} \phi_k u_i(x_k) - \sum_{k=1}^{n_p} \frac{\partial \phi_k}{\partial x_i} p(x_k)).$$

We simplify the notation using

$$u_i^{up} = (\frac{1}{\tau} + \frac{\nu}{L_d^2} + \frac{||\tilde{u}||}{L_c})^{-1} \frac{||\tilde{u}||}{L_c} u_i^{up}$$

and

$$\delta u_i = (\frac{1}{\tau} + \frac{\nu}{L_d^2} + \frac{||\tilde{u}||}{L_c})^{-1} (\frac{u_i^{t-1}}{\tau} + \frac{\nu}{L_d^2} \sum_{k=1}^{n_p} \phi_k u_i(x_k) - \sum_{k=1}^{n_p} \frac{\partial \phi_k}{\partial x_i} p(x_k)).$$

This approach can be seen as a skew upwind method

$$u_i(ip) = u_i^{up} + \delta u_i,$$

for details we refer to [RT74, Näg04]. In this work a convex combination of two approaches is used

$$u_i(ip) = \omega u_i^{up} + (1-\omega) \sum_{k=1}^{n_p} \phi_k(ip) u_i(x_k).$$

For details on this scheme we refer to [RR96]. In this scheme u_i^{up} may be either a pure upwind method or the upwind approximation of the PAC upwind scheme. The parameter ω can be chosen for each element depending on the relation of convection and diffusion, i.e.,

$$\omega = \frac{Pe^2}{5 + Pe^2}$$

with the Peclet number $Pe = \frac{u_n L}{\nu}$.

9.3 Boundary conditions

Our implementation can handle different types of boundary conditions. On the structure subproblem, one can apply time dependent displacements and forces. These Dirichlet and Neumann conditions are standard for Finite Elements in structure mechanics. The Dirichlet boundary conditions

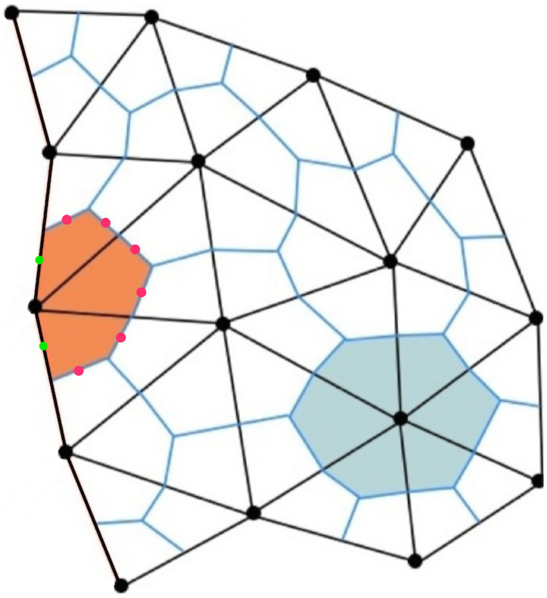


Figure 9.10. *At the boundary each node has just a half control volume (orange) compared to an inner node with a full control volume (blue). Therefore additional integration points (green) are constructed. The boundary integration points however can be treated as the normal integration points.*

is enforced by replacing the respective row of the algebraic system with a trivial condition. The Neumann values are assembled into the right hand side.

The fluid discretization is based on a vertex-centered approach. The control-volumes are based on the dual grid. When dealing with boundary integrals we have to take into account a different situation compared to the inner control-volumes. At each boundary control-volume we have to introduce additional boundary integration points as shown in Figure 9.10. These boundary integration points can be and are handled as normal integration points, i.e., the boundary integration points provide a contribution to the flow over the boundary of a sub-control volume. So the fluxes over the surfaces of a control volume are computed as,

$$\int_{\partial CV} un_i \, dS \approx \sum_{ip} un_i + \sum_{bip} un_i,$$

where *bip* denotes the boundary integration points. We use no-slip conditions for the velocities at the walls Γ_t^{wall}. The velocity at Γ_t^{wall} is directly set as a Dirichlet condition at the nodes. In case of a non moving wall we enforce

$$u(x_i) = u^{wall}(x_i).$$

For a moving wall we set the according relative velocity.

Inflow boundary conditions are given by Dirichlet constrains at the boundary. The inflow boundary is given by a flow profile, i.e. a given mass flux. The pressure at the boundary is free, in the

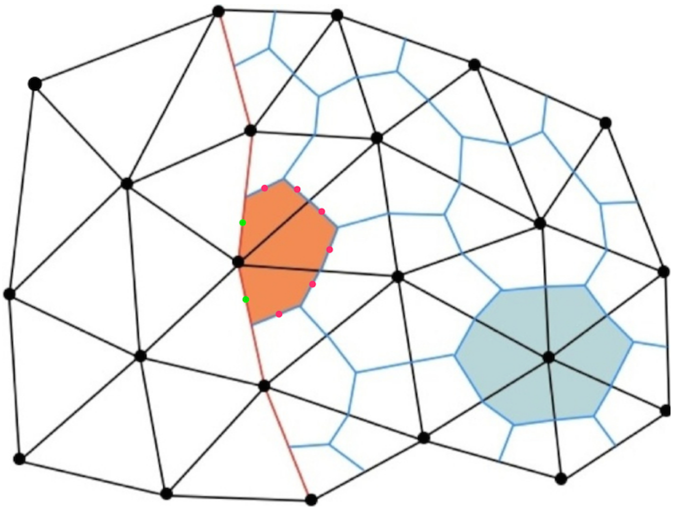


Figure 9.11. *At the fluid structure interface (red polygon) we are in a similar situation as at the domain boundary. Here we also construct additional fluid structure interface integration points (green), that are used as boundary integration points.*

sense that it is independent of the given inflow velocity. In the equation of momentum of the Navier Stokes equations the inflow value for the specific boundary node is directly used

$$u(x_i) = u^{in}(x_i) \text{ on } \Gamma_t^{in}.$$

For most of the computations we use a pressure outflow condition, i.e., set a Dirichlet value for the pressure

$$p(x_i) = 0 \text{ on } \Gamma_t^{out}.$$

For a laminar flow this outflow condition assures an equal outflow.

9.4 Discretization of the coupling conditions coupling

Since we use conforming meshes, the degrees of freedom for fluid and structure are located at the same places and we can enforce the coupling conditions at the same nodes. The velocity coupling at the fluid structure interface can be seen as a special case of a wall condition of the fluid. In this case we modify the entries in the matrix, without setting a value on the right hand side. The corresponding value is set by the coupling assembler that modifies also the entry for the respective structure velocity. For these modifications we refer to Chapter 11. For the dynamic coupling we assemble an additional term that acts as a force of the fluid on the structure. For clarity we want to remark that this is not a boundary condition, but an interface condition

$$\int_{\Gamma^I} \sigma^s n \cdot v^s \, dS = -\int_{\Gamma^I} \sigma^f n \cdot v^s \, dS \tag{9.3}$$

$$\tag{9.4}$$

The following term is assembled on the fluid structure interface

$$\int_{\Gamma^I} \nu(\nabla u^f + (\nabla u^f)^T)n + p^f n \, dS = \int_{\Gamma^I} t^f \, dS. \tag{9.5}$$

Since we use the same shape function for the Finite Element and Finite Volume discretization the shape function stays the same. Although this term is a fluid force, it is multiplied with the same test function as used in the structure domain. The force is composed of a velocity and a pressure component

$$\sigma^f(u,p) = \begin{pmatrix} -p & 0 & 0 \\ 0 & -p & 0 \\ 0 & 0 & -p \end{pmatrix} + \nu \begin{pmatrix} \frac{2\partial u_1}{\partial x} & \frac{\partial u_2}{\partial x} + \frac{\partial u_1}{\partial y} & \frac{\partial u_3}{\partial x} + \frac{\partial u_1}{\partial z} \\ \frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x} & \frac{2\partial u_2}{\partial y} & \frac{\partial u_3}{\partial y} + \frac{\partial u_2}{\partial z} \\ \frac{\partial u_1}{\partial z} + \frac{\partial u_3}{\partial x} & \frac{\partial u_2}{\partial z} + \frac{\partial u_3}{\partial y} & \frac{2\partial u_3}{\partial z} \end{pmatrix}.$$

The nodal forces for an structure element at the interface are therefore assembled as

$$\int_{\Gamma^I} \sigma^f n \cdot v^s \, dS \tag{9.6}$$

where for n we use the outer normal of the structure domain and σ^f denotes the discrete viscous stress tensor. This term is assembled using the same quadrature rules as are used for the local structure stiffness matrix.

9.5 Summary of the discretization

We have introduced different time stepping schemes in Chapter 8, the Backward Euler, the Crank-Nicolson, and a second order DIRK scheme. For the discretization of the fluid domain velocity we use a geometric averaging approach. This is suitable for all three time stepping schemes.

In this chapter we have presented the spacial discretization for structure and fluid. The assembly for both subproblems is performed element by element. Our implementation supports tetrahedra and hexahedra organized in an unstructured conforming multi-level grid structure. Especially tetrahedra in an unstructured grid are suitable for real word problems, since almost arbitrary shaped geometries can be meshed. We use a conforming mesh for fluid and structure, that contains the same type of elements. The discretization of the structure is done using standard $\mathcal{P}_1$ Finite Elements. For the fluid we use a specially tailored discretization. We compute a dual mesh, that provides a control volume for each node. On these control volumes we use a Finite Volume method for the discretization of the fluid. The specific feature of this discretization is that the degrees of freedom for the fluid, velocities and pressure, are all located at the nodes. Due to this non-staggered scheme a stabilization is needed. We use a stabilization based on an idea of Schneider and Raw [SR87], that was later improved by Karimian and Schneider [KS95]. Here a local momentum equation is solved at each element. This gives a relation between the degrees of freedom for the fluid velocity and the pressure at the nodes and the integration points. The diffusion is discretized using an idea of Raw [Raw85]. Some improvements were suggested in the work of Rentz-Reichert [RR96] and Nägele [Näg04]. Furthermore we have presented a special upwind method based on physical advection correction for the convective term. The stabilization adds an additional term to the continuity equation. For a uniform grid with mesh size h, this would add a term of the form

$$Ch^2\Delta p$$

with a constant C that depends on the treatment of the convective term.

The force coupling condition is discretized as a stress acting on the structure using standard Finite Element shape and test functions. For the velocity coupling we modify the entries in the system matrix as the handling of Dirichlet conditions. This is done for both fluid and structure. A similar modification is done for the coupling of the geometry problem to the structure.

10 Mesh motion and geometric coupling

The fluid structure interface moves in the course of time. There are different techniques available to keep track of the interface e.g Immersed Boundary method or Arbitrary Lagrangian Eulerian method. The Arbitrary Lagrangian Eulerian method is widely used and maybe the most popular method for fluid simulations on moving domains. In the Arbitrary Lagrangian Eulerian formulation the boundaries of the computational domain for the fluid may move and the positions of the mesh nodes in the inner of the computational domain have to be updated to avoid tangled or inverted elements and to maintain a reasonable mesh quality. This mesh update is not prescribed and can be constructed in manifold ways.

In the following we give a short overview of possible strategies for the fluid domain mesh update. For a more comprehensive overview we refer to [Wic11] and the references therein. The main goal in the arbitrary movement of the inner of the fluid domain is to keep a valid mesh with good properties. The meaning of good properties for a mesh are not immediately clear and depend on the application. Usual criteria are smoothness, local mesh angles, aspect ratio, and orientation. The main task for the mesh movement process is to retain the properties of the initial mesh as good as possible. Our solver strategy for the full coupled fluid structure interaction problem is based on the use of iterative solvers. The rate of convergence of iterative solvers depends on the spectral radius of the matrices constituting the system of equations. A poor mesh quality affects directly the spectral radius of this matrices and will therefore have an impact on the solver efficiency in the simulation.

We distinguish between two different approaches for the geometry subproblem: A geometric implicit and a geometric explicit one [BTT13, CDFQ11]. In both cases the physically motivated coupling conditions and the geometric coupling are treated implicitly. The difference is in the treatment of the fluid mesh motion of the interior mesh nodes. In the geometric implicit case the mesh update is computed simultaneously with the physical coupling for fluid velocities and structure displacements and velocities. For the implicit handling of the domain movement one can find different strategies in literature, based on solving a Laplace problem [Bar09, Wic11], an elastic equation [Wic11, BS06, STB03, TBMJ92] or biharmonic equation [Wic11, Hel03]. The strategy of solving an additional elastic equation for the fluid motion has been found to be able to deal with moderate and large deformation, if the material parameter are chosen in a reasonable way. This usually means that the more closer to the fluid structure interface the more stiffer material parameters for the auxiliary elasticity problem should be chosen. However, the use of an Laplacian strategy for the mesh adaptation is also suitable for many applications of interest, such as blood flow simulation [Bar09]. The geometric explicit case is different to these mesh adaptation procedures. It is based on an extrapolation for the fluid domain displacements, primarily to avoid tangled and distorted elements on the first layers of elements at the moving boundary or interface.

10.1 Geometric implicit strategy

For the implicit handling of the fluid domain motion we follow two different strategies. In the first, the mesh displacements have to satisfy the Laplace equation. In the second one an elasticity equation is used. In the first case we assemble a Laplace problem into the system matrix

$$-\Delta d^f = 0$$

with the coupling conditions

$$d^f = d^s \text{ on } \Gamma^I_t.$$

For the second strategy a stationary elasticity problem is assembled, with the same coupling condition as for the Laplace problem. In this approach we can use different elastic material parameter, which may have a large impact on the mesh deformations, especially close to the fluid structure interface. Here we solve a problem of the form

$$-\nabla \cdot \sigma(d^f) = 0$$

and the coupling condition

$$d^f = d^s \text{ on } \Gamma^I_t.$$

Both the Laplacian and the elasticity auxiliary problem are used in a weak sense and we use Finite Elements for the discretization. For the details of the used Finite Elements discretization we refer to Chapter 9.

10.2 Geometric explicit strategy

In the geometric explicit approach we split the monolithic problem in to two sub-systems that can be solved independently. Similar methods have been previously successfully applied for fluid structure interaction simulations [BTT13, BQQ08, CDFQ11]. Instead of solving an auxiliary problem for the fluid mesh motion we fix displacements in the inner of the fluid domain. The interface displacements are computed fulfilling the geometric coupling condition. In a second step we compute an update in the interior of the fluid domain and an extrapolation of the new positions of the fluid mesh. The mesh adaptation is carried out using the Mesh Quality Improvement Toolkit (Mesquite) [BFDK$^+$03]. Mesquite is a mesh quality improvement tool for tetrahedral, hexahedral, and hybrid structured and unstructured meshes. In our implementation the mesh adaptation uses different mesh optimization algorithms and provides parallel smoothing and untangling. For our purposes we get the best performance by alternating the use of swapping and smoothing methods. We use multilevel

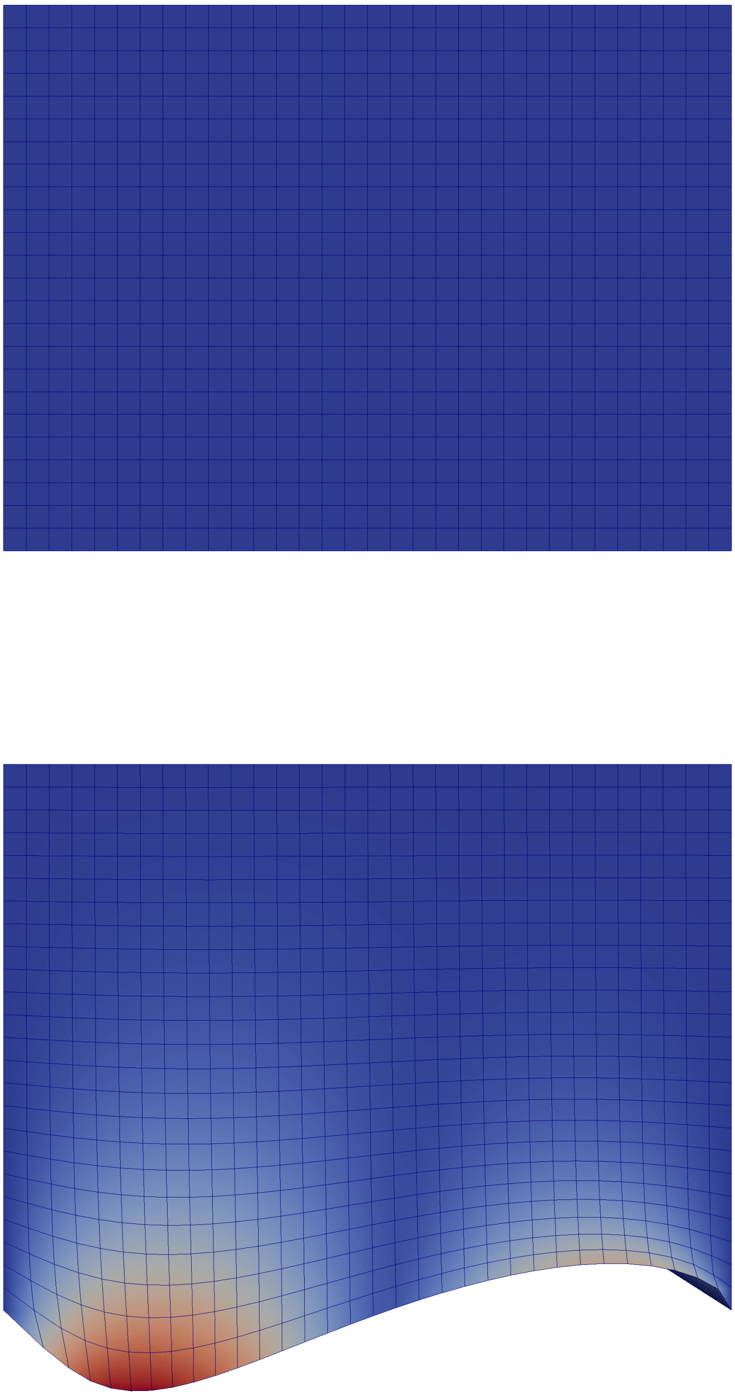


Figure 10.1. *Undeformed fluid domain (top) and implicit smoothed fluid domain with warped deformation, the color scale indicates magnitude of the fluid domain deformations (below). One can see that the deformations at the boundary are propagated to the interior of the domain.*

methods for the preconditioning of the iterative solver, therefore we can apply the mesh adaptation process on different levels. This makes the geometric explicit method even more appealing in the case of moderate displacements in a time step. The smoothing can be used very efficiently on the coarsest level while the computed deformations are interpolated to the finer grids. In addition, the geometric explicit method reduces the number of Newton steps. For details we refer to Chapter 14. In summary a time step has the following structure:

1. Compute an extrapolation for the fluid domain velocities using the displacements at the fluid structure interface. We compute the extrapolation for the fluid domain velocity using the structure interface displacements

   $$d_t^s = d_t^f \text{ on } \Gamma_t^I$$

   We smooth the distribution of nodes in the fluid domain, which provides new positions in Ω_{t+1} and correspondingly a displacement vector $d_{t+\tau}^f$

   $$w_{t+\tau} = \frac{d_{t+\tau}^f - d_t^f}{\tau}$$

2. Solve the coupled problem with coupling conditions at the interface

   $$\begin{aligned} u_t^f &= \frac{\partial}{\partial t} d_t^s \\ \sigma_t^s \cdot n_t^s + \sigma_t^f \cdot n_t^f &= 0 \\ d_t^s &= d_t^f \end{aligned}$$

3. Next time step

Finally we want to remark that the geometric explicit strategy restricts the deformation at the fluid structure interface per time step. For larger deformation of the layer of elements of the fluid domain at the interface, we get usually in increase of linear iterations in the solution process or even worse tangled elements.

11 Fully coupled system

In this chapter we consider the structure of the discretized monolithic coupled system in more detail. We always use the implicit coupling for the kinematic and dynamic coupling condition. The displacement coupling is also done implicitly but we use different strategies for the fluid domain displacements. For the fully implicit treatment of the fluid domain displacements we use either a stationary linear elastic auxiliary problem (with different material properties, depending on he problem) or we solve a vectorial Laplace problem. For the explicit treatment of the fluid domain displacements we split the solving process into two separate steps. In this chapter we will explain the differences in the coupling schemes in detail using the Jacobian of the Newton method.

11.1 Implicit coupled system

We use a fully implicit coupling for the two physical coupling conditions and the displacement coupling conditions. The coupling of velocities is carried out by employing a trivial modification in the system matrix for the velocities at the fluid structure interface. The force coupling is done by assembling stresses that act on the structure. The fluid domain displacement problem is coupled by enforcing the equality of positions at the fluid structure interface to the displacements of the structure. The following system gives an overview of the coupling concept

$$\begin{pmatrix} \text{Fluid} & 0 & \text{Velocity coupling} \\ 0 & \text{Fluid geometry problem} & \text{Fluid displacement coupling} \\ \text{Force coupling} & 0 & \text{Structure} \end{pmatrix} \begin{pmatrix} u^f_{t+\tau} \\ d^f_{t+\tau} \\ d^s_{t+\tau} \end{pmatrix} = \begin{pmatrix} f_f \\ 0 \\ f_s \end{pmatrix}.$$

In this schematic representation we have summarized the structure of degrees of freedom to the single variable d^s and all fluid degrees of freedom to the variable u^f. In the following we will keep the notation for the fluid degree of freedom u^f, while we use for the geometry fluid domain problem d^f and split the degrees of freedom for the structure into displacements d^s velocities v^s. The system contains 10 degrees of freedom for each mesh node, three for the fluid velocities, one for the fluid pressure, three for displacements of fluid and structure and three for structure velocities.

After linearization of the non-linear problem we get a linear system. The following system is obtained using a backward Euler time discretization. The force coupling is carried out by assembling a block matrix for the stresses of the fluid that act on the structure [Bar09]. In our formulation

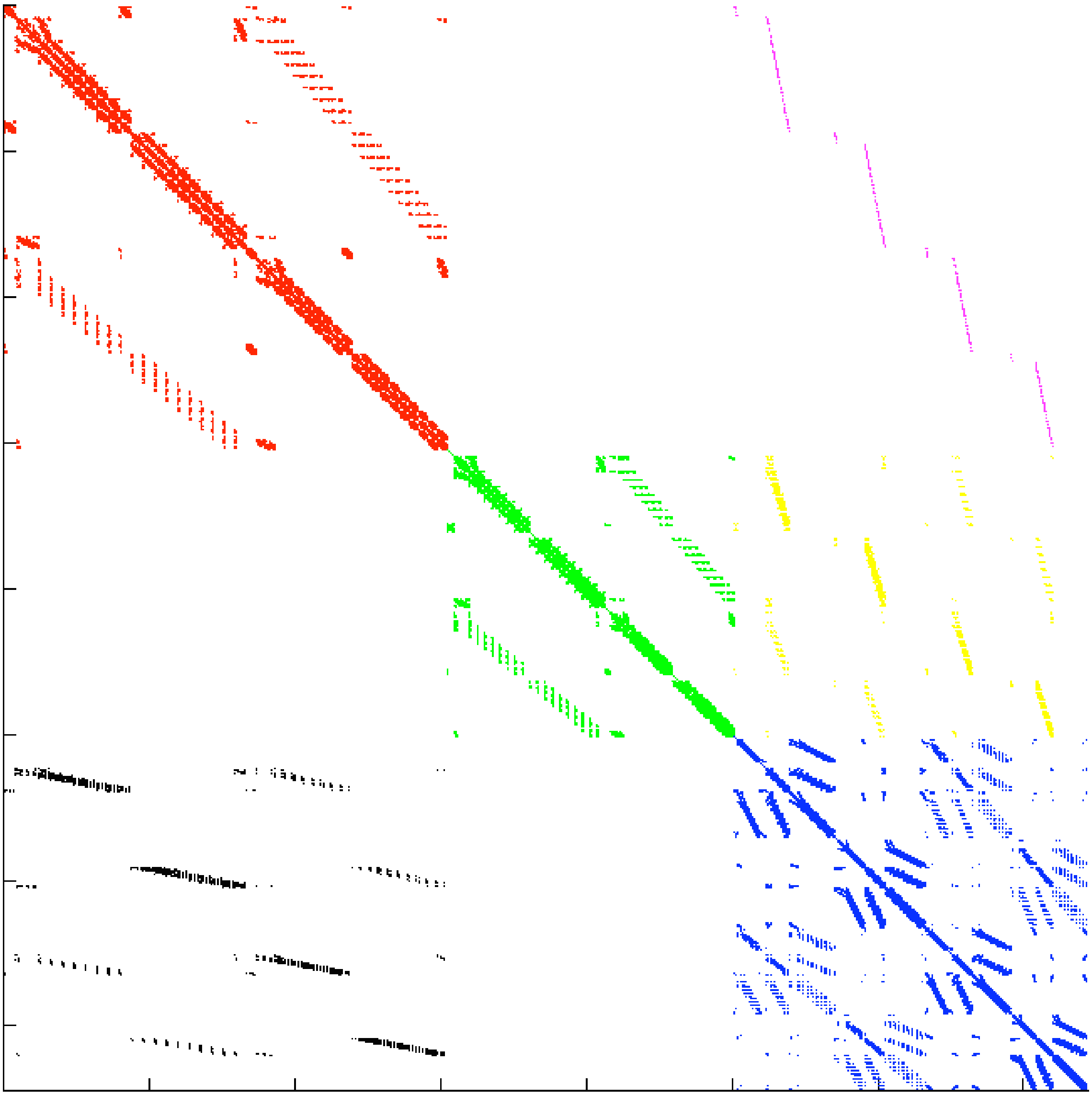


Figure 11.1. *System matrix for the geometrical implicit coupled system ordered by equation: red is the fluid system, green the Laplace equation for the fluid domain smoothing process, blue the structure domain. The non diagonal blocks are the coupling terms, starting from top with the velocity coupling (magenta), then the position coupling (yellow) for the fluid domain smoother and finally the force coupling (black).*

we ignore the derivatives of the fluid with respect to the fluid domain displacements. The removal of some matrix entries leads to a quasi Newton method and we lose the quadratic convergence behavior of the Newton method. Especially the derivatives of the fluid to the fluid domain motion can be computationally expensive [Cro11]. The full sparsity pattern of the coupled system of a simple test case is shown in Figure 11.1. In the following equation the degrees of freedom are ordered by equations and we use the same color as we use in equation Figure11.1.

$$\begin{pmatrix} F^{ff} & F^{f\Gamma^I} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I^{\Gamma^I} & 0 & 0 & 0 & 0 & 0 & -I^{\Gamma^I} \\ 0 & 0 & G & G & 0 & 0 & 0 & 0 \\ 0 & 0 & G & G & 0 & -I & 0 & 0 \\ 0 & 0 & 0 & 0 & M^{ss} & M^{s\Gamma^I} & \tau M^{ss} & \tau M^{s\Gamma^I} \\ 0 & 0 & 0 & 0 & M^{\Gamma^I s} & M^{\Gamma^I\Gamma^I} & \tau M^{\Gamma^I s} & \tau M^{\Gamma^I\Gamma^I} \\ 0 & 0 & 0 & 0 & -\tau K^{ss} & -\tau K^{s\Gamma^I} & M^{ss} & M^{s\Gamma^I} \\ 0 & \tau A^f & 0 & 0 & -\tau K^{\Gamma^I s} & -\tau K^{\Gamma^I\Gamma^I} & M^{\Gamma^I S} & M^{\Gamma^I\Gamma^I} \end{pmatrix} \begin{pmatrix} u^f_{t+\tau} \\ u^{f\Gamma^I}_{t+\tau} \\ d^f_{t+\tau} \\ d^{f\Gamma^I}_{t+\tau} \\ d^s_{t+\tau} \\ d^{s\Gamma^I}_{t+\tau} \\ v^s_{t+\tau} \\ v^{s\Gamma^I}_{t+\tau} \end{pmatrix} = \begin{pmatrix} f^f \\ f^{f\Gamma^I} \\ 0 \\ 0 \\ f^s \\ f^{s\Gamma^I} \\ f^s \\ f^{s\Gamma^I} \end{pmatrix}.$$

The red block is the fluid system, green the geometry problem for the fluid domain smoothing process, blue the structure domain block. The non diagonal blocks are the coupling terms, starting from top with the velocity coupling (magenta), then the position coupling for the fluid domain smoother (yellow) and finally the force coupling (black). The structure blocks are obtained using reduction of order for the structure equation.

11.2 Geometric explicit system

We have to deal with different kind of nonlinearities in the coupled system. In our experiments we have observed that the geometry implicit coupling introduces a strong nonlinear behavior for many examples. Therefore, we have implemented a geometric explicit coupling. The following equation shows the modified system in the geometric explicit case

$$\begin{pmatrix} F^{ff} & F^{f\Gamma^I} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_{\Gamma^I} & 0 & 0 & 0 & 0 & 0 & -I_{\Gamma^I} \\ 0 & 0 & I & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & -I & 0 & 0 \\ 0 & 0 & 0 & 0 & M^{ss} & M^{s\Gamma^I} & \tau M^{ss} & \tau M^{s\Gamma^I} \\ 0 & 0 & 0 & 0 & M^{\Gamma^I s} & M^{\Gamma^I\Gamma^I} & \tau M^{\Gamma^I s} & \tau M^{\Gamma^I\Gamma^I} \\ 0 & 0 & 0 & 0 & -\tau K^{ss} & -\tau K^{s\Gamma^I} & M^{ss} & M^{s\Gamma^I} \\ 0 & \tau A^f & 0 & 0 & -\tau K^{\Gamma^I s} & -\tau K^{\Gamma^I\Gamma^I} & M^{\Gamma^I s} & M^{\Gamma^I\Gamma^I} \end{pmatrix} \begin{pmatrix} u^f_{t+\tau} \\ u^{f\Gamma^I}_{t+\tau} \\ d^f_{t+\tau} \\ d^{f\Gamma^I}_{t+\tau} \\ d^s_{t+\tau} \\ d^{s\Gamma^I}_{t+\tau} \\ v^s_{t+\tau} \\ v^{s\Gamma^I}_{t+\tau} \end{pmatrix} = \begin{pmatrix} f^f \\ f^{f\Gamma^I} \\ 0 \\ 0 \\ f^s \\ f^{s\Gamma^I} \\ f^s \\ f^{s\Gamma^I} \end{pmatrix}.$$

Here we use the strategy which is already explained in Chapter 10. For clarity we repeat the basic ideas. At the fluid structure interface we employ the coupling of displacements to the fluid domain as we do in the implicit case. However, in the interior of the fluid domain we keep the position fixed. In our method we use extrapolated positions of the fluid domain vertices using all positions and an extrapolated velocity of the fluid domain. In each time step we use a Laplacian smoother (Mesquite) to retain the mesh quality of the fluid domain

$$\begin{pmatrix} -G & -G \\ 0 & I \end{pmatrix} \begin{pmatrix} d^{f}_{t+\tau} \\ d^{f\Gamma^I}_{t+\tau} \end{pmatrix} = \begin{pmatrix} 0 \\ f^{f\Gamma^I} \end{pmatrix}.$$

The use of a geometric explicit coupling is beneficial to reduce the number of Newton steps. For details and a detailed comparison to the full implicit coupling we refer to Chapter 14.

12 Solver and preconditioner

In this chapter we address solution methods for the coupled algebraic system presented in Chapter 11. Especially in three dimensions design of a scalable solution process for the monolithic coupled algebraic system is very demanding, because the system is large and it is usually ill conditioned. The main challenge is to find a suitable preconditioning strategy for a parallel solver method. In Chapter 8 we have introduced a class of Runge Kutta methods for time propagation. In each time step at least one nonlinear system has to be solved. In this work Newton's method is used for the linearization of the nonlinear problem and an iterative method to solve the linearized problem. We use a Krylov subspace method and different multilevel methods for preconditioning. In Algorithm 1 the general solution process is sketched. We start without loss of generality with $t = 0$ and use one of the time stepping schemes introduced in Chapter 8.

Algorithm 1 Sketch of the employed solution algorithm

```
$t = 0$
while $t \leq T$ do
  for $n = 1, \dots,$ max nonlinear steps do
    assemble nonlinear defect $d_k$
    if $d_k \leq tol_{nl}$ then
      break
    else
      solve linear problem
      compute new correction $c$
      update solution
    end if
  end for
  $t = t + \tau$
end while
```

In the following we give a detailed overview of this solution process.

12.1 Quasi Newton method

The discretization of the coupled system leads to a large system

$$N(x) = b$$

with a nonlinear Operator $N(x)$, that has to be solved in each time step. We define the nonlinear defect

$$d(x) := N(x) - b = 0$$

We can now use Taylor expansion to approximate the function $F(x) = N(x) - b$ in the vicinity of x_0 with a function $L(x)$. For the construction of $L(x)$ we use

$$L(x) := F(x_0) + J(x_0)(x - x_0)$$

where $J(x)$is the Jacobi matrix with entries

$$(J(x))_{ij} = \frac{\partial F_i}{\partial x_j}(x).$$

With the the notation $c_k := x_{k+1} - x_k$, which is the correction, we can write the k-th iteration of the Newton method as

$$J_k c_k = -d_k.$$

The computation of the Jacobi matrix and the defect depends on the used time discretization scheme. We compute the approximation of the Jacobi matrix J analytically. In our experiments we use an approximation for the Jacobi matrix, where we neglect some of the terms, for details we refer to Chapter 11. The matrix consists of two parts

$$J_n = s_m J^M_{n+1} + s_a J^A_{n+1}$$

where J^M is a mass matrix and J^A the stiffness matrix contribution. The parameters s_m and s_a are determined by the time discretization scheme and can be calculated using the Butcher arrays in Chapter 8. The defect d is recomputed in every nonlinear iteration

$$d_{n+1} = N(x_{n+1}) - b.$$

For a more robust convergence behavior we sometimes use a damping strategy, where the next iterate is computed as

$$x_{k+1} = x_k + \lambda c_k,$$

with a parameter $\lambda \in (0,1]$, that can either be given as fixed valued or can be computed with a line search strategy. We use a simple strategy that uses the error reduction rate, after the solution of a time step. We compare the new defect $d = d(\lambda)$ with the old defect d^{old}

$$\frac{\|d\|}{\|d^{old}\|} \leq \varepsilon,\ \varepsilon \leq 1.$$

We can double λ up to λ_{max} until either the ε criteria or a chosen λ_{max} is reached.

12.2 Splitting methods

We want to solve the equation

$$Ax = b$$

where A is a given $n \times n$ non-singular matrix. Splitting methods, as the Jacobi method or the Gauss-Seidel method, use a splitting of the matrix A, i.e.,

$$A = M - N,$$

where M should be an approximation of A, that can easily be inverted. The iteration procedure is then given by

$$x_{n+1} = x_n + M^{-1}(b - Ax_n).$$

We get the Jacobi method when choosing M as the diagonal of A. For the Gauss-Seidel method A is decomposed into a lower triangular matrix L and a strictly upper triangular matrix U. The Successive Over-Relaxation (SOR) method can be written in the form $M = \frac{1}{\omega}D + L$ where ω is a relaxation factor.

Likewise, the incomplete LU decomposition can be derived using a decomposition of A. For sparse matrices we usually get a fill-in in L and U. The idea of the incomplete LU decomposition is to reduce the fill-in and put some of the entries of A into a matrix R, i.e,

$$A = LU - R.$$

The use of Jacobi or Gauss-Seidel method usually leads to poor convergence rates. However, the block version of Jacobi or Gauss-Seidel methods are popular and robust preconditioning techniques [HS91]. A different class of iterative methods are Krylov subspace methods that we introduce in the following.

12.3 Krylov subspace correction methods

Krylov subspace methods are well established iterative methods for linear systems with large sparse matrices. In a Krylov subspace correction methods one constructs an m dimensional Krylov space

$$K_m = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\},$$

for a given matrix A and a residuum $r_0 = b - Ax_0$. The original problem $Ax = b$ is projected to the lower dimensional Krylov subspaces and we get a series of lower dimensional problems. An iterative process minimizes the defect $d_m = b - Ax_m$ in a certain norm in K_m. An introduction and detailed description of Krylov methods can be found in [BBC$^+$94]. The main design feature and difference between Krylov methods is the choice of the minimizer as well as the number of iterations carried out. Choosing the energy norm one obtains the Conjugate-Gradient method. For a symmetric positive definite matrix A we can construct a functional

$$F(x) = \frac{1}{2}x^T Ax - b^T x$$

with

$$\nabla F(x) = Ax - b = -r.$$

The minimizer of the functional is equivalent to the solution of the original problem $Ax = b$. The minimizer is constructed using the steepest descent direction under the condition that the direction is A-conjugate with respect to the old directions. We start with a first basis vector p_0 that is the negative of the gradient of the functional F at the initial guess x_0 ,i.e,

$$p_0 = b - Ax_0.$$

For the k-th step of this process we can compute a residual

$$r_m = b - Ax_m.$$

We want to remark that the directions of the basis vectors are constructed conjugate to each other. The next descent direction is built of the current residual and all previously computed search directions

$$p_m = r_m - \sum_{i=0}^{m-1} \alpha_j p_j$$

with coefficients

$$\alpha_j = -\frac{(Ar_m, p_j)}{(Ap_j, p_j)}$$

with $j = 0, \dots, m-1$ and p_m is constructed incrementally as

$$p_m = r_m - \frac{(Ar_m, p_{m-1})}{(Ap_{m-1}, p_{m-1})} p_{m-1}.$$

The convergence of the CG method is assured only for a symmetric positive definite matrix A. The Bi-Conjugate-Gradient (BiCG) method is also suitable for non symmetric matrices. In the BiCG method we build two sequences of orthogonal vectors

$$r_{m+1} = r_m - \alpha_m A^T p_m,$$

$$\tilde{r}_{m+1} = \tilde{r}_m - \alpha_m A^T \tilde{p}_m$$

with

$$\alpha_m = \frac{(r_m, \tilde{r}_m)}{(Ap_m, \tilde{p}_m)}$$

The corresponding search directions are computed by

$$p_{m+1} = r_m - \beta_m p_m,$$

$$\tilde{p}_{m+1} = \tilde{r}_m - \beta_m \tilde{p}_m$$

with

$$\beta_m = \frac{(r_{m+1}, \tilde{r}_{m+1})}{(Ap_m, \tilde{p}_m)}.$$

For all residuals and the search directions the following biorthogonality holds for all $i \neq j$

$$(\tilde{r}_i, r_j) = 0$$

and

$$(\tilde{p}_i, Ap_j) = 0.$$

The BiCG method has the disadvantage that no minimizer is computed. In the solution process this results in unbalanced convergence behavior. In the computation of the two series of residuals a multiplication with the transposed matrix of A is required. In the Bi-Conjugate-Gradient Stabilized (BiCGStab) method the residual vector is computed using a polynomial P of A

$$r_m = P_m(A)r_0$$

and

$$\tilde{r}_m = Q_m(A)P(A)r_0.$$

Here Q_m is a polynomial of the form

$$Q_m(A) = (I - \omega_1 A)(I - \omega_2 A)\dots(I - \omega_m A).$$

The constants ω_i minimise $\tilde{r}_m = (I - \omega_m A)s_m$ with $s_m = \tilde{r}_{m-1} - \alpha_i A\tilde{p}_i$. We use a preconditioned version of the BiCGStab method, for details we refer to [BBC$^+$94].

12.4 Multilevel methods

Krylov methods tend to have an irregular and slow convergence behavior depending on the condition and the eigenvalues of A. We can get an considerable improved convergence behavior using a good preconditioning strategy [Näg04]. We use a left preconditioner M for the system $Ax = b$ and get

$$M^{-1}Ax = M^{-1}b.$$

This is an appropriate strategy if the Krylov method has a better convergence behavior for the matrix $M^{-1}A$. However, we have to solve an additional system

$$My = z$$

within each iteration. Preconditioning can be combined with iterative methods. We use a special class of preconditioning techniques based on multilevel techniques, a geometric multigrid and a restricted additive Schwarz method. In a multilevel method we use iterative solver on meshes with different mesh sizes. Standard iterative solver have the ability to reduce high frequent error components very fast. In the context of multigrid methods these solvers are called smoother. The basic idea of a multigrid is to use the smoothing property of iterative solvers on a fine level and then restrict the smoothed defect to a coarser level, i.e a coarser mesh, and repeat the procedure. For details on multigrid methods we refer to the books of Hackbusch [Hac85] and Wesseling [Wes04].

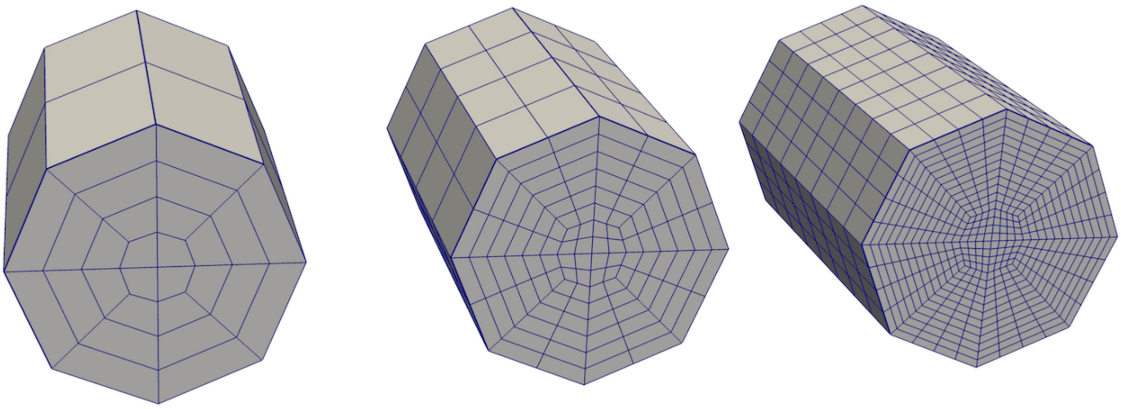


Figure 12.1. *The uniform refinement of a curved surface.*

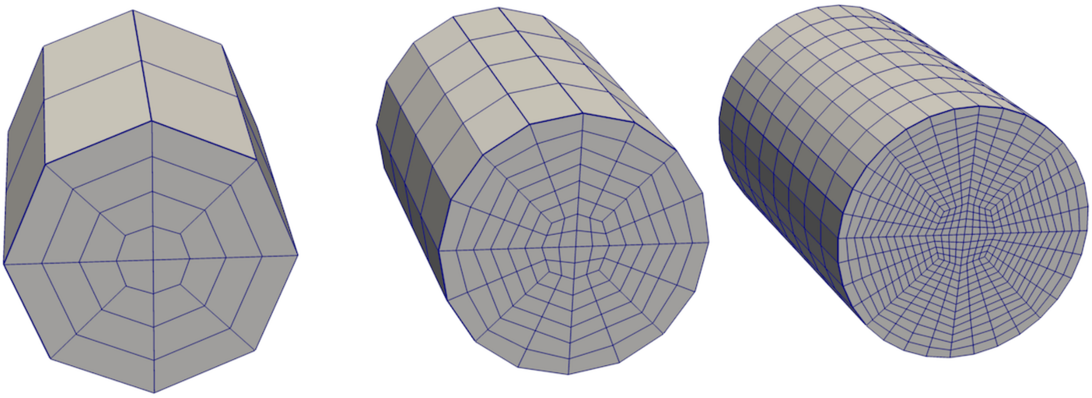


Figure 12.2. *The uniform refinement of a curved surface. We use an adaptation of the generated nodes to get a better representation of the curved surface.*

The basic idea of the additive Schwarz method is to split the domain into smaller sub-domains. We partition the domain into overlapping or non-overlapping sub-domains. On these subdomains we can solve the problem independently. However, we have to do this in an iterative way since we have to exchange the solution at the boundaries between neighboring sub-domains. To improve the additive Schwarz method we use a two level concept. The second level is used to enable fast exchange of information between non-neighboring sub-domains and therefore to speed up the convergence of the whole method. In the next sections we will introduce the main tools for the geometric multigrid and the two-level additive Schwarz method. Finally we introduce the methods themselves.

The hierarchy for a multilevel method is created by uniform refinement of a coarse mesh. As an example a sequence of two refinements can be found in Figure 12.1 However, for many cases standard nested mesh hierarchies lead to a very poor description of the boundary and interfaces of the computational domain. To avoid a large number of mesh nodes to resolve curved surfaces on the coarsest level, we use an adaptation strategy where we adjust the boundary and interface nodes on the refined meshes. The mesh sequence corresponding to the example in Figure 12.1 can be found in Figure 12.2. In this process we generate new positions for the corresponding nodes using a high resolved description of the computational domain. On each level l we can write the algebraic problem as

$$A_l x_l = b_l,\ l = 0, ..., L.$$

For the transfer of information between the different levels, we use restriction and prolongation operators that are introduced in the following.

12.5 Grid transfer

We introduce a prolongation and a restriction to transfer information between different levels of the multilevel hierarchy.

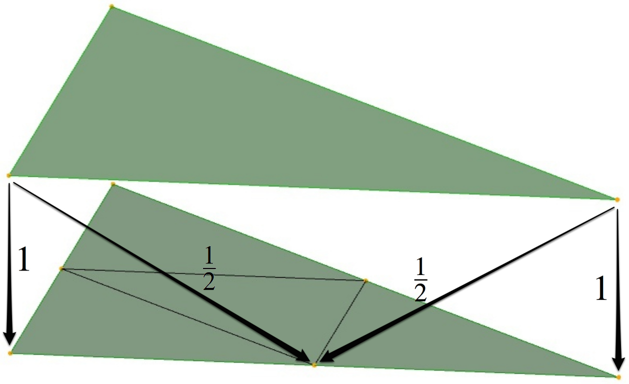


Figure 12.3. *We use a uniform refinement strategy to build mesh hierarchy. Here we show the weights for the interpolation between a parent and child elements.*

We define a map from the coarse level $l-1$ to the finer level l. This interpolation operator is also called prolongation: $P_l : V_{l-1} \rightarrow V_l$ and a simple way to construct a prolongation operator is to use the already introduced basis functions. An example for a triangle is shown in Figure 12.3. The operator from the fine level to the coarse level is called restriction and is given by the adjoint

$$R_l = (P_l)^T.$$

We use these prolongation and restriction operators to transfer nodal values from a coarse mesh to a finer mesh and vice versa.

12.6 Multigrid method

We have already introduced the major tool for the geometric multigrid method. In the following we use

- Mesh hierarchy,
- Transfer operators,
- Coarse problem solver,
- Iterative solver, that we also call smoother in the context of multigrid methods,

to set up the geometric multigrid.

The basic idea of a multigrid method is to use the smoothing ability of iterative solvers to reduce the high frequency error in the solution on each level. The iterative method is therefore only applied for a few iterations on one level. The residual is then restricted to a coarser level. Here again some iterations of the smoother are carried out. This procedure is usually repeated until the mesh is coarse enough to use a direct solver. The coarse solution is interpolated to the finer level. Whenever necessary post smoothing steps, again with an iterative method, are performed.

In Algorithm 2 we describe a multigrid method. Here the parameters k and l specify the number of pre and post smoothing steps. For $\gamma = 1$ we get a so call V-cycle, while for $\gamma = 2$ we get a so called W-cycle. For the differences between a V and a W cycle we refer to Figure 12.4.

12.7 Additive Schwarz method

The additive Schwarz method is a domain decomposition [SBG04, QV99, SBG96, Woh01, Xu92, XZ98] strategy. Domain decomposition in general is a preconditioning technique which is very well suited for parallel implementation. The basic idea is to split the solution domain into smaller sub-domains. Formally we first partition the domain Ω into N overlapping or non-overlapping sub-domains $\Omega_i, i = 1,...,N$. An often used method in this class of domain decomposition techniques

Algorithm 2 Multigrid algorithm, for $\gamma = 1$ we get a V-cycle and for $\gamma = 2$ we get a W-cycle

```
if level = coarsest then
    Compute coarse grid correction
else
    Apply $k$ pre-smoothing steps
    for $i = 1, \dots, \gamma$ do
        Compute residual
        Restrict residual
        Call this algorithm on the coarser level to solve the residual problem on the coarser grid.
        Correct the fine solution
    end for
    Apply $l$ post-smoothing steps
end if
```

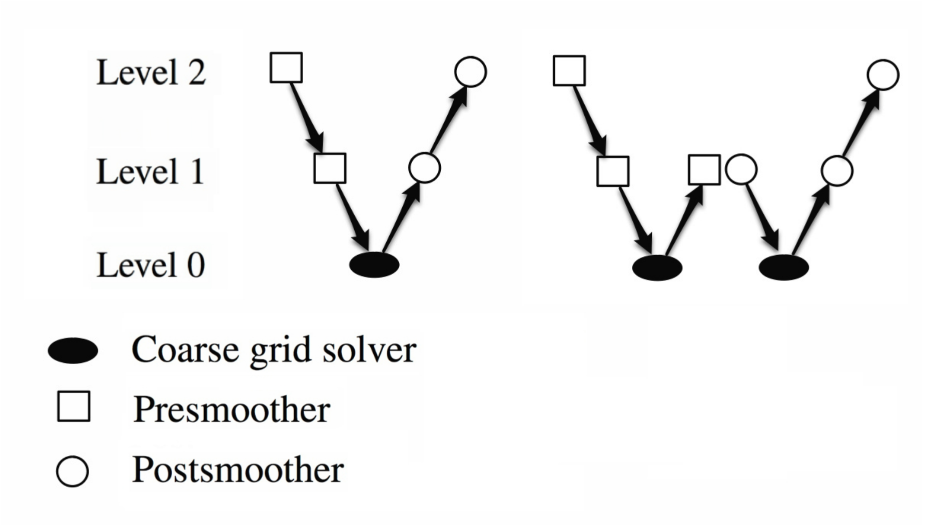


Figure 12.4. *Different multigrid cycles. Left: V-cycle. Right: W-cycle.*

are additive Schwarz methods [Lio78, Cai91, DSW93, FS98, SBBN00, NS02, TW10]. We write the additive Schwarz method as a Richardson method

$$x^{k+1} = x^k + (\sum_{i=1}^{N} Q_i)(b - Ax^k)$$

with a preconditioner

$$M = (\sum_{i=1}^{M} Q_i) = (\sum_{i=1}^{N} R_i^T A_i^{-1} R_i).$$

Here, on each sub-domain Ω_i an operator A_i is constructed, which is the restriction of the matrix A. Additive Schwarz methods are well analyzed for elliptic problems. For other problems theoretical results are not very well developed.

To improve the additive Schwarz method, we use a two-level concept. The basic idea of the two-level approach is to enable fast exchange of information between non-neighboring sub-domains. For example, if there are k sub-domains between sub-domain Ω_i and sub-domain Ω_j, in the one-level Schwarz method it can take up to k iterations to propagate information from sub-domain Ω_i to sub-domain Ω_j. In a two-level approach a coarse grid is constructed on which the solution process is less expensive but on which the global structure of the solution can still be represented well. The restricted additive two-level approach is then given by

$$M_{\text{ad}} = (R_0)^T A_0^{-1} R_0 + \sum_{j=1}^{N} (R_j^0)^T A_j^{-1} R_j$$

where R_0 is the restriction from the fine grid to the coarse grid and R_j^0 is a restriction that does not include overlap. Our implementation of the two-level restricted additive Schwarz methods consists of several components. According to the first term we solve a coarse grid problem with the methods mentioned above for the coarse grid problem. Then we interpolate the solution using the prolongation operators, that we also use for the multigrid. Here, we also use the mesh hierarchy transferring the solution of the coarse mesh level by level to the finest mesh. We use the restricted one-level additive Schwarz method to solve the fine problem. Our restricted additive Schwarz solver is based on PETSc [BGMS97, BAA$^+$14a, BAA$^+$14b].

12.8 Assembling

We can distinguish between two different ways the matrices on the different level can be computed. A usual strategy to construct a multilevel hierarchy is to use a Galerkin process. Here one uses transfer operators P_l and R_l to compute the stiffness matrices on the other levels. When using standard operators for P and R the upwind schemes behave like central differences upwind schemes on coarser levels. We mentioned already in the Chapter 9 that, especially for convection dominant flow problems, central difference scheme have several disadvantages. The second way is a direct assembly of the matrices on every level in the level hierarchy. The assembly of the stiffness and mass matrix scales quite well, since all elements are uniquely distributed to different processors and the assembling can be done without communication. For details we refer to Chapter 14. Therefore we can efficiently use direct assembling on different levels instead of using a Galerkin approach.

12.9 Ordering of unknowns

The system that has to be solved consists of different variables, velocities, pressure and displacements for the fluid, displacements and velocities for the structure. The order of these equations has a significant impact on the solving process. In general we can distinguish at least between two concepts:

- ordering the Jacobi matrix by sub problem as shown in Figure 11.1,
- ordering by nodes, which results in a vector of blocks with size 10. Corresponding to that the system matrix has a block structure of size 10 as well.

In the ordering by sub problems we would get a matrix with rather big blocks. The second ordering (by node) results in smaller blocks which can be handled more efficiently [Bar09, Näg04, Met04].

12.10 Coupling implementation

In order to keep the modularity of the subproblems, the assembler of the fully coupled system is implemented as a wrapper. Therein the different element assembler for the specific fields are called. For the fluid a Stokes and a Navier Stokes assembler are currently implemented. On the structure side we have implemented a Finite Element assembler for linear elastic problems. The coupling consists of three assembler

- one for the velocity coupling,
- the assembler for the force coupling,
- and the assembler for the geometry coupling.

For the geometry coupling various options have been implemented, such as the already mentioned implicit and explicit method, but also geometric matching and non-matching coupling at the interface is implemented. However, this was not used for the examples presented in this thesis.

The implementation of assembling is not optimized for reducing the total assembling time, but it provides additional flexibility for changing the sub-assembler routines for fluid, structure and coupling conditions. Therefore the underlying models for fluid can be changed without touching the other routines. For the scaling behavior of the assembler we refer to Chapter 14.

12.11 Fluid mesh smoothing

We have already discussed the geometric explicit coupling of the fluid structure interaction problem. In this scheme we solve two problems. First the fully implicit coupled fluid structure interaction problem is solved where we enforce the nodes of the mesh in the inner of the fluid domain to stay on their geometric positions. In a second step, we apply a smoothing procedure to the fluid mesh. Since our solver already uses a multilevel hierarchy, we can also use the nested mesh hierarchy for an efficient computation of a smooth distribution of the fluid mesh. Depending on the underlying geometry the smoothing of the mesh can often be carried out on the coarsest mesh. Since the distribution of the mesh can be chosen arbitrarily in the interior we can simply interpolate the solution to the finer levels. For more demanding geometries an intermediate level for the smoothing process can also be chosen. In this case the smoothed mesh displacements are restricted and interpolated to other levels of the hierarchy. Usually it is not necessary to use the finest level for the mesh smoothing process.

13 Applications

In this chapter we present two different applications in fluid structure interaction and in which the developed code was already used. We start with an example from blood flow simulation. The second example is about flow sensor optimization.

13.1 Blood flow simulation

In the last years computational fluid structure interaction has been seen more and more in the field of blood flow simulation. This includes the simulation of implants and blood pumps, flows through different type of vessels and many more. The typical Reynolds number for flow in blood vessels varies from a few hundred to a couple of thousand directly at the heart. However, the flow is usually laminar. The vascular wall is viscoelastic and may be exposed to large deformations. In most simulations simplified models for the structure are used. For simple blood vessel simulations often also a linear elastic model is assumed. We are aware that much more sophisticated models for blood flow simulation exist. On the fluid side the use of a Finite Volume methods in general have been shown to be advantageous and reliable [MAFB^{+}10].

One distinguishes different kinds of blood vessels: arteries, the aorta, arterioles, venules, capillaries and veins. The diameter varies from up to three centimeter to several micrometer. In very small vessels, e.g, venules, the flow of blood can neither be modeled using a Newtonian fluid nor can the Navier Stokes equations be used as a reasonable model. Blood is composed mainly of blood cells and blood plasma. Blood plasma consists mostly of water. Blood cells can be subdivided into three categories, red blood cells (erythrocytes), white blood cells (leukocytes) and platelets (thrombocytes). They have an elastic behavior and undergo large deformation when passing through capillaries and venules. In these small vessels the behavior of blood is non Newtonian since the relation between the shear stress and the shear rate is not linear. However, for larger vessels, i.e larger than

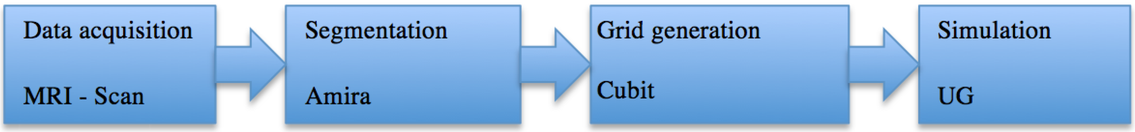


Figure 13.1. *Workflow set up for the simulation with patient specific geometries.*

0.1 cm, blood can be assumed as a Newtonian fluid in most cases. The fluid can thus be simulated employing the incompressible Navier Stokes equations. Within this thesis we have set up a full work flow using medical data. The intermediate steps of the work flow are shown in Figure 13.1. Medical data coming from MRI or CT scan are usually provided in DICOM format. In the following segmentation process, to each voxel a material is assigned. In our workflow AMIRA [SWH05] is used for the segmentation. In Amira the segmentation can be done manually, automatically or by means of semi-automatic tools. The final result of the segmentation procedure is a surface description of the domain. Usually we use a triangulation for the surface representation. The surface representation, since it bases on a voxel representation, requires additional smoothing. In this step of the workflow a high resolution model of the surface is generated. This high quality representation is later used to adapt the boundary and interface nodes of the refined meshes in the multilevel solver method to get a better geometry representation within the mesh hierarchy. Finally we use CUBIT [Tea] for the mesh generation process. Usually a comparatively coarse mesh is created that is refined to build up a mesh hierarchy. As already mentioned we modify the mesh positions on the finer level to get a better geometry representation.

For a sample simulation we have used real patient data from the DICOM image sets of OsiriX [RSR04]. Figure 13.2 shows an aneurysm of the thoracic aorta. The simulation of aneurysm and haemodynamics in general is quite demanding and topic of current research [BHZ$^+$10, dL09, CDFQ11, CRD$^+$11]. In this work we realized the full work flow for realistic biomechanical simulations. The validation of the blood flow simulations is not part of this work.

13.2 A bionic flow sensor for hydrodynamic metering

In collaboration with the center of advanced european studies and research (caesar) in Bonn, Germany, our implementation was used for the simulation of a sensor for evaluation of the flow rate of fluid in a micro canal. Within that project the code was used for optimizing the shape of a prototype sensor.

As a biological model, the electrosensory and lateral line mechanosensory systems of fish is used. The main aim at the Institute of Computational Science in Lugano was the optimization of the shape of a capacitive sensor under certain restrictions. The sensor, which is embedded in a flow canal is deformed by the fluid impact. The general design of the sensor and the canal can be seen in Figure 13.3. The sensor consists of a brick or cylinder shaped bar, which is the main obstacle in the fluid flow. This bar is located on top of a capacitor. The capacitor is segmented in different parts where a positive or negative change in the capacity of a specific area of the capacitor can be measured. For the optimization of the sensor different shapes of the canal, the bar and of the capacitor were tested. The force of the fluid acting on the bar and the capacitor result in a change of the capacity of the sensor. After some delay the next sensor is deformed. Using the delay in the signal peak of the different sensors, one can compute the fluid velocity. In Figure 13.4 two extremes of a simplified version of a model sensor are shown. Here one can easily see that the different shape of the bar has a crucial impact on the deformation of the capacitor.

The capacitance C_0 of a capacitive displacement sensor with two parallel conductive plates, with

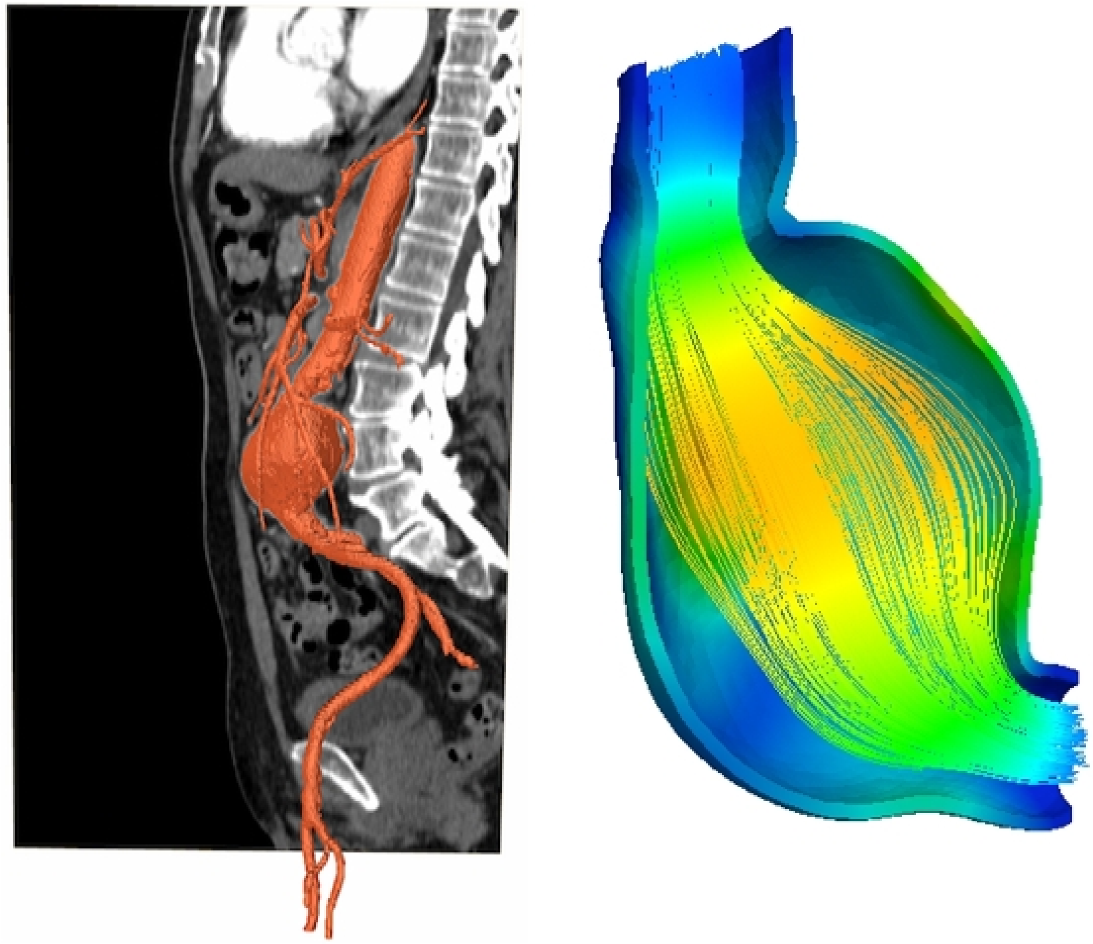


Figure 13.2. *Left: MRI image and segmented geometry. Right: Streamlines of the flow in an aneurysm. The color of the stream lines indicates the pressure distribution. The color of the structure shows Mises stresses.*

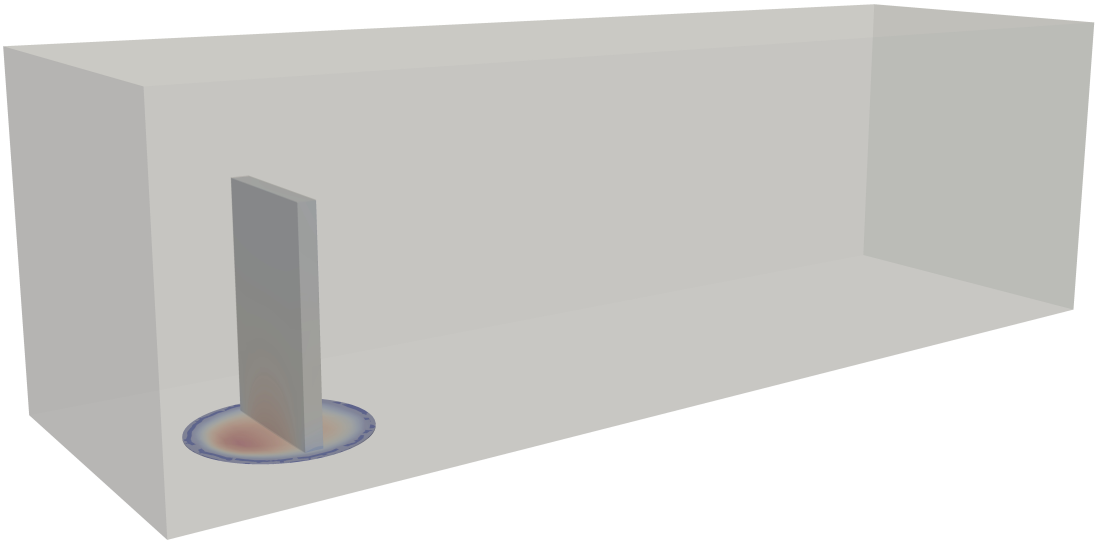


Figure 13.3. *Canal with sensor.*

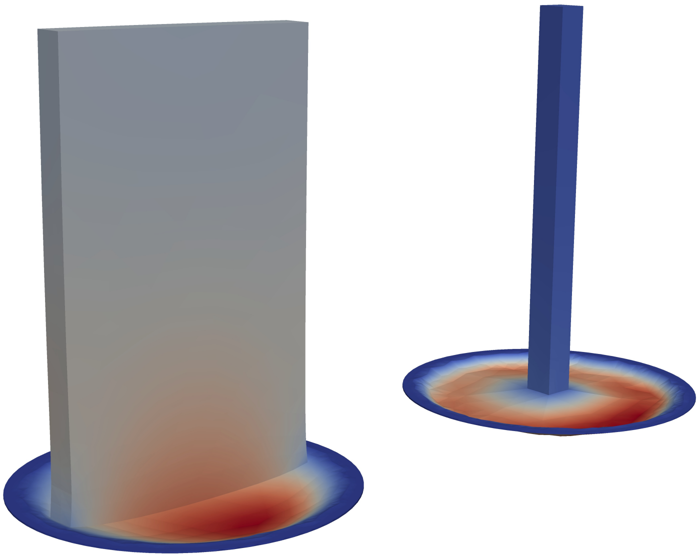


Figure 13.4. *Two simplified version of a sensor with warped deformations.*

a separation of size *d* between, is given by the flowing relation

$$C_0 = \frac{\varepsilon_0 K A}{d}.$$

Here ε_0 is the permittivity of free space constant, *K* is the dielectric constant of the material between and *A* is the area of the parallel conductive plates. The computation of the capacity change of the capacitor can be done using the deformation of the capacitor plates [Elg99]. In the computation of the capacitance we ignore any contribution of fringing fields since the deformation of the gap of the two parallel conductive plates are small compared to the area of the plates. The upper conductive plate is deformed by the fluid flow, while the lower plate remains undeformed. The change of capacitance ΔC between the deformed plate and the undeformed plate can be calculated as

$$\Delta C = C - C_0 = \int_A \frac{\varepsilon_0 K}{d - w(x,y)} dA - \frac{\varepsilon_0 K A}{d}$$

and w(x,y) is the deformed plate deflection.

14 Numerical results

In this chapter we compare different preconditioning strategies for the coupled fluid structure interaction problem using the proposed geometric implicit and geometric explicit discretizations. In the first section we describe the test example used in this chapter, then we give a comparison of the scaling behavior of the solver.

14.1 Setting

We use a cuboid flow canal with an inserted brick. The canal is 4 meter long and has a width and height of 2 meter. The brick is inserted 1.5 meter behind the inflow surface. It is 0.5 meter long and has a height and a width of 1 meter. The setting is show in Figure 14.1. The inflow has a mean velocity of 6 $\frac{m}{s}$. We use a backward Euler with a constant time step size of 0.001 seconds and a viscosity of $\nu = 0.01\frac{m^2}{s}$. At the outflow boundary we set a zero pressure condition.

This example was chosen for two reasons:

- The number of Newton steps does not vary much from time step to time step, and
- the generation of a series of hexahedral meshes were we double the number of mesh nodes is rather simple.

For the weak scaling tests we need a sequence of meshes to keep the ratio $\frac{\text{number degrees of freedom}}{\text{number processors}}$ constant. We start with a coarse mesh, that is refined in just one spacial direction, keeping the resolution in the other spacial directions constant. We repeat this procedure for the next mesh using another spacial direction, the generated sequence is shown in Figure 14.3. This strategy doubles the number of mesh nodes in the sequence of meshes. However two-thirds of the constructed meshes are anisotropic meshes. In this Chapter we compare two different preconditioners for the BiCGStab method. The absolute tolerance of the linear solver is set to 10^{-6} while the tolerance for the Newton method is 10^{-5}. The coarse grid solver is a Block Jacobi in combination with an incomplete LU decomposition were we use a tolerance of 10^{-6}. The pre and postsmoothing processes also uses the Block Jacobi in combination with an incomplete LU decomposition with a relative tolerance of 10^{-3} and a maximum of 30 smoothing steps. With this setting we usually perform 25 to 30 smoothing steps on each level. For the two-level restricted additive Schwarz method we also use a Block Jacobi in combination with an LU decomposition as coarse grid solver.

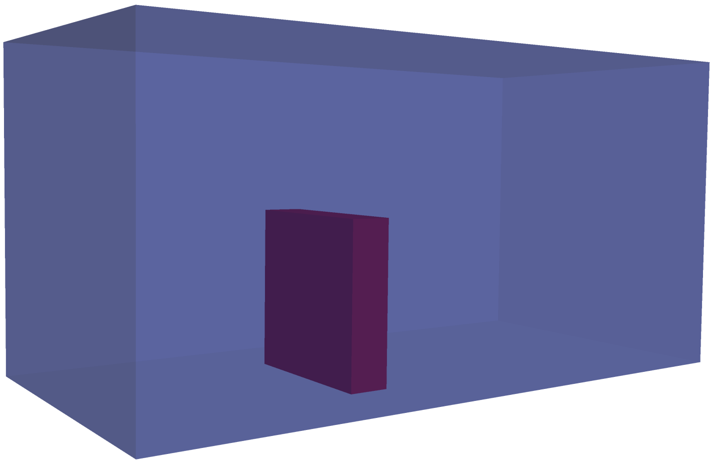


Figure 14.1. *Geometry of the scaling benchmark simulation.*

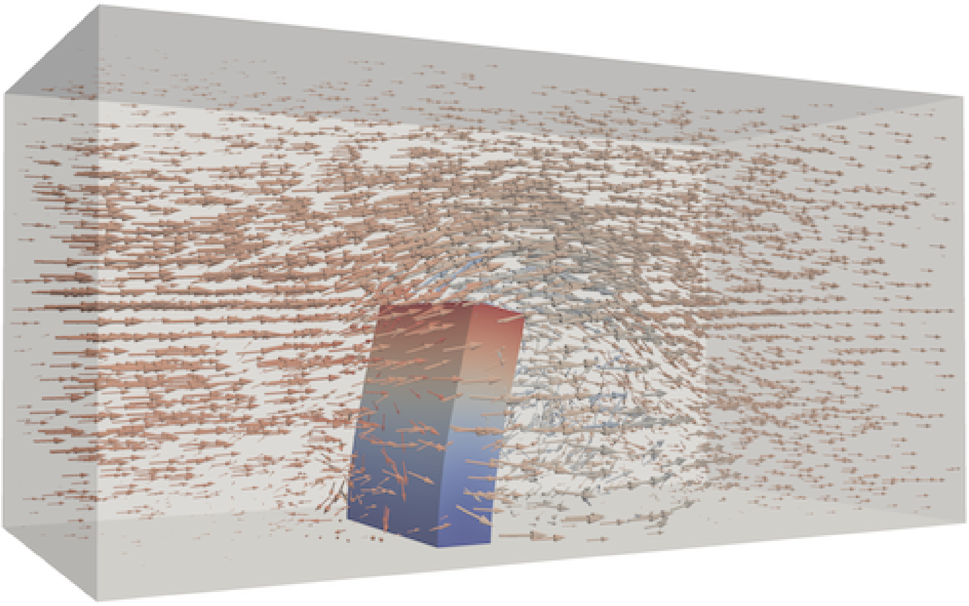


Figure 14.2. *Illustration of the flow in the scaling test case. The flow directions are given by arrows colored with the pressure distribution. We show warped structure displacements and color indicates the displacements distribution.*

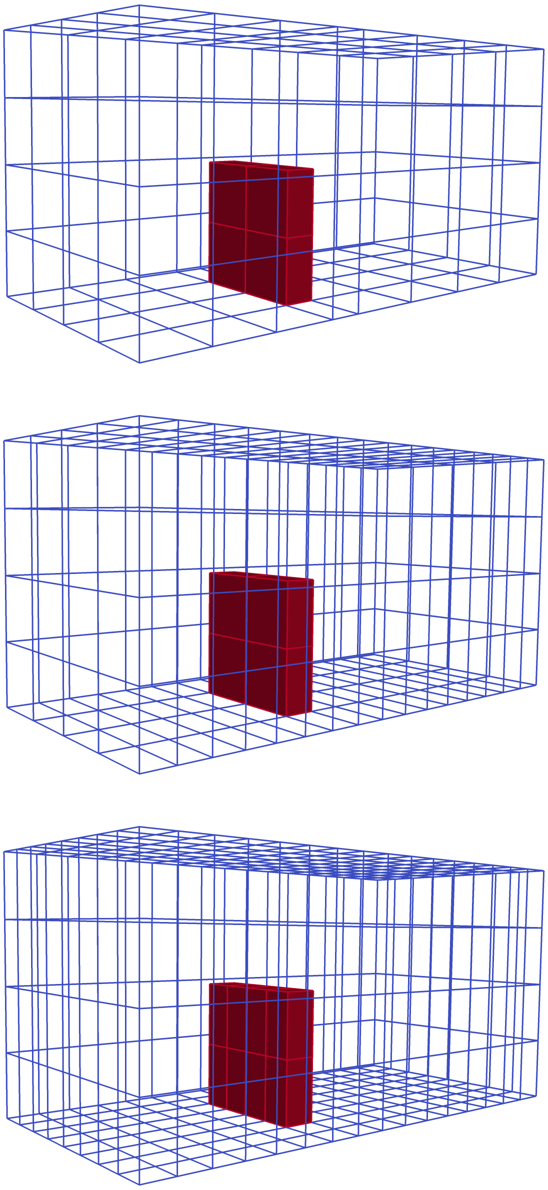


Figure 14.3. *Sequence of meshes: Non uniform refined just in one spacial direction.*

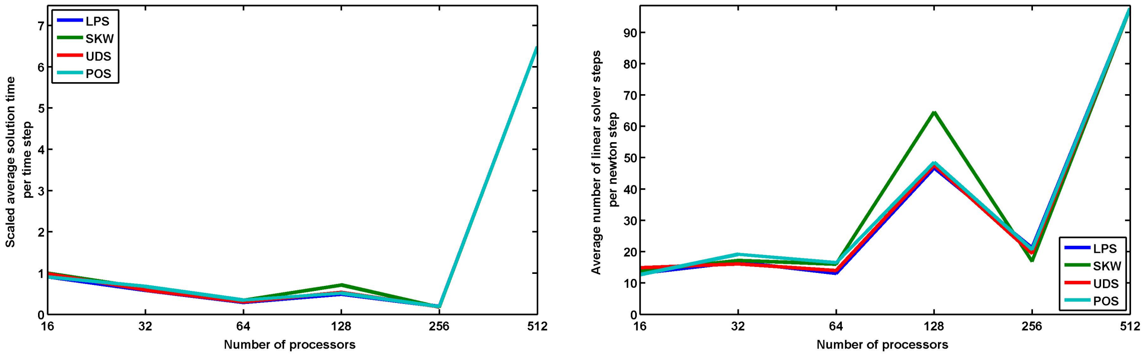


Figure 14.4. *Strong scaling for the multigrid solver in the geometric implicit case. Left: solution time. Right: Average number of linear iterations per Newton iteration.*

14.2 Scalability

In this section we will compare scalability results of different preconditioning strategies. For the computations we use the Cray XE6 "Monte Rosa" at the Swiss National Supercomputing Center with 2 × 16-core AMD Opteron Interlagos 2.1 GHz. For strong scaling we start with an amount of 16 nodes and use the lowermost mesh depicted in Figure 14.3 that fits in memory after two uniform refinements. On the finest level for the strong scaling test we get 364.650 degrees of freedom. We use a BiCG-Stab solver with different preconditioning strategies. In the following we compare the multigrid with the two-level restricted additive Schwarz method with an overlap of one for the geometric implicit and explicit handling of the fluid domain motion. For the multigrid we get good strong scaling up to 256 computing nodes. The solver times and number of linear iterations per Newton step can be found in Figure 14.4. For the geometric explicit discretization we see a very irregular convergence behavior of the multigrid, therefore we use the geometric explicit discretization just with the restricted additive Schwarz method. The strong scaling results for the two-level restricted additive Schwarz in the geometric implicit case can be found in Figure 14.5. Here we get almost perfect scaling up to 512 computing nodes. The behavior of the geometric explicit restricted additive Schwarz is shown in Figure 14.6. Compared to the geometric implicit version, more linear solver steps are needed, but the number stays rather constant in the range of 128 to 512 computing nodes and we get an considerable decrease, up to 50 percent, in the solver time. This is due to a decrease of the needed number of Newton steps.

For the explicit treatment of the fluid domain displacement degrees of freedom we observe a signifiant difference in the number of Newton steps needed to solve the system. The geometric explicit case typically reduces the number of Newton steps by up to sixty percent. For the stabilizations LPS, SKW and UDS we get a similar behavior as shown in Figure 14.8, were we have chosen the UDS stabilization to be representative for LPS and LPS as well. The POS stabilization shows a different behavior that is also shown in Figure 14.8. For this stabilization the reduction of Newton steps in the geometric explicit case is usually less than for the other stabilization. We also see this

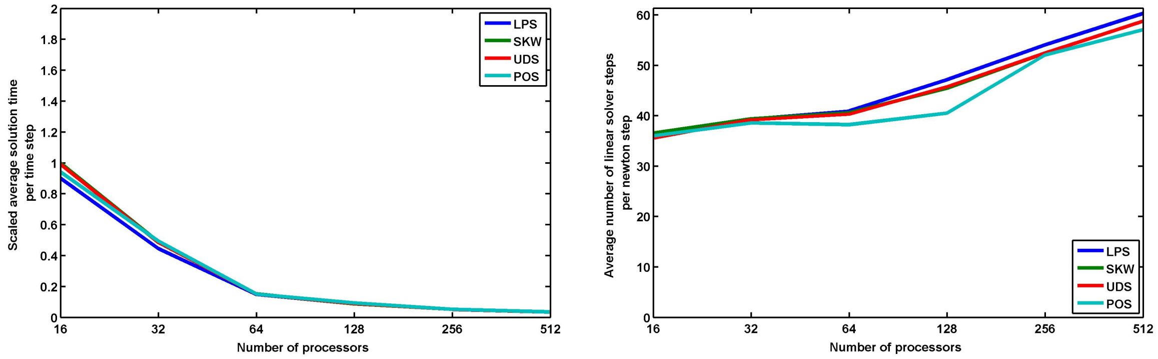


Figure 14.5. *Strong scaling for the additive Schwarz in the geometric implicit case. Left: solution time. Right: Average number of linear iterations per newton iteration.*

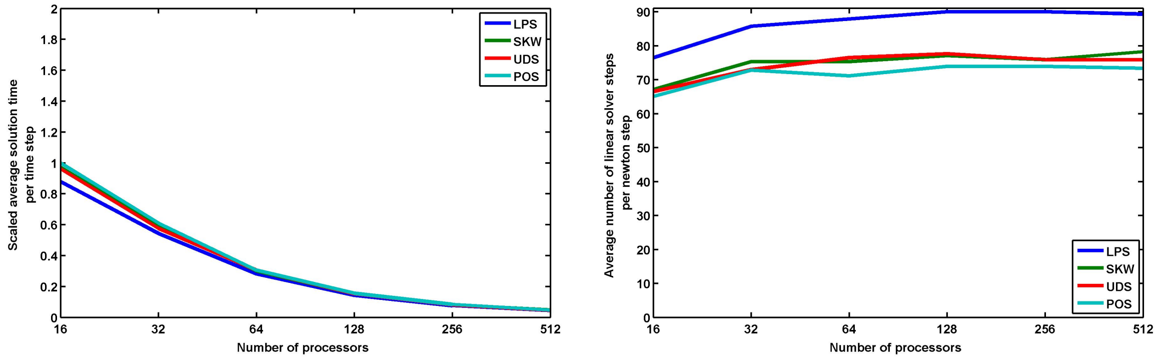


Figure 14.6. *Strong scaling for the additive Schwarz in the geometric explicit case. Left: solution time. Right: Average number of linear iterations per newton iteration.*

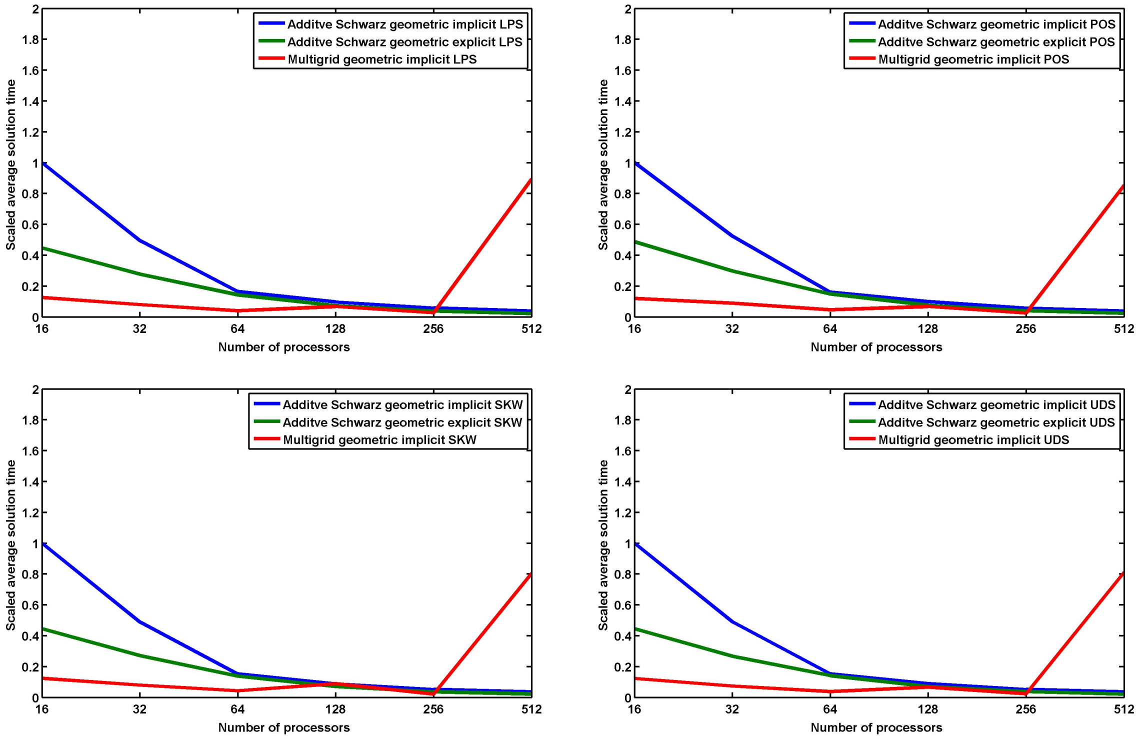


Figure 14.7. *Strong scaling results for the geometric implicit and explicit additive Schwarz method and the multigrid for different upwind schemes.*

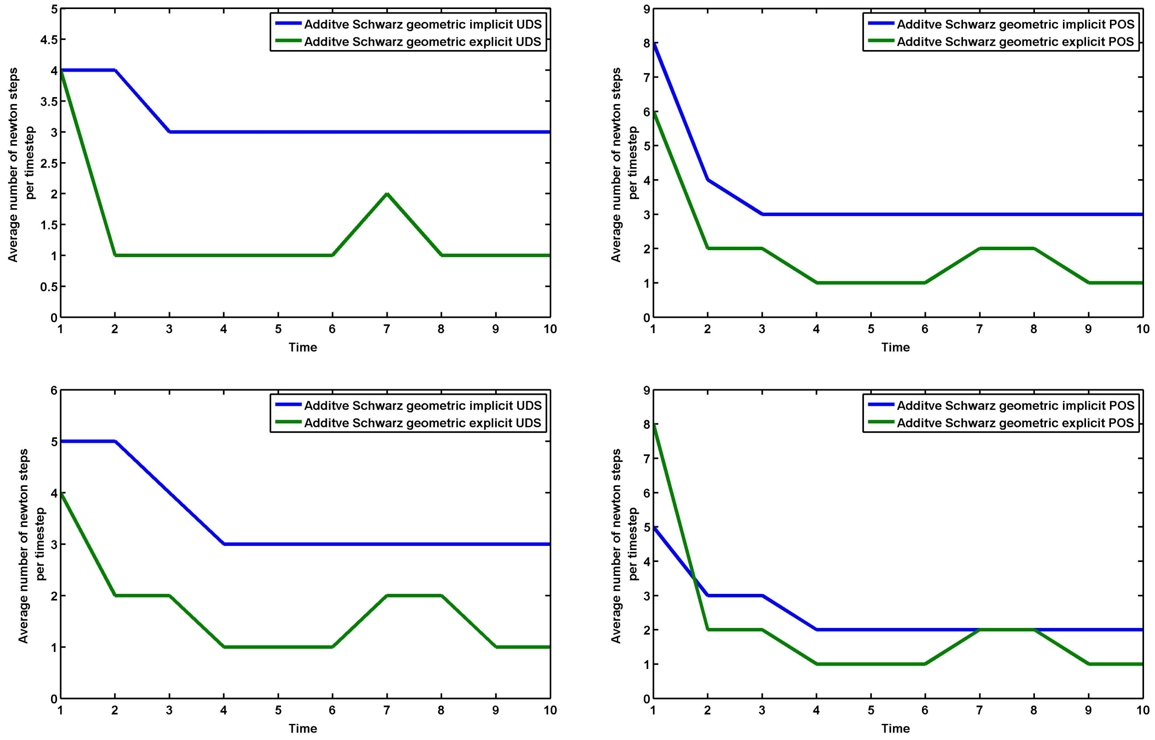


Figure 14.8. *We compare number of Newton steps per time step for the geometric implicit and explicit case for the UDS and the POS stabilization for two different coarse meshes. The coarse meshes are refined four times and we use the restricted additive Schwarz for solving the system. The first row shows the the number of Newton steps for the first mesh shown in Figure 14.3. The second row shows the the number of Newton steps for the second mesh shown in Figure 14.3. Both meshes are refined two times.*

behavior for more complex computational domains. In Figure 14.7 we compare the solution times for the proposed upwind schemes for all three different preconditioning methods. We see a similar behavior for all upwind schemes with some small differences in the solution times. Up to 256 computing nodes the multigrid is the most efficient preconditioning strategy. For 512 computing nodes we get a notable increase in the number of linear iterations. This mostly comes from the smoothing process within the multigrid method.

The matrix and defect assembly can be done mostly in parallel without any communication between processors. This is due to the fact that the load balancing is based on distributing elements to processors. The assembling of mass and stiffness matrix is carried out element by element. The computed element contributions are finally added to a global matrix, only this last step requires communication along processor interfaces to obtain a consistent matrix. The assembling is done in four steps: fluid, fluid geometry, structure, and the assembling of the coupling. This strategy is not optimized for maximal performance, but it provides flexibility for the replacement and extension

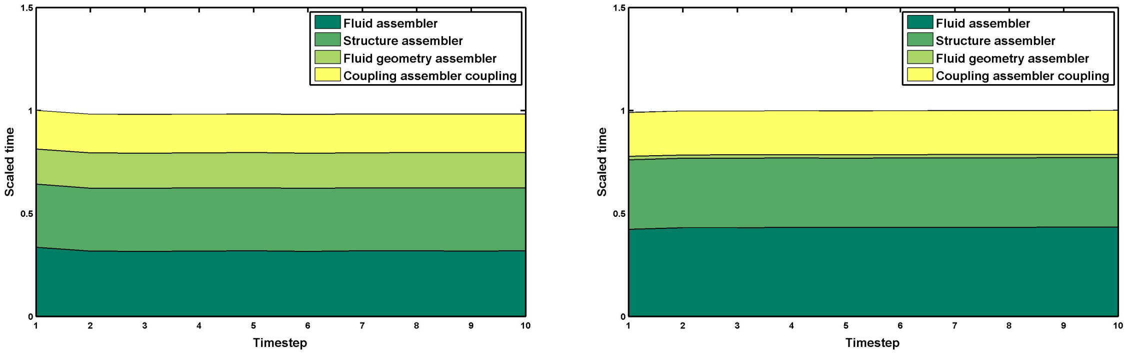


Figure 14.9. *Assembling times for the test example for the first ten time steps of the benchmark example. Left: Geometric implicit. Right: geometric explicit.*

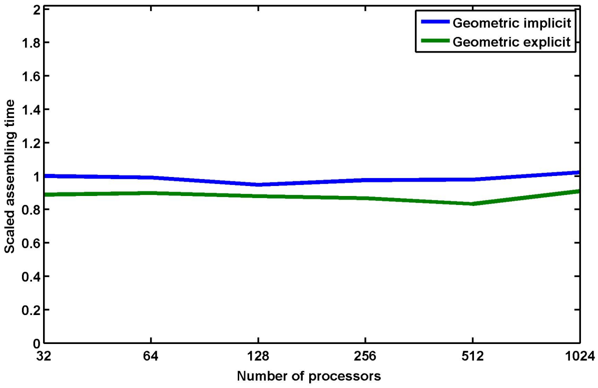


Figure 14.10. *Weak scaling of the assembler for geometric implicit and explicit.*

of the existing assembling routines. In Figure 14.9 we show a typical time distribution for the assembling process. The assembly phase in the geometric explicit case is faster than the geometric implicit case, because in the geometric explicit assembly we set simply a trivial condition to the displacements degrees of freedom for the fluid domain. The speedup clearly depends on the fluid structure domain ratio. In Figure 14.9 the weak scaling behavior of the assembler is shown.

For the weak scaling test we keep the ratio

$$\frac{\text{number degrees of freedom}}{\text{number processors}}$$

constant. As before we use the sequence of meshes shown in Figure 14.3. The resulting number of

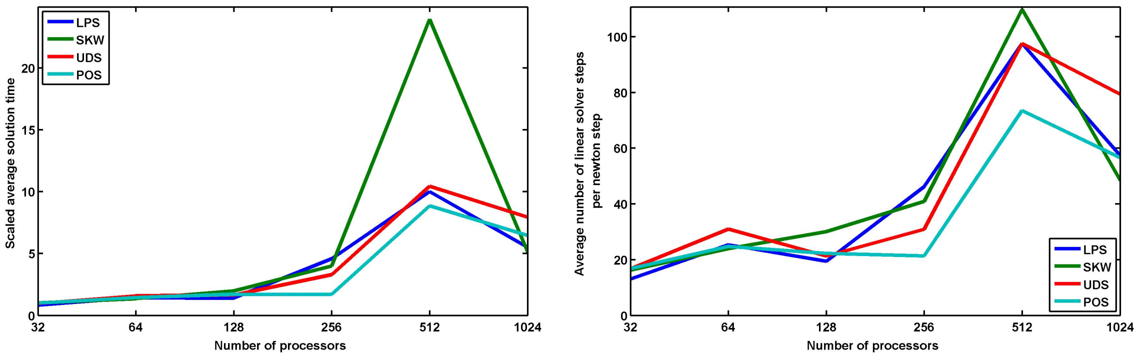


Figure 14.11. *Weak scaling for the multigrid in the geometric implicit case. Left: solution time. Right: Average number of linear iteration per newton iteration.*

degrees of freedom for the meshes is given the following table:

number of processors	degrees of freedom
32	187850
64	364650
128	707850
256	1404810
512	2767050
1024	5450250

For the multigrid we observe a large variation in the number of linear solver steps and in the solution time for different aspect ratios of the elements of the mesh, shown in Figure 14.11. A similar behavior of the multigrid is reported for the pure fluid discretization [Näg04]. For 64 and 512 computing nodes we get a (local) peak in the number of Newton iterations. These two meshes have the same aspect ratio for the elements of the mesh. For a small number of processors, up to 128, we get a good weak scaling behavior for the multigrid preconditioning. For the two-level restricted additive Schwarz method we observe in the geometric implicit case a more equal behavior in the number of linear solution steps for the LPS, SKW and UDS upwind methods. The POS method however shows an increase in solution time for 256 processors as can be seen in Figure 14.12. The two-level restricted additive Schwarz method in the geometric explicit case shows a similar behavior for the number of linear solution steps for all upwind methods. In comparison to the geometric explicit case we need about one third more linear solver steps per Newton iteration. For details we refer to Figure 14.12 and Figure 14.13.

For a comparison of the solution time of the multigrid and the two-level additive Schwarz method in the geometric implicit and explicit case we refer to Figure 14.14. Here we see that the multigrid is the fastest preconditioning method up to 128 or 256 processors depending on the upwind scheme. For more processors the additive Schwarz method is more robust than the multigrid. In the range of 256 to 1024 computing nodes the geometric explicit case is more efficient

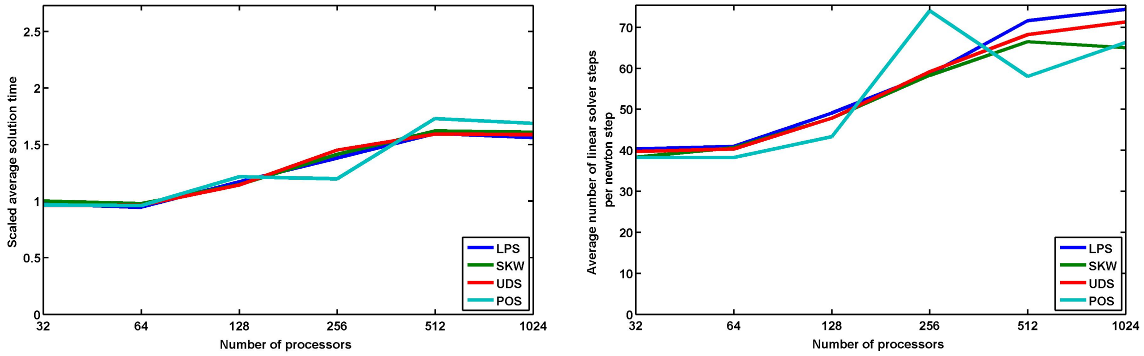


Figure 14.12. *Weak scaling for the additive Schwarz in the geometric implicit case. Left: solution time. Right: Average number of linear iteration per Newton iteration.*

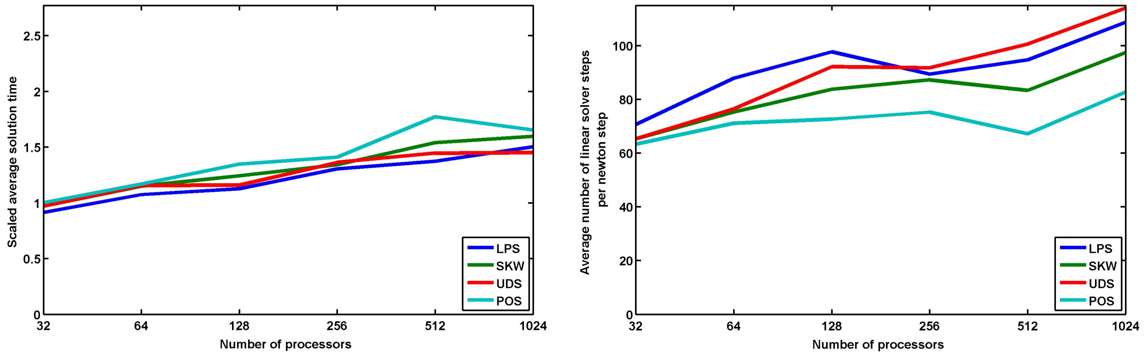


Figure 14.13. *Weak scaling for the additive Schwarz in the geometric explicit case. Left: solution time. Right: Average number of linear iteration per Newton iteration.*

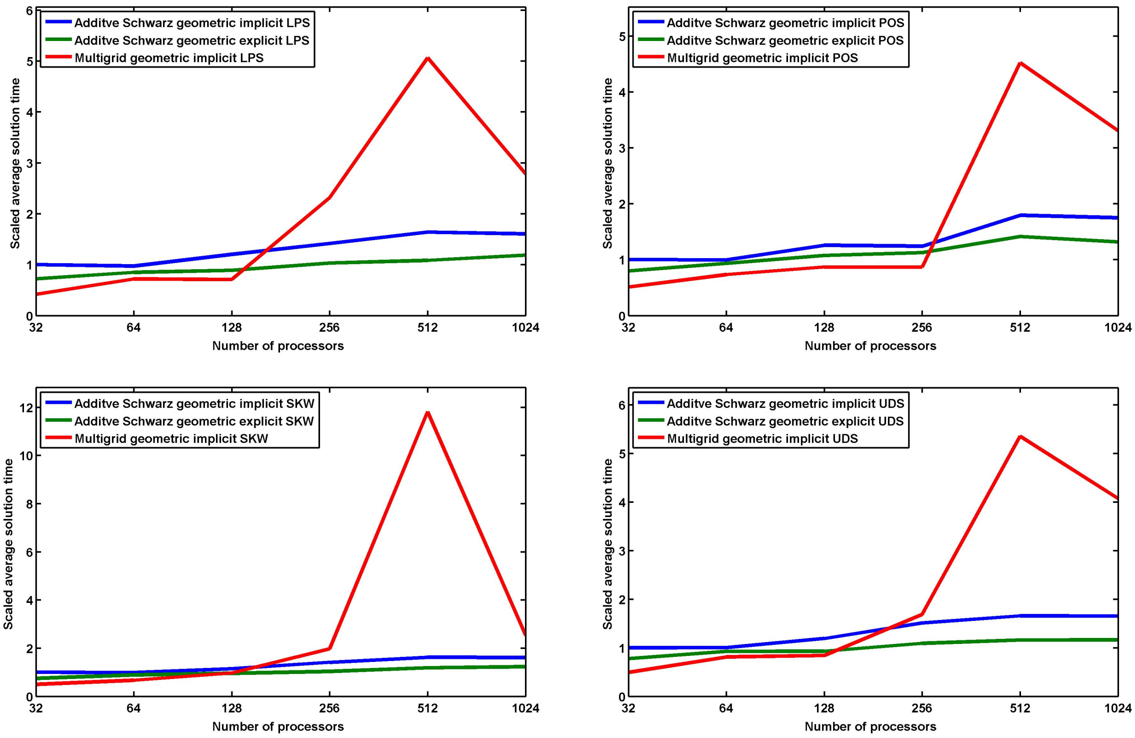


Figure 14.14. *Solution time for different upwinding schemes.*

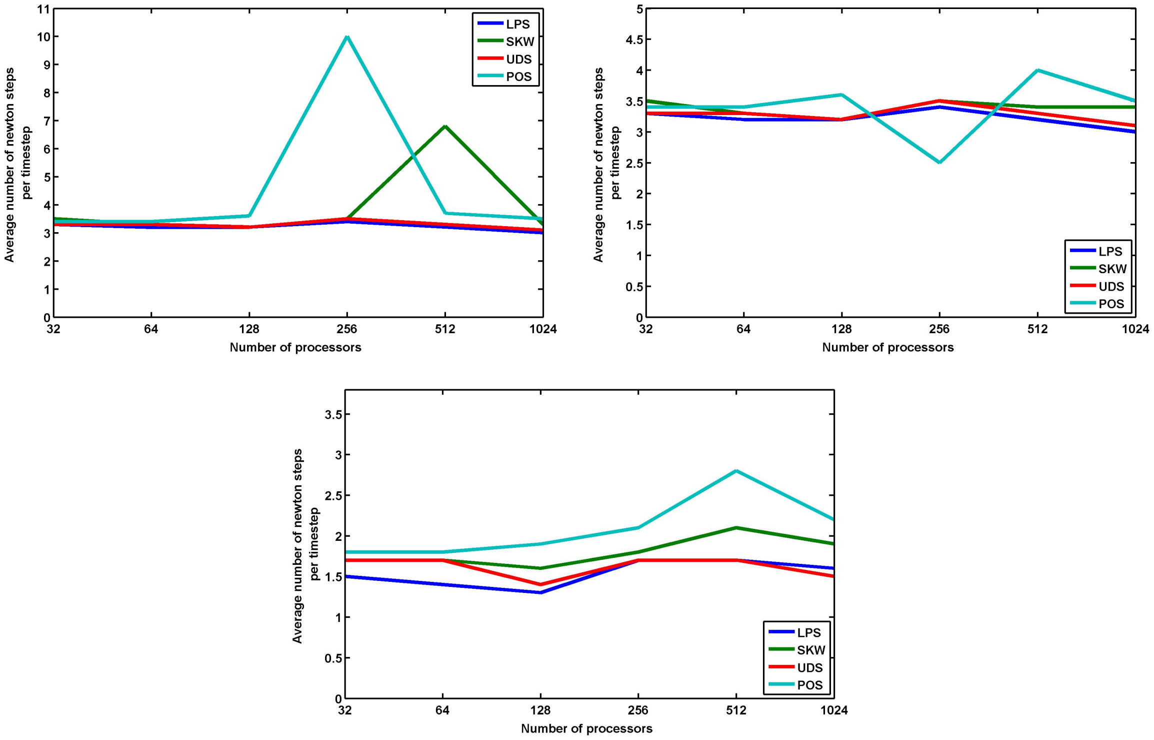


Figure 14.15. *We compare the average number of Newton steps for different upwinding schemes. In the first row we compare the geometric implicit case. Left: Multigrid, Right: Two-level restricted additive Schwarz. In the second row the number of Newton steps for the geometric explicit case for the two-level restricted additive Schwarz is shown.*

than the geometric implicit using the two-level additive Schwarz method. For the POS upwinding scheme we see again a more irregular behavior compared to the other upwinding schemes.

In the following we compare the two-level restricted additive Schwarz in the geometric implicit and explicit case for different overlaps for the Schwarz method. In this comparison we use the UPS, SKW and LPS upwind schemes. In Figure 14.12 we present the results for the geometric implicit case and an overlap of two. We see a decrease in the number of linear iterations and a smoother behavior in the number of Newton steps for the SKW upwind scheme compared to an overlap of one. In the geometric explicit case that is shown in Figure 14.17, we can see an almost perfect scaling up to 1024 processors. As in the geometric implicit case we see a decrease in the number of linear iterations and a very similar behavior in the number of Newton steps compared to an overlap of one. For an overlap of three this behavior continues. We compare the different upwind schemes for an overlap of one, two and three in Figure 14.16 and Figure 14.17. The number of linear iterations decrease, but with an increase of solution time. In Figure 14.20 we compare the solution time for different overlaps and upwind schemes. For all upwind schemes the geometric explicit two-level restricted Schwarz method with an overlap of 1 is the fastest preconditioning strategy.

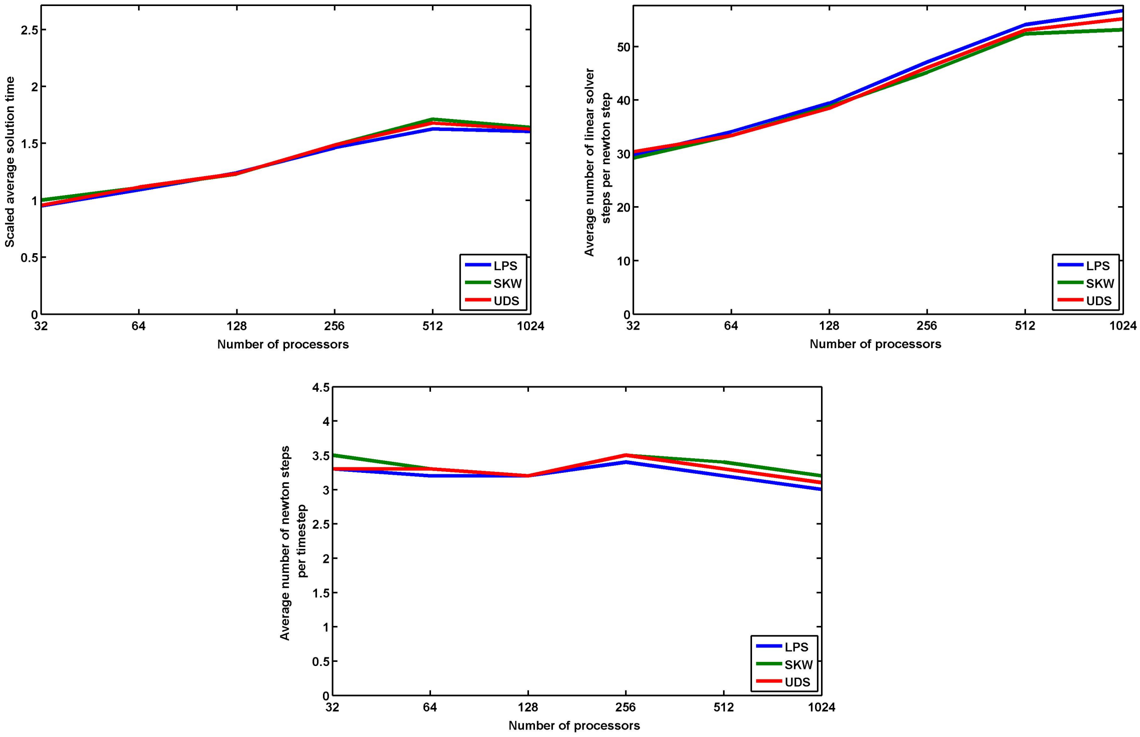


Figure 14.16. *We compare different upwind schemes for the geometric implicit case using the additive Schwarz method with an overlap of two. In the first row on the left we show solution times, on the right we show number of linear iterations per Newton step. In the second row we present the average number of Newton steps.*

The solution times for UDS, SKW and LPS behave similarly for all all overlaps. We may conclude that the stabilization of the fluid and the aspect ratio of the elements has a significant impact on the convergence rates of the geometric multigrid. Our results show that the two-level restricted additive Schwarz method is a robust and efficient preconditioning strategy for the coupled fluid structure interaction problem. Especially in combination with the geometric explicit discretization we get a good scaling behavior up to 1024 computing cores.

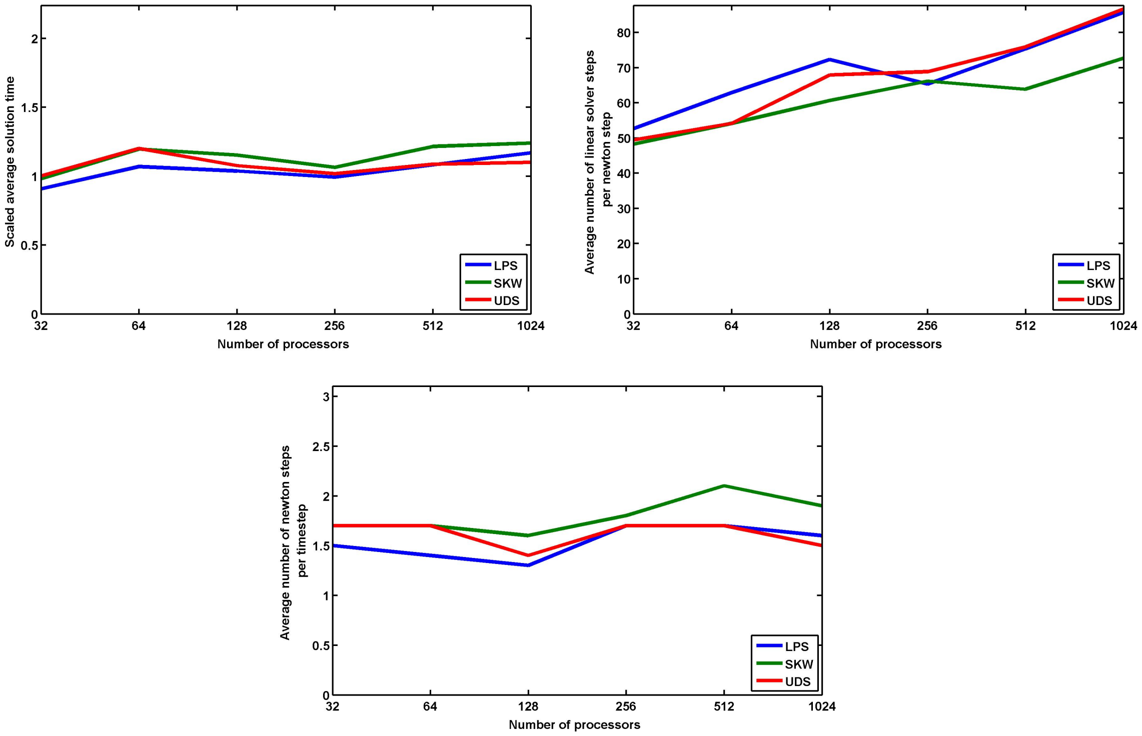


Figure 14.17. *We compare different upwind schemes for the geometric explicit case using the additive Schwarz method with an overlap of two. In the first row on the left we show solution times, on the right we show number of linear iterations per Newton step. In the second row we present the average number of Newton steps.*

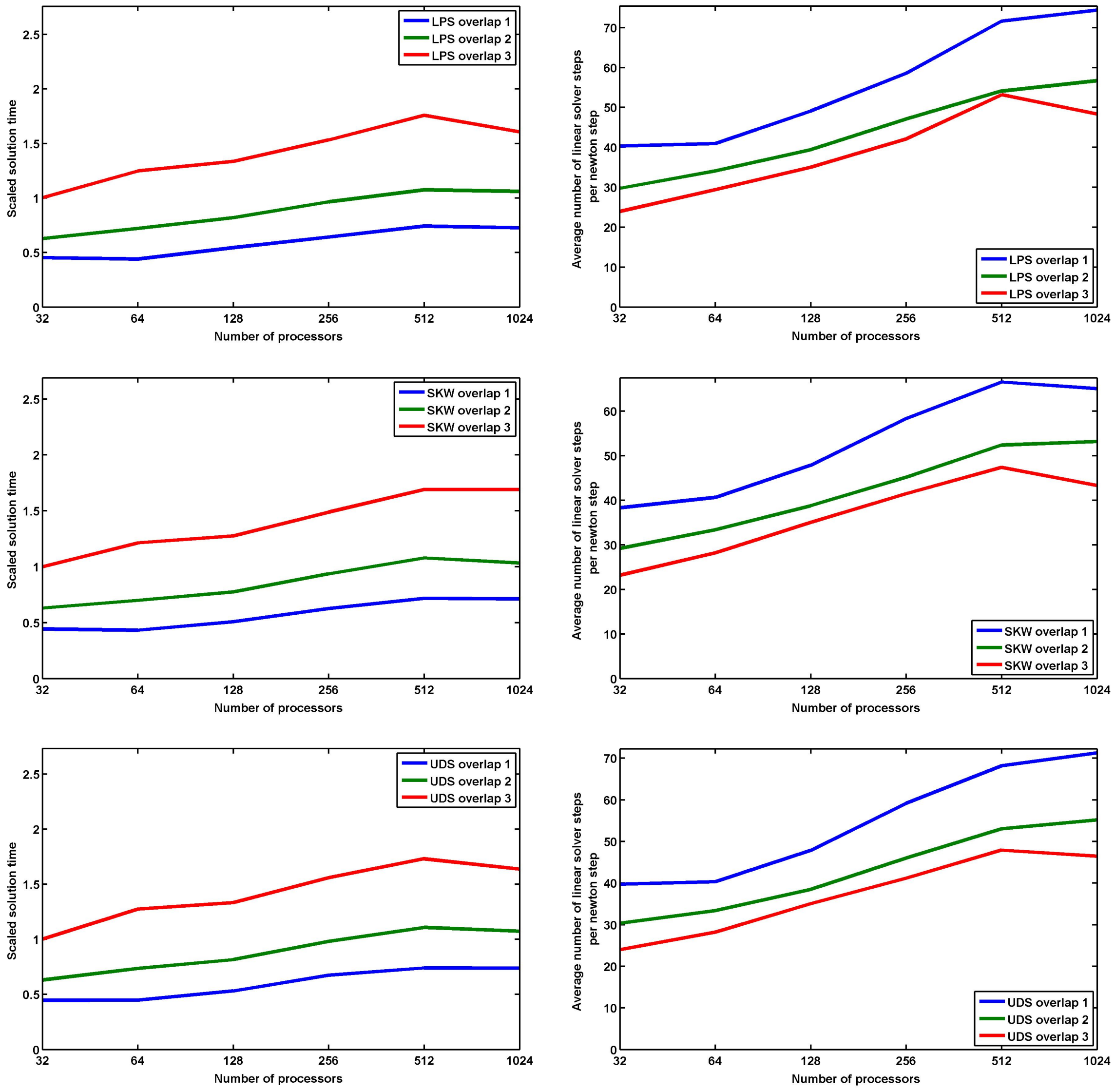


Figure 14.18. *Weak scaling for the additive Schwarz method in the geometric implicit case with an overlap of two. Left: solution time. Right: Average number of linear iterations per Newton iteration.*

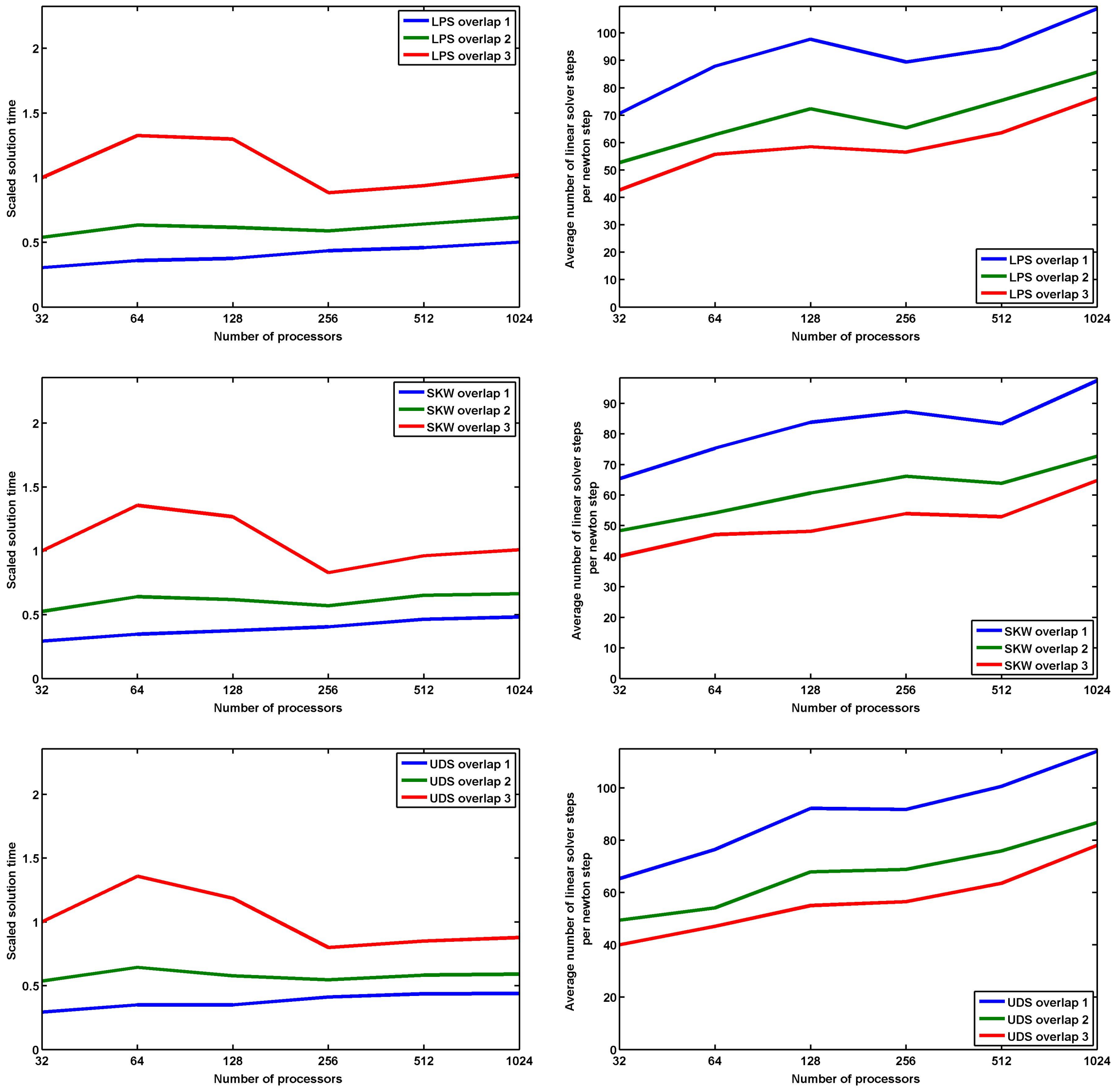


Figure 14.19. *Weak scaling for the additive Schwarz method in the geometric explicit case with an overlap of two. Left: solution time. Right: Average number of linear iterations per Newton iteration.*

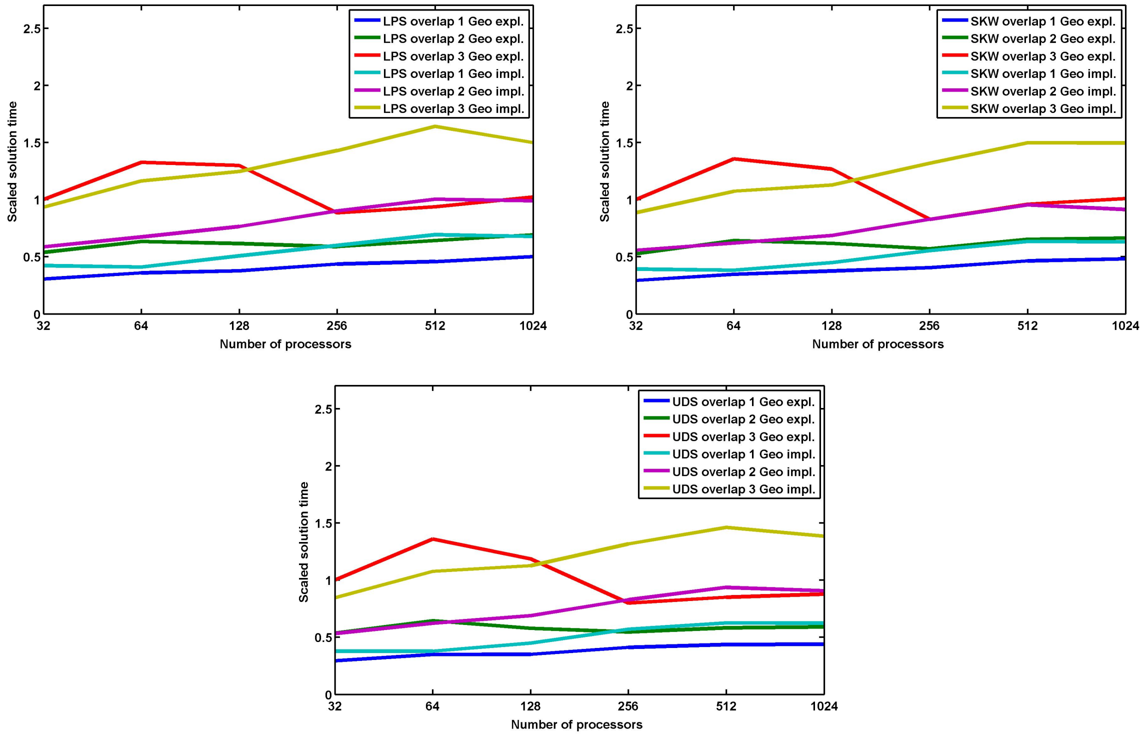


Figure 14.20. *Comparison of different upwind schemes for the geometric implicit and explicit case using different overlaps.*

15 Conclusion

In this thesis we present a coupling scheme that combines different discretizations in a monolithic approach for fluid structure interaction problems. A major challenge in fluid and structure interaction is the transfer of forces and velocities at the fluid structure interface in a stable and efficient way. The use of two different discretizations has so far been restricted to partitioned coupling concepts, where for many applications the design of a stable coupling is far from trivial. In a monolithic scheme the coupling is stable by design, since it is done in an implicit way. However, so far monolithic coupling schemes have only employed one type of discretization - typically the Finite Element method - even though the use of different tailored discretizations is very attractive in this context as well. The main contribution of the present work is the demonstration of a complete simulation framework that enables the scalable solution of large-scale fluid structure interaction problems using a heterogeneous monolithic coupling scheme.

In our method for the fluid discretization we use a vertex centered Finite Volume method where all unknowns are collocated at the mesh vertices. The structure is discretized using Finite Elements. The developed implementation was used for two different real world applications. First we describe an application from the field of bionics, where we consider the optimization of a sensor for the measurement of flow velocities. Second we describe an example in blood flow simulation. Since we simulate real world problems, as a consequence of the complexity of the geometries, we end up with algebraic systems with a large number of degrees of freedom. Therefore we have combined scalable solvers for the heterogeneous coupled problem. We use Newton's method to linearize the fully implicit coupled nonlinear fluid structure problem and apply a Krylov subspace method. For the Krylov method we present two different preconditioning strategies: a geometric multigrid and a restricted two-level additive Schwarz method based on a hierarchy of unstructured meshes. We model the fluid by means of the incompressible Navier Stokes equations in an Arbitrary-Eulerian-Lagrangian formulation. This leads to an additional quantity to be coupled, displacements for the fluid domain. For the arising fluid domain displacement problem we presented two strategies. In the first we solve the fluid domain problem using a Laplacian or an elasticity auxiliary problem, which is also fully implicitly coupled. In a second approach the fluid domain displacements in the interior of the fluid domain are handled explicitly, while the deformations of the fluid domain at the interface are handled fully implicitly. The geometric explicit strategy leads to a significant decrease in the number of Newton iterations compared to the geometric implicit coupling. We presented a comparison of the two preconditioning strategies in combination with the geometric implicit and explicit coupling of the fluid domain deformations for a benchmark problem. The two-level restricted additive Schwarz with the geometric explicit coupling approach is a robust solver and shows good scaling behavior up to 1024 processors.

Our implementation has been designed with modularity in mind such that the replacement or extension of subproblems is straightforward. A possible extensions for the structure subproblem is the use of a model for large displacements. As part of future work a turbulence model could be incorporated into the fluid subproblem or the compressible Navier Stokes equations could be used to model additional phenomena.

In our work we have developed a fully implicit heterogeneous coupling method for the simulation of fluid structure phenomena discretized with a mixed Finite Volume/Finite Element discretization. This thesis covers all aspects from the modeling to the scalable parallel solution of the arising problems with multi-level and domain decomposition techniques. The implementation developed for this work provides a solid foundation for further research.

Bibliography

[Ast10] M. Astorino. *Interaction Fluide-Structure dans le Système Cardiovasculaire. Analyse Numérique et Simulation*. These, INRIA; Université Pierre et Marie Curie-Paris VI, April 2010.

[BAA$^+$14a] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, K. Rupp, B. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11-Revision 3.5, Argonne National Laboratory, 2014.

[BAA$^+$14b] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, K. Rupp, B. Smith, and H. Zhang. PETSc Web page. `http://www.mcs.anl.gov/petsc`, 2014.

[Bar09] A.T. Barker. *Parallel monolithic fluid-structure interaction algorithms with application to blood flow simulation*. PhD thesis, 2009.

[BBC$^+$94] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, 1994.

[BBFS90] R. Bank, J. Bürgler, w. Fichtner, and R. Smith. Some upwinding techniques for finite element approximations of convection-diffusion equations. *Numerische Mathematik*, 58(1):185–202, 1990.

[BCHZ08] Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43(1):3–37, 2008.

[Bey98] J. Bey. *Finite-Volumen- und Mehrgitter-Verfahren für elliptische Randwertprobleme*. PhD thesis, Tübingen, Univ., Diss., 1998.

[BFDK$^+$03] M. Brewer, L. Freitag Diachin, P. Knupp, T. Leurent, and D. Melander. The mesquite mesh quality improvement toolkit. 2003.

[BG04] D. Boffi and L. Gastaldi. Stability and geometric conservation laws for ale formulations. *Computer Methods in Applied Mechanics and Engineering*, 193(42-44):4717–4739, 2004.

[BGMS97] S. Balay, W. Gropp, L. McInnes, and B. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[BHZ$^+$10] Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, X. Liang, T. Kvamsdal, R. Brekken, and J.G. Isaksen. A fully-coupled fluid-structure interaction simulation of cerebral aneurysms. *Computational Mechanics*, 46(1):3–16, 2010.

[BNV09] S. Badia, F. Nobile, and C. Vergara. Robin-Robin preconditioned krylov methods for fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 198(33-36):2768–2784, 2009.

[BQQ08] S. Badia, A. Quaini, and A. Quarteroni. Modular vs. non-modular preconditioners for fluid-structure systems with large added-mass effect. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50):4216–4232, 2008.

[Bra07] D. Braess. *Finite elements : Theory, Fast Solvers, and Applications in Elasticity Theory*. Cambridge University Press, 2007.

[BS06] H. Bungartz and M. Schafer. *Fluid-Structure Interaction-Modelling, Simulation, Optimisation*. Springer Science, Berlin Heidelberg, 2006. aufl. edition, 2006.

[BTT13] Y. Bazilevs, K. Takizawa, and T.E. Tezduyar. *Computational Fluid-Structure Interaction-Methods and Applications*. Wiley, New York, 1. auflage edition, 2013.

[BvZ03] H. Bijl and A.H. van Zuijlen. Efficient partitioned solution methods for the computation of fluid-structure interactions. In K.J. Bathe, editor, *Computational Fluid and Solid Mechanics 2003*, pages 1259–1262. Elsevier Science Ltd, Oxford, 2003.

[BW97] J. Bey and G. Wittum. Downwind numbering: robust multigrid for convection—diffusion problems. *Applied Numerical Mathematics*, 23(1):177–192, 1997.

[BW00] H. Braess and P. Wriggers. Arbitrary Lagrangian Eulerian finite element analysis of free surface flow. *Computer Methods in Applied Mechanics and Engineering*, 190(1-2):95–109, 2000.

[Cai91] X.C. Cai. Additive schwarz algorithms for parabolic convection-diffusion equations. *Numer. Math*, 60:41–61, 1991.

[CDFQ11] P. Crosetto, S. Deparis, G. Fourestey, and A. Quarteroni. Parallel algorithms for fluid-structure interaction problems in haemodynamics. *SIAM J. Sci. Comput.*, 33(4):1598–1622, July 2011.

[CGN05] P. Causin, J. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194:4506–4527, October 2005.

[Cia02] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems-*. SIAM, Philadelphia, 2nd revised edition edition, 2002.

[CLL06] C.Y. Chee, H.P. Lee, and C. Lu. Evaluating the biomechanical behaviour of the erythrocyte at large deformation using a three-dimensional fluid-structure interaction model. *Journal of Biomechanics*, 39(Supplement 1):S583–S583, 2006. Abstracts of the 5th World Congress of Biomechanics.

[CRD$^+$11] P. Crosetto, P. Reymond, S. Deparis, D. Kontaxakis, N. Stergiopulos, and A. Quarteroni. Fluid-structure interaction simulation of aortic blood flow. *Computers & Fluids*, 43(1):46–57, 2011.

[Cro11] P. Crosetto. *Fluid-Structure Interaction Problems in Hemodynamics*. PhD thesis, SB, Lausanne, 2011.

[DBHV10] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels. Bubble simulations with an interface tracking technique based on a partitioned fluid-structure interaction algorithm. *Journal of Computational and Applied Mathematics*, 234(7):2303–2310, 2010. Fourth International Conference on Advanced Computational Methods in ENgineering (ACOMEN 2008).

[Dep04] S. Deparis. *Numerical analysis of axisymmetric flows and methods for fluid-structure interaction arising in blood flow simulation*. PhD thesis, SB, Lausanne, 2004.

[DHA$^+$10] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, and J. Vierendeels. Performance of partitioned procedures in fluid-structure interaction. *Computers and Structures*, 88(7-8):446–457, 2010.

[DHBPS03] J. De Hart, F. Baaijens, G. Peters, and P. Schreurs. A computational fluid-structure interaction analysis of a fiber-reinforced stentless aortic valve. *Journal of Biomechanics*, 36(5):699–712, 2003. Cardiovascular Biomechanics.

[dL09] M. de Luca. *Mathematical and numerical models for cerebral aneurysm wall mechanics*. PhD thesis, Politecnico di Milano, 2009.

[DP07] W.G. Dettmer and D. Peric. A fully implicit computational strategy for strongly coupled fluid-solid interaction. *Archives of Computational Methods in Engineering*, 14(3):205–247, 2007.

[DR06] P. G. Drazin and N. Riley. *The Navier-Stokes Equations: A Classification of Flows and Exact Solutions (London Mathematical Society Lecture Note Series)*. Cambridge University Press, 2006.

[DSW93] M. Dryja, B. F. Smith, and O. Widlund. Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. *SIAM J. Numer. Anal*, 31:1662–1694, 1993.

[Elg99] H. Elgamel. A simple and efficient technique for the simulation of capacitive pressure transducers. *Sensors and Actuators A: Physical*, 77(3):183–186, 1999.

[FG04] C. Farhat and P. Geuzaine. Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids . *Computer Methods in Applied Mechanics and Engineering*, 193(39-41):4073–4095, 2004. The Arbitrary Lagrangian-Eulerian Formulation.

[FGG01] C. Farhat, P. Geuzaine, and C. Grandmont. The Discrete Geometric Conservation Law and the Nonlinear Stability of ALE Schemes for the Solution of Flow Problems on Moving Grids . *Journal of Computational Physics*, 174(2):669–694, 2001.

[FLLT98] C. Farhat, M. Lesoinne, and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157(1-2):95–114, 1998.

[FMRT08] C. Foias, O. Manley, R. Rosa, and R. Temam. *Navier-Stokes Equations and Turbulence (Encyclopedia of Mathematics and its Applications)*. Cambridge University Press, 2008.

[FN04] L. Formaggia and F. Nobile. Stability analysis of second-order time accurate schemes for ale-fem. *Computer Methods in Applied Mechanics and Engineering*, 193(39-41):4097–4116, 2004.

[For73] B. Fornberg. On the instability of leap-frog and Crank-Nicolson approximations of a nonlinear partial differential equation. *Mathematics of Computation*, 27(121):45–57, 1973.

[För07] C. Förster. *Robust methods for fluid-structure interaction with stabilised finite elements*. PhD thesis, Stuttgart, Univ., Diss., 2007.

[FOT$^+$06] H. Fukunari, M. Oshima, R. Torii, H. Watanabe, and T. Hisada. Fluid-structure interaction finite element analysis of middle cerebral artery aneurysm. *Journal of Biomechanics*, 39(Supplement 1):S364–S364, 2006. Abstracts of the 5th World Congress of Biomechanics.

[FP99] J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer, Berlin, 1999.

[FS98] A. Frommer and B. Szyld. Weighted Max Norms, Splittings, and Overlapping Additive Schwarz Iterations, 1998.

[Gal94] G.P. Galdi. *An Introduction to the Mathematical Theory of the Navier-Stokes Equations: Volume 2: Nonlinear Steady Problems*. Springer, 1994.

[GF00] H. Guillard and C. Farhat. On the significance of the geometric conservation law for flow computations on moving meshes. *Computer Methods in Applied Mechanics and Engineering*, 190(11-12):1467–1482, 2000.

[GR05] C. Grossmann and H. Roos. *Numerische Behandlung partieller Differentialgleichungen*. Vieweg+Teubner Verlag, Wiesbaden, 2005.

[GV03] J.F. Gerbeau and M. Vidrascu. A quasi-newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37:631–647, 7 2003.

[Hac85] W. Hackbusch. *Multi-grid methods and applications*. Springer-Verlag, Berlin New York, 1985.

[Hau10] A. Hauser. *Large-Eddy-Simulation auf uniform und adaptiv verfeinerten Gittern: adaptive LES, Diskretisierungs- und Modellfehler bei der LES, Wärme- und Stofftransport, Mehrgitterverfahren, statischer Mischer*. PhD thesis, Heidelberg, Univ., Diss., 2010.

[Hel03] B. Helenbrook. Mesh deformation using the biharmonic operator. *International Journal for Numerical Methods in Engineering*, 56(7):1007–1021, 2003.

[HHB08] M. Heil, A. Hazel, and J. Boyle. Solvers for large-displacement fluid-structure interaction problems: segregated versus monolithic approaches. *Computational Mechanics*, 43(1):91–101, 2008.

[Hir07] C. Hirsch. *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics-The Fundamentals of Computational Fluid Dynamics*. Butterworth-Heinemann, Oxford, 2007.

[HP97] W. Hackbusch and T. Probst. Downwind Gauß-Seidel Smoothing for Convection Dominated Problems. *Numerical Linear Algebra with Applications*, 4(2):85–102, 1997.

[HR90] J. Heywood and R. Rannacher. Finite-element approximation of the nonstationary navier-stokes problem. part iv: Error analysis for second-order time discretization. *SIAM Journal on Numerical Analysis*, 27(2):353–384, 1990.

[Hro07] J. Hron. Fluid-structure interaction with applications in biomechanics. *Nonlinear Analysis: Real World Applications*, 8(5):1431–1458, 2007.

[HS91] M. Hegland and P. Saylor. Block Jacobi Preconditioning of the Conjugate Gradient Method on a Vector Processor, 1991.

[HWD04] B. Hübner, E. Walhorn, and D. Dinkler. A monolithic approach to fluid-structure interaction using space-time finite elements. *Computer Methods in Applied Mechanics and Engineering*, 193(23-26):2087–2104, 2004.

[Joh09] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover Publications, New York, dover. edition, 2009.

[KAS08] A.R. Khoei, M. Anahid, and K. Shahim. An extended arbitrary Lagrangian-Eulerian finite element method for large deformation of solid mechanics. *Finite Elements in Analysis and Design*, 44(6-7):401–416, 2008.

[KBB09] K. Khanafer, J. Bull, and R. Berguer. Fluid-structure interaction of turbulent pulsatile flow within a flexible wall axisymmetric aortic aneurysm model. *European Journal of Mechanics-B/Fluids*, 28(1):88–102, 2009.

[KF99] B. Koobus and C. Farhat. Second-order time-accurate and geometrically conservative implicit schemes for flow computations on unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering*, 170:103–129, 1999.

[KGF$^+$10] U. Küttler, M. Gee, Ch. Förster, A. Comerford, and W. A. Wall. Coupling strategies for biomedical fluid-structure interaction problems. *International Journal for Numerical Methods in Biomedical Engineering*, 26(3-4):305–321, 2010.

[KNE$^+$08] S. Kock, J. Nygaard, N. Eldrup, E. Fründ, A Klærke, W. Paaske, E/ Falk, and W. Kim. Mechanical stresses in carotid plaques using MRI-based fluid-structure interaction models. *Journal of Biomechanics*, 41(8):1651–1658, 2008.

[KS95] S. M. H. Karimian and G. E. Schneider. Pressure-based control-volume finite element method for flow at all speeds. *AIAA Journal*, 33(9):1611–1618, 2014/08/05 1995.

[KS04] R. Kamakoti and W. Shyy. Fluid-structure interaction for aeroelastic applications. *Progress in Aerospace Sciences*, 40(8):535–558, 2004.

[KTZ09] I. Kukavica, A. Tuffaha, and M. Ziane. Strong solutions to a nonlinear fluid structure interaction system. *Journal of Differential Equations*, 247(5):1452–1478, 2009.

[LDV96] R. D. Lazarov, Mishev I. D., and P. S. Vassilevski. Finite volume methods for convection-diffusion problems. *SIAM J. Numer. Anal*, 33:31–55, 1996.

[LeV02] R. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, Cambridge, 2002.

[Lio78] P.L. Lions. Interpretation stochastique de la methode alternee de Schwarz. *C. R. Acad. Sci. Paris*, pages 325–328, 1978.

[LTM01] P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3039–3067, 2001.

[LY04] D. C. Lo and D. L. Young. Arbitrary Lagrangian-Eulerian finite element analysis of free surface flow using a velocity-vorticity formulation. *Journal of Computational Physics*, 195(1):175–201, 2004.

[MAFB$^+$10] F. D. Molina-Aiz, H. Fatnassi, T. Boulard, J. C. Roy, and D. L. Valera. Comparison of finite element and finite volume methods for simulation of natural ventilation in greenhouses. *Comput. Electron. Agric.*, 72(2):69–86, July 2010.

[Met04] M. Metzner. *Mehrgitterverfahren für die kompressiblen Euler- und Navier-Stokes-Gleichungen mit besonderer Betrachtung des schwach kompressiblen Falles*. PhD thesis, Heidelberg, Univ., Diss., 2004.

[MvBdB05] C. Michler, E. H. van Brummelen, and R. de Borst. An interface Newton-Krylov solver for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 47(10-11):1189–1195, 2005.

[Näg04] S. Nägele. *Mehrgitterverfahren für die inkompressiblen Navier-Stokes Gleichungen im laminaren und turbulenten Regime unter Berücksichtigung verschiedener Stabilisierungsmethoden*. PhD thesis, Heidelberg, Univ., Diss., 2004.

[Ngu10] V. Nguyen. An arbitrary Lagrangian-Eulerian discontinuous Galerkin method for simulations of flows over variable geometries. *Journal of Fluids and Structures*, 26(2):312–329, 2010.

[Nob01] F. Nobile. *Numerical approximation of fluid-structure interaction problems with application to haemodynamics*. PhD thesis, SB, Lausanne, 2001.

[NS02] R. Nabben and D. B. Szyld. Convergence Theory of Restricted Multiplicative Schwarz Methods. *SIAM J. Numer. Anal.*, 40:2318–2336, June 2002.

[NW07] S. Nägele and G. Wittum. On the influence of different stabilisation methods for the incompressible Navier-Stokes equations . *Journal of Computational Physics*, 224(1):100–116, 2007. Special Issue Dedicated to Professor Piet Wesseling on the occasion of his retirement from Delft University of Technology.

[Ode91] J. Oden. Finite elements: An introduction. In *Finite Element Method*, volume 2 of *Handbook of Numerical Analysis*. Elsevier, 1991.

[Oha01] R. Ohayon. Reduced symmetric models for modal analysis of internal structural-acoustic and hydroelastic-sloshing systems. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3009–3019, 2001.

[Pen09] G. Pena. *Spectral element approximation of the incompressible Navier-Stokes equations in a moving domain and applications.* PhD thesis, SB, Lausanne, 2009.

[Pfl99] C. Pflaum. Robust convergence of multilevel algorithms for convection-diffusion equations. *SIAM J. Numer. Anal.*, 37(2):443–469, December 1999.

[PRK07] I. Pivkin, P. Richardson, and G. Karniadakis. Dissipative Particle Dynamics Simulations of Deformable Red Blood Cells in Small Blood Vessels. In *APS Meeting Abstracts*, page 34012, March 2007.

[Qua09] A. Quaini. *Algorithms for fluid-structure interaction problems arising in hemodynamics.* PhD thesis, SB, Lausanne, 2009.

[QV99] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations (Numerical Mathematics and Scientific Computation).* Oxford University Press, USA, 1999.

[QV08] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations.* Springer Series in Computational Mathematics. Springer, 2008.

[Raw85] M.J. Raw. *A New Control-volume-based Finite Element Procedure for the Numerical Solution of the Fluid Flow and Scalar Transport Equations.* PhD thesis, University of Waterloo. Dept. of Mechanical Engineering, 1985.

[RR96] H. Rentz-Reichert. *Robuste Mehrgitterverfahren zur Lösung der inkompressiblen Navier-Stokes-Gleichung: ein Vergleich.* PhD thesis, Stuttgart, Univ., Diss., 1996.

[RRIOn10] P.B. Ryzhakov, R. Rossi, S.R. Idelsohn, and E. Oñate. A monolithic Lagrangian approach for fluid-structure interaction problems. *Computational Mechanics*, 46(6):883–899, 2010.

[RSR04] A. Rosset, L. Spadola, and O. Ratib. Osirix: An open-source software for navigating in multidimensional dicom images. *Journal of Digital Imaging*, 17(3):205–216, 2004.

[RST08] H. Roos, M. Stynes, and L. Tobiska. *Robust Numerical Methods for Singularly Perturbed Differential Equations-Convection-Diffusion-Reaction and Flow Problems.* Springer Science and Business Media, Berlin Heidelberg, 2008.

[RT74] G.D. Raithby and K.E. Torrance. Upstream-weighted differencing schemes and their application to elliptic problems involving fluid flow. *Computers and Fluids*, 2(2):191–206, 1974.

[SBBN00] D. B. Szyld, M. Benzi, M. Benzi, and R. Nabben. Algebraic Theory of Multiplicative Schwarz Methods. *Numer. Math*, 89(89):605–639, 2000.

[SBC$^+$07] S. Sathe, R. Benney, R. Charles, E. Doucette, J. Miletti, M. Senga, K. Stein, and T.E. Tezduyar. Fluid-structure interaction modeling of complex parachute designs with the space-time finite element techniques. *Computers and Fluids*, 36(1):127–135, 2007. Challenges and Advances in Flow Simulation and Modeling.

[SBG96] B. F. Smith, P. E. Bjørstad, and W. D. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, New York, NY, USA, 1996.

[SBG04] B. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations (Volume 0)*. Cambridge University Press, 2004.

[SBTP01] K. Stein, R. Benney, T. Tezduyar, and J. Potvin. Fluid-structure interactions of a cross parachute: numerical simulation. *Computer Methods in Applied Mechanics and Engineering*, 191(6-7):673–687, 2001.

[SL04] W. Shangguan and Z. Lu. Modelling of a hydraulic engine mount with fluid-structure interaction finite element analysis. *Journal of Sound and Vibration*, 275(1-2):193–221, 2004.

[SR87] G. E. Schneider and M. J. Raw. Control volume finite-element method for heat transfer and fluid flow using colocated variables-1. computational procedure. *Numerical Heat Transfer*, 11(4):363–390, 1987.

[STB03] K. Stein, T. Tezduyar, and R. Benney. Mesh moving techniques for fluid-structure interactions with large displacements. *Journal of Applied Mechanics*, 70(1):58–63, 01 2003.

[SWH05] D. Stalling, M. Westerhoff, and H. Hege. Amira: a highly interactive system for visual data analysis. 2005.

[TBMJ92] T. E. Tezduyar, M. Behr, S. Mittal, and A. Johnson. Computation of unsteady incompressible flows with the stabilized finite element methods: Space-time formulations, iterative strategies and massively parallel implementations. *ASME Pressure Vessels Piping Div Publ PVP*, 246:7–24, 1992.

[Tea] Cubit Development Team. Cubit mesh generation environment volume 1: Users manual.

[TL79] P. D. Thomas and C. K. Lombard. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal*, 17(10):1030–1037, 2014/04/10 1979.

[TO01] T. Tezduyar and Y. Osawa. Fluid-structure interactions of a parachute crossing the far wake of an aircraft. *Computer Methods in Applied Mechanics and Engineering*, 191(6-7):717–726, 2001.

[TSS06] T. E. Tezduyar, S. Sathe, and K. Stein. Solution techniques for the fully discretized equations in computation of fluid-structure interactions with the space-time formulations. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5743–5753, 2006.

[TW10] A. Toselli and O. Widlund. *Domain Decomposition Methods-Algorithms and Theory (Springer Series in Computational Mathematics)*. Springer, 2010.

[TYW$^+$02] D. Tang, C. Yang, H. Walker, S. Kobayashi, and D. Ku. Simulating cyclic artery compression using a 3d unsteady model with fluid-structure interactions. *Computers and Structures*, 80(20-21):1651–1665, 2002.

[Wes04] P. Wesseling. *An introduction to multigrid methods*. R.T. Edwards, Philadelphia, 2004.

[Wic11] T. Wick. *Adaptive finite element simulation of fluid-structure interaction with application to heart-valve dynamics*. PhD thesis, Heidelberg, Univ., Diss., 2011, 2011.

[Wic13] T. Wick. Stability estimates and numerical comparison of second order time-stepping schemes for fluid-structure interactions. In Andrea Cangiani, Ruslan L. Davidchack, Emmanuil Georgoulis, Alexander N. Gorban, Jeremy Levesley, and Michael V. Tretyakov, editors, *Numerical Mathematics and Advanced Applications 2011*, pages 625–632. Springer Berlin Heidelberg, 2013.

[WKHD05] E. Walhorn, A. Kölke, B. Hübner, and D. Dinkler. Fluid-structure coupling within a monolithic model involving free surface flows. *Computers and Structures*, 83(25-26):2100–2111, 2005.

[Woh01] B. I. Wohlmuth. *Discretization Methods and Iterative Solvers Based on Domain Decomposition (Lecture Notes in Computational Science and Engineering)*. Springer, 2001.

[WRS$^+$05] B. Wolters, M. Rutten, G. Schurink, U. Kose, J. de Hart, and F. van de Vosse. A patient-specific computational model of fluid-structure interaction in abdominal aortic aneurysms. *Medical Engineering and Physics*, 27(10):871–883, 2005.

[WSKH04] H. Watanabe, S. Sugiura, H. Kafuku, and T. Hisada. Multiphysics simulation of left ventricular filling dynamics using fluid-structure interaction finite element method. *Biophysical Journal*, 87(3):2074–2085, 2004.

[Xu92] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, 34:581–613, December 1992.

[XZ98] J. Xu and J. Zou. Some nonoverlapping domain decomposition methods. *SIAM Rev.*, 40:857–914, December 1998.

[ZRTC93] H. Zhang, M. Reggio, J. Trépanier, and R. Camarero. Discrete form of the GCL for moving meshes and its implementation in CFD schemes. *Computers and Fluids*, 22(1):9–23, 1993.