Technische Universität München
Lehrstuhl Computation in Engineering

ALE-type and fixed grid fluid-structure interaction involving the p-version of the Finite Element Method

Stefan Kollmannsberger

Vollständiger Abdruck der von der Fakultät für Bauingenieur- und Vermessungswesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. K.-U. Bletzinger
Prüfer der Dissertation:		
	1.	Univ.-Prof. Dr.rer.nat. E. Rank
	2.	Univ.-Prof. Dr.-Ing. habil. M. Krafczyk,
		Technische Universität Carolo-Wilhelmina zu Braunschweig

Die Dissertation wurde am 24.09.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Bauingenieur- und Vermessungswesen am 15.02.2010 angenommen.

para las pajaritas

Abstract

This treatise addresses the efficient, numerical simulation of the interaction between fluids and structures. The discretization of the structure is either based on high-order hexahedral or quadrilateral elements (p-version). Both types of elements allow for an independent choice of the polynomial degrees for the different local directions as well as for the different components of the cartesian displacement vectors, which provides scope for strictly two- or three-dimensional structural discretizations. Three different approaches for the fluid are regarded: the Finite Volume Method, the Spectral Element Method and the Lattice Boltzmann Method. Whereas the first two approaches discretize the ALE form of the incompressible Navier-Stokes equations, the Lattice Boltzmann Method discretizes the Boltzmann equation stemming from the kinetic gas theory with a fixed, hierarchical grid. Algorithms are suggested for the coupling of these discretizations to the structural p-version, which are then verified and validated against benchmark examples. In this context, the coupling to the Lattice Boltzmann Method turned out to be particularly favourable.

Zusammenfassung

Diese Arbeit befasst sich mit der effizienten, numerischen Simulation der gegenseitigen Wechselwirkung zwischen Fluiden und Strukturen. Die Diskretisierung der Struktur basiert auf Hexaeder- oder auf Viereckselementen hoher Ordnung (p-Version). Beide Elementtypen erlauben eine anisotrope Wahl der Polynomgrade für die lokalen Richtungen und für die Komponenten des kartesischen Verschiebungsvektors. Dies ermöglicht effiziente, strikt zwei- oder dreidimensionale Diskretisierungen. Für das Fluid werden unterschiedliche Ansätze verwendet: eine Finite Volumen Methode, eine Spektralelement Methode und eine Lattice Boltzmann Methode. Die ersten beiden Ansätze diskretisieren die ALE-Form der inkompressiblen Navier-Stokes Gleichung; die Lattice Boltzmann Methode hingegen diskretisiert die aus der Gaskinetik stammende Boltzmanngleichung mit einem Finite Differenzen Ansatz auf einem ortsfesten, hierarchischen Gitter. Es werden Verfahren zur Kopplung dieser Ansätze an die mit der p-Version diskretisierten Struktur vorgeschlagen und an Referenzbeispielen verifiziert und validiert. Hierbei hat sich die Kopplung an die Lattice Boltzmann Methode als besonders vorteilhaft herausgestellt.

Preamble

This treatise emerged as part of my work at the *Chair for Computation in Engineering* in the framework of the project *Elemente hoher Ordung in der Fluid-Struktur Wechselwirkung* from fall 2005 to 2009. It was funded by the *Deutsche Forschungsgemeinschaft*.

I would now like to express my gratitude to everyone who has contributed to this thesis.

Special thanks go to the best boss and supervisor Prof. Dr.rer.nat. Ernst Rank. His enthusiasm to get involved creatively in many diverse topics was and still is most inspiring.

I would like to thank Prof. Dr.-Ing. habil. Manfred Krafczyk for accepting to act as a second examiner. I am proud to have been guided by such a renowned expert in the field.

To Prof. Dr.-Ing. habil. Alexander Düster who co–initiated this project. His door was always open and questions were never left unanswered. Thanks for the ongoing interest and support.

A large part of this thesis contains results achieved in close cooperation with Sebastian Geller. One common goal and eternal debugging sessions made us understand each others way of thinking. It was fun to tackle all these difficulties with you, Sebastian.

There are may others who have contributed to this work. Some, like Sebastian, also became friends along the way. Iason Papaioannou —Greek genius— and Christian Sorger —the sports man. Both have contributed to this treatise in form of their own master thesis and assistant work. Martin Schlaffer: thanks for being such a good partner in discussion and reflection.

Thank you Dominik Scholz, my predecessor in this project, for not leaving a mess behind when you left the chair and for introducing me to the topic. And thanks to Georg Sehlhorst for critical but constructive comments and being a fun office mate for the largest part of this thesis.

I am very grateful to my parents who are always there when I need their support and formed me as a person. Looks like their efforts went somewhere.

Thanks to Mirely who gave up her previous life for love, including a Fulbright scholarship and a PhD position in the US. I still find it amazing that she did this to be with me in freaking cold Germany. Without her and our daughter Abi, most of the things I do would be less than half the fun.

Stefan Kollmannsberger,
Winter 2009/2010

Contents

Chapter 1

Introduction

Computational fluid-structure interaction aims to predict the behaviour of fluids and structures as a result of their interplay. The basic idea is simple and easily explained by imagining a leaf on a tree subject to wind. The air i.e. the fluid flows around the leaf thereby exerting forces on it. The leaf i.e. the structure deflects. As a result, the wind is forced to change its direction in the proximity of the leaf which in turn changes the forces exerted on the structure et cetera. The deflection of the leave may become static after a while or its motion may continue periodically or a-periodically. In the worst case the motion of the leaf may become so large as to detach the leaf from the tree or lead to its destruction. These effects may not be computed by only considering one physical field. On the contrary, only the consideration of their interplay leads to a concise description forming the basis for drawing conclusions and making forecasts.

The same situation may arise in aircraft wings, turbine blades, tall buildings, membrane or shell like structures in civil engineering, marine engineering or blood flow in arteries. Fluid-structure interaction is thus an important phenomenon. As a matter of course, it is interesting to know in advance if and how an aircraft wing or turbine blade vibrates, if and how tall buildings, membrane or shell structures deform under wind loads and if or how aneurysms in arteries have to be treated. All of these phenomena have in common that the two fields are interacting at their boundary. As such, they are termed surface coupled problems.

Civil i.e. non military applications of surface coupled fluid-structure interaction are countless. Yet, it comes at no surprise that first investigations into computational algorithms were carried out in a military context as means of studying submarines [43].

Many approaches and methods emerged from the intensive research carried out on the subject since. In an effort to evaluate selected methods from the broad range of possibilities of computing fluid-structure interaction, the “Deutsche Forschungsgemeinschaft” has established the research group 493. One goal of this research group was to set up and establish a Benchmark for fluid-structure interaction similar to the famous cylinder in a cross flow benchmark [133]. This benchmark exists in two forms, one of purely numerical nature and one set up as an experiment. The results of these Benchmarks will be published in a common paper and presented to the community in upcoming conferences. It is to serve industry and academia alike to evaluate new methods, calibrate new models or verify and validate computational codes.

On the most general level and following [147], fluid-structure interaction can be divided into two main problem categories. Class one problems have two physically different domains whose common boundary defines an interface, such as a solid body submerged or in contact with a fluid. Class two problems deal with overlapping fluid and structural domains, such as seepage through porous media. In this thesis class one problems are addressed in which the submerged structural body undergoes large (i.e. geometrically nonlinear), transient deformations.

There are two different approaches for solving this type of fluid-structure interaction. The monolithic approach discretizes the two separate domains using a similar discretization scheme and solves the resulting equation system within *one* solver. The compatibility conditions at the interface are treated inherently within this equation system. By contrast, the partitioned approach uses separate solvers for the fluid and the structural system. The two solvers then need to communicate on their common boundary to fulfill the continuity conditions required on this interface. In this thesis such a partitioned solution strategy is employed.

In a partitioned approach, each domain may utilize any type of discretization considered efficient for its field. For the fluid, however, two main approaches have been employed in the context of partitioned *f*luid-*s*tructure *i*nteraction (FSI) for large deformations. The fluid is usually described either (a) on a suitably moving grid (*A*rbitrary *L*agrangian *E*ulerian (ALE)-formulation) or (b) on a fixed Cartesian grid. In both cases, the geometry of the moving boundary needs to be described explicitly by means of an interface. From a geometric point of view, the main difference between (a) and (b) is that, for the ALE-formulation, the movement of the boundary is propagated into the fluid domain i.e. the fluid discretization needs to follow the description of the interface not only at the boundary but also in the interior of the domain. Regardless of the underlying discretization method (Finite Elements, Finite Volumes), the fluid is usually discretized by means of a mesh, which needs to fulfill certain geometric requirements (e.g. element size and aspect ratio). This, in turn, restricts the movement of the boundary. Difficulties arise in particular in situations where boundary deflections are very large with regard to the fluid domain concerned, which may make it necessary to remesh the fluid domain. Remeshing can be a challenging task, especially in three dimensions, not only because of the meshing process itself, but also because of the need to transfer the data between the old and the new mesh correctly.

In case (b) these restrictions do not apply. Here, the structure can move freely over the fixed Cartesian fluid grid without imposing restrictions onto the discretization of the fluid. The difficulty hereby lies elsewhere, namely in establishing the interface conditions between the structure and the fluid. Another challenge in case (b) is to accurately resolve boundary layers as body fitted meshes are not innate to this formulation. However, at the boundary, adaptive mesh refinement may be used to address this issue.

A combination of cases (a) and (b) is also possible but not discussed in this thesis. One approach proposed by [47] is to use a Chimera scheme in which a fixed Eulerian fluid background mesh is overlapped by a moving, deformable fluid mesh, called patch, which is attached to the structure and uses an ALE formulation. The interaction is then preferably achieved by performing a Dirichlet-Neumann or a Dirichlet-Robin type coupling. Here, the forces are transferred from the fixed background mesh to the outer boundary of the moving patch and the velocities from the structural boundary to the patch and from there to the fixed background mesh. This approach can provide a remedy to the body fitted mesh problems mentioned above while simultaneously allowing for large deflections.

Another interesting approach utilizing an XFEM formulation for the fluid is taken by [54, 86, 138]. An interface mesh is introduced to which both solvers couple by means of mortar methods. To be able to represent the interface sharply, the fluid domain elements are enriched by step functions to deal with the jumps occurring in the pressure and velocity fields when the structure strides over the fluid. Both approaches are compared in [139, 55].

For the sake of completeness, the Particle Element Method proposed in [69], is mentioned as a representative example of another approach for solving FSI problems with large deformations. A good overview, including a comparison between the methods belonging to categories (a) and (b), but with a different focus, is also given in [87]. To complete the overview it is remarked that partitioned fluid-structure interaction can be considered as an application of domain decomposition methods into which [131] gives a purely mathematical insight.

In this thesis both cases (a) and (b) are investigated where the structure is always discretized via finite elements of high order.

This thesis is structured as follows:

The structural part is covered in Chapter 2 which introduces the relevant equations describing the dynamics of deformable bodies and their discretization by means of the p-version of the Finite Element Method. This method is used throughout the thesis for the computation of the structure.

The basic fluid equations are treated in Chapter 3. First, the fluid is described as a continuum in Chapter 3.1. The resulting equations form the basis for a discretization on deformable grids corresponding to a computation of fluid-structure interaction of type (a). Chapter 3.2 briefly describes the Finite Volume Method (FVM) while Chapter 3.3 introduces the Spectral Element Method (SEM) as two representatives of discretization methods for fluids. The Lattice Boltzmann Method is introduced in Chapter 3.4 as an attractive alternative. It stems from gas dynamics and treats the fluid by modelling the interaction of particles. The Lattice Boltzmann Method discretizes the fluid via a fixed, undeformable grid. It is utilized in this thesis as a representative for fluid-structure interaction of type (b).

Some relevant algorithms concerning partitioned fluid-structure interaction for surface coupled problems are introduced in Chapter 4 and put into perspective. Utilizing high order finite element methods in fluid-structure interaction requires a close look to the data exchange at the interface and demands methods beyond the current standard in Fluid Structure interaction which are introduced here as well.

All methods are evaluated against the Benchmarks defined in Chapter 5. These have been the result of the common effort of the Research Group 496 by the "Deutsche Forschungsgemeinschaft" and the author of this thesis is a member to this group. These methods presented in the previous chapters are applied and refined in Chapter 6 where the fluid is computed by the commercial Finite Volume solver CFX and in Chapter 7 where the fluid is discretized by the Spectral Element Method solver $N\epsilon\kappa\tau\alpha\rho$. Here, the limits of the methods w.r.t. fluid structure interaction with very large deflections becomes eminent. A fixed grid type coupling (case (b)) to a lattice Boltzmann solver is verified and validated in Chapter 8. This approach can be a remedy to some of the problems encountered in an ALE formulation.

As the combination of p-FEM and lattice Boltzmann turned out to be an efficient alternative, some show case examples are presented in Chapter 9. They motivate the use of these methods beyond Benchmarks towards examples possibly of interest to industrial applications before conclusions are drawn in Chapter 10.

Chapter 2

Computational structural dynamics

This chapter serves to introduce only the most important concepts of geometrically non-linear structural dynamics. This theory forms the basis upon which the structural solver AdhoC is based. These concepts are not new and have been presented countless times by numerous authors of which Wriggers [143], Bonet and Wood [11] and Szabo [125] were found to be most helpful. The following quick introduction is based on the notation in [143].

2.1 Basic mechanics for large deflections

In the Lagrangian description commonly employed for structural mechanics, the coordinate system is attached to the particles of the continuum and deforms with them. In this context, states of particles before deformation are denoted by capital letters or / and an index $\cdot_0$. This undeformed configuration is also called material or initial configuration. States of particles after a deformation are labeled by minuscules. The deformed state is termed spatial configuration.

The position of a particle is described by its position vector $\mathbf{X}$ before and by $\mathbf{x}$ after deformation as depicted in Figure 2.1.

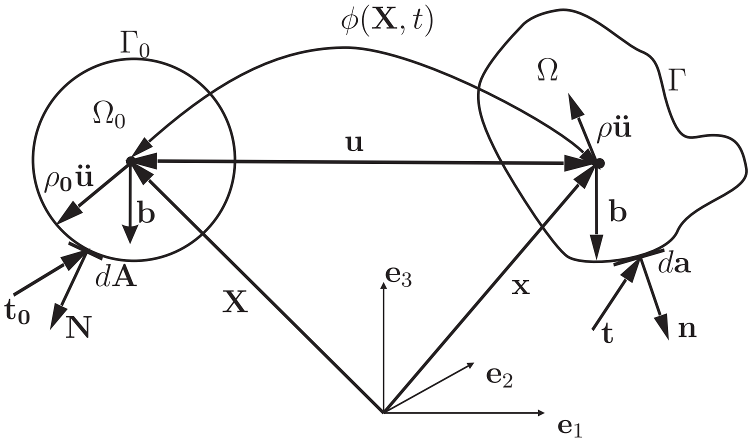


Figure 2.1: Configurations of a particle and forces acting on a domain Ω within a boundary Γ

The domain of the body where the particle is contained is generally denoted by Ω or by A in the case it is stressed that a two dimensional body is described. Its boundary is in general denoted either by Γ or by ∂A for a two dimensional description. The particle deformation is characterized by the time dependent mapping:

$$\mathbf{x}(t) = \phi(\mathbf{X}, t) \tag{2.1}$$

The displacement of a particle is the difference of its position vectors:

$$\mathbf{u} = \mathbf{x} - \mathbf{X} \tag{2.2}$$

The change of the deformed configuration with respect to the initial configuration is expressed by the gradient of the mapping Equation (2.1) and termed deformation gradient $\mathbf{F}$:

$$\mathbf{F} = \frac{d\mathbf{x}}{d\mathbf{X}} = \mathrm{Grad}\,\mathbf{x} = \mathrm{Grad}\,\phi(\mathbf{X}, t) = \mathrm{Grad}\mathbf{u} + \mathbf{I} \tag{2.3}$$

The unit tensor is denoted by $\mathbf{I}$. While $\mathbf{F}$ describes the absolute change of the position of a particle, strain is defined as the change of the positions of two particles relative to each other. It is determined by computing the difference of the square of the positions of the two particles before and after deformation:

$$\begin{aligned} d\mathbf{x}^T d\mathbf{x} - d\mathbf{X}^T d\mathbf{X} &= (\mathbf{F}d\mathbf{X})^T \mathbf{F} d\mathbf{X} - d\mathbf{X}^T d\mathbf{X} \\ &= d\mathbf{X}^T \left(\mathbf{F}^T \mathbf{F}\right) d\mathbf{X} - d\mathbf{X}^T \mathbf{I}\, d\mathbf{X} \\ &= d\mathbf{X}^T \left(\mathbf{F}^T \mathbf{F} - \mathbf{I}\right) d\mathbf{X} \\ &= d\mathbf{X}^T 2\mathbf{E}\, d\mathbf{X} \end{aligned} \tag{2.4}$$

where

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{I}) \tag{2.5}$$

is the Green-Lagrangian strain tensor. It describes the relation between strains and displacements with respect to (w.r.t.) the undeformed configuration.

While the undeformed volume is obtained by $dV = dX_1 dX_2 dX_3$, changes of volume and area can be computed by:

$$J = det\mathbf{F} \tag{2.6}$$

$$d\mathbf{a} = J\,\mathbf{F}^{-T} d\mathbf{A} \tag{2.7}$$

$$dv = JdV \tag{2.8}$$

where $d\mathbf{a} = \mathbf{n}da$ and $d\mathbf{A} = \mathbf{N}dA$ with $\mathbf{n}$ and $\mathbf{N}$ being the unit normal vector perpendicular to the surface element da and dA, respectively (see Figure 2.1). A detailed derivation is given e.g. in [11].

Strains are caused by forces which are related to tractions and/or stresses. Important measures are the spatial traction vector $\mathbf{t}$:

$$\mathbf{t}(\mathbf{n}) = \boldsymbol{\sigma}\mathbf{n} \tag{2.9}$$

in which the Cauchy stress tensor $\boldsymbol{\sigma}$ describes the stress state w.r.t. the deformed configuration. Its counterpart, the material traction vector:

$$\mathbf{t}_0(\mathbf{N}) = \mathbf{P}\mathbf{N} \tag{2.10}$$

is defined by the first Piola-Kirchhoff stress tensor $\mathbf{P}$. For the scope of this thesis it is not suitable to use this stress measure due to its asymmetry[1] which is why the symmetric second Piola-Kirchhoff stress tensor $\mathbf{S}$ is utilized:

$$\mathbf{S} = \mathbf{F}^{-1}\mathbf{P} = J\,\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-T} \tag{2.11}$$

All forces acting on a differential volume must be in equilibrium. They are constituted of the rate of change of momentum within the considered volume and the sum of all surface and volume forces.

$$\partial_t \int\limits_{\Omega} \rho\dot{\mathbf{u}}\,\mathrm{d}\Omega = \oint\limits_{\Gamma} \mathbf{t}\,\mathrm{d}\mathbf{a} + \int\limits_{\Omega} \rho\mathbf{b}\mathrm{d}v \tag{2.12}$$

where the spatial quantities ρ, $\mathbf{b}$ and $\dot{\mathbf{u}}$ refer to the density, the mass related to the force of gravity and the velocity, respectively. This balance equation can be transformed to the material configuration by application of Equations (2.7) and (2.8) to give:

$$\partial_t \int\limits_{\Omega_0} \rho_0\dot{\mathbf{u}}\,\mathrm{d}V = \oint\limits_{\Gamma_0} \mathbf{t}\,\mathrm{d}\mathbf{A} + \int\limits_{\Omega_0} \rho_0\mathbf{b}\mathrm{d}V \tag{2.13}$$

From Equation (2.13) it can be deduced that density and tractions transform according to:

$$\rho_0 = \rho J \tag{2.14}$$
$$\mathbf{t}_0 = J\mathbf{F}^{-T}\mathbf{t} \tag{2.15}$$

Application of the Cauchy theorem 2.9, and the divergence theorem of Gauss to convert the boundary integral to a volume integral leads to the following fundamental Cauchy equations of motion in spatial and reference configuration.

$$\mathrm{div}\boldsymbol{\sigma} + \rho\mathbf{b} = \rho\ddot{\mathbf{u}} \tag{2.16}$$
$$\mathrm{Div}\mathbf{F}\mathbf{S} + \rho_0\mathbf{b} = \rho_0\ddot{\mathbf{u}} \tag{2.17}$$

In Equation (2.17) the second Piola-Kirchhoff stress tensor $\mathbf{S}$ is already incorporated and $\ddot{\mathbf{u}}$ refers to the acceleration of the considered particle. Neither Equation (2.17) nor Equation (2.16) are closed and must be supplemented with a material law. In this thesis only the St.-Vernant-Kirchhoff material is used[2]:

$$\mathbf{S} = \mathbb{C}\mathbf{E} \tag{2.18}$$

where $\mathbb{C}$ is a constant material tensor (see e.g. [11])
The weak form of Equation (2.17) is obtained via application of a standard Galerkin procedure involving the multiplication of Equation (2.17) with a suitable test function $\delta\mathbf{v} = \{\mathbf{v}|\mathbf{v} = 0\,\mathrm{on}\,\Gamma_D\}^3$, integration over the domain, partial integration of the emerging first term

[1] its basis is defined in both configurations as $\sum_{i,I=1}^{3} P_{iI}e_i \otimes E_I$

[2] which is, strictly spoken, only suitable for small strains. However, it does allow for large deflections. The combination of large deflections with small strains is, for example, present in the plate like structures dealt with in this thesis.

[3] the subscript $\cdot_D$ refers to a Dirichlet boundary. Additionally $\mathbf{v}$ needs to be a member of the energy space $\overset{\circ}{\mathbb{E}}$, i.e. functions whose derivative is square integrable, see e.g. [125]

and application of the Gaussian theorem. The procedure and its background is presented in detail e.g. in [110, 143]. The weak form reads:

$$\delta\mathfrak{W}(\mathbf{u},\mathbf{v}) = \underbrace{\int\limits_{\Omega_0} \mathbf{S} : \delta\mathbf{E}\, \mathrm{d}V}_{\text{(a)}} + \underbrace{\int\limits_{\Omega_0} \rho_0 \ddot{\mathbf{u}} \cdot \delta\mathbf{v}\, \mathrm{d}V}_{\text{(b)}} - \underbrace{\int\limits_{\Omega_0} \rho_0 \mathbf{b} \cdot \delta\mathbf{v}\, \mathrm{d}V}_{\text{(c)}} - \underbrace{\int\limits_{\Gamma_0} \mathbf{t}_0 \cdot \delta\mathbf{v}\, \mathrm{d}\mathbf{A}}_{\text{(d)}} = 0 \qquad (2.19)$$

This weak form is also termed virtual work, as the test functions $\delta\mathbf{v}$ can be interpreted as a virtual displacement which is indicated by the symbol δ. In this context, the individual terms represent the virtual work due to (a) the internal stress due to strains, (b) d'Alambert's acceleration force, (c) the body forces e.g. due to gravity and the (d) external forces due to tractions on the boundary. The abbreviation $\delta\mathbf{E}$ is the variation of the Green-Lagrange strain tensor:

$$\delta\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^T \mathrm{Grad}\delta\mathbf{v} + \mathrm{Grad}^T \delta\mathbf{v}\mathbf{F}\right) \qquad (2.20)$$

Supplemented with suitable boundary and initial conditions Equation (2.19) forms the basis of the solution to problems of computational continuum dynamics via the Finite Element Method. In general, it covers all types of nonlinearities, of which only geometrical nonlinearities are considered in this thesis. This manifests itself in the nonlinear strain-displacement relation of Equation (2.5).

In this thesis, a nonlinear solver based on the Newton-Raphson procedure is utilized. This procedure requires the linearization of Equation (2.19) via a Taylor series according to:

$$\mathfrak{W}(\bar{\mathbf{u}} + \Delta\mathbf{u}, \mathbf{v}) = \mathfrak{W}(\bar{\mathbf{u}}, \mathbf{v}) + \mathcal{D}\mathfrak{W}(\bar{\mathbf{u}}, \mathbf{v})[\Delta\mathbf{u}] = 0 \qquad (2.21)$$

It is carried out w.r.t. the displacements $\mathbf{u}$ at the linearization point $\bar{\mathbf{u}}$ in the direction $\Delta\mathbf{u}$. The directional derivative is denoted by $\mathcal{D}$. Only the nonlinearity of the first term in Equation (2.19) is considered and linearized w.r.t. $\mathbf{u}$[4]. This has been carried out in detail e.g. in [143], and the final form of the directional derivative is:

$$\mathcal{D}\mathfrak{W}(\bar{\mathbf{u}}, \mathbf{v})[\Delta\mathbf{u}] = \mathcal{D}\int\limits_{\Omega} \mathbf{S} : \delta\mathbf{E}\, \mathrm{d}V[\Delta\mathbf{u}] = \int\limits_{\Omega_0} \mathrm{Grad}\,\Delta\mathbf{u}\,\bar{\mathbf{S}} : \mathrm{Grad}\,\delta\mathbf{v}\mathrm{d}V + \int\limits_{\Omega_0} \delta\mathbf{E} : \mathbb{C} : \bar{\mathcal{D}}\mathbf{E}[\Delta\mathbf{u}]\mathrm{d}V \qquad (2.22)$$

The first term in Equation (2.22) is called the geometric component. It needs to be accounted for in the computation of the stiffness matrix when large deformations are to be considered

[4] In FSI with large displacements, the external forces $\mathbf{t}_0$ depend on the deformation and should therefore be linearized as done i.e. with follower loads. However, in the framework of a partitioned solution, it is more convenient to disregard the dependency on the load configuration in the linearization of Equation (2.19) and treat it incrementally in the overall search for a converged equilibrium position within the FSI cycles. Then, only the change of the area on which the force acts needs to be considered as the direction of the force emerges automatically. The area change can conveniently be respected via a direct integration of the loads in the deformed configuration as presented in Chapter 8.4.1.

The acceleration $\ddot{\mathbf{u}}$ can be dropped in the linearization as well since, at this continuous level, $\ddot{\mathbf{u}}$ is considered as a primary unknown variable and its nonlinear connection to $\mathbf{u}$ is treated at the discrete level in Chapter 2.2. Additionally, $\delta\mathbf{v}$ and $\mathbf{b}$ are independent of $\mathbf{u}$ and are dropped as well.

and vanishes for small displacements. The second component is the constitutive component and remains for small as well as large displacements. It contains the variation of the Green-Lagrangian strain tensor $\delta\mathbf{E}$ defined in 2.20 and its directional derivative is given by:

$$\bar{\mathcal{D}}\mathbf{E}[\Delta\mathbf{u}] = \frac{1}{2}\left(\bar{\mathbf{F}}^T\,\mathrm{Grad}\Delta\mathbf{u} + \mathrm{Grad}^T\Delta\mathbf{u}\bar{\mathbf{F}}\right) \tag{2.23}$$

For small displacements $\mathbf{F} \equiv \mathbf{I}$ such that Equation (2.23) transforms to the engineering strain tensor $\boldsymbol{\varepsilon} = \frac{1}{2}\left(\mathrm{Grad}^T\mathbf{u} + \mathrm{Grad}\,\mathbf{u}\right)$.

2.2 *p*-FEM discretization for structural dynamics

2.2.1 Preliminary discussion

Two and three-dimensional structures are investigated in this thesis for which elements of high orders are utilized. These have been implemented in the in-house *p*-FEM code AdhoC[31]. These elements are either based on a hexahedral or a quadrilateral element formulation utilizing the shape functions introduced in [125, 32, 126]. The following presentation of the discretization will be based on the three-dimensional formulation as the two-dimensional one can be obtained via cancellation of the terms associated with the third coordinate.

2.2.2 Spatial discretization

In principle, an incremental solution is sought to Equation (2.19) where at each step Equation (2.22) must be solved and the solution is updated accordingly. This results in the following basic procedure:

$$\begin{aligned} \text{solve} &: \quad \mathcal{D}\mathfrak{W}(\bar{\mathbf{u}},\mathbf{v})[\Delta\mathbf{u}] = -\mathfrak{W}(\bar{\mathbf{u}},\mathbf{v}) \\ \text{update} &: \quad \bar{\mathbf{u}} \leftarrow \bar{\mathbf{u}} + \Delta\mathbf{u} \end{aligned} \tag{2.24}$$

Equations (2.24) are iterated until the update of the linearization state $\Delta\mathbf{u}$ is small.
A Bubnov Galerkin discretization is utilized, where the only difference w.r.t. low order methods is that the degree of the Ansatz functions is arbitrarily high. The displacement field is approximated by $\mathbf{u} = \mathbf{N}\hat{\mathbf{u}}$ with $\mathbf{N}$ being the matrix of global shape functions:

$$\mathbf{N} := \begin{bmatrix} N_1(x,y,z) & 0 & 0 & N_2(x,y,z) & 0 & \dots & 0 \\ 0 & N_1(x,y,z) & 0 & 0 & N_2(x,y,z) & \dots & 0 \\ 0 & 0 & N_1(x,y,z) & 0 & 0 & \dots & N_G(x,y,z) \end{bmatrix} \tag{2.25}$$

To improve the condition number of the stiffness matrix, hierarchic shape functions instead of standard nodal based Lagrange shape functions are chosen[5]. The vector $\hat{\mathbf{u}}$ contains the

[5] This hierarchy can also be utilized for constructing error estimators, see [38] for example.

coefficients of the Ansatz functions $\mathbf{N}$ approximating the displacement field $\mathbf{u}$[6]. All computations aim at determining these coefficients and the displacement field is only recovered in a postprocessing step. The hat $\hat{}$ will thus be dropped to simplify notation in the sequel. Computations are, in general, carried out on the local element level and assembled into a global matrix as common practice in the FEM. This point is therefore disregarded as well for notational simplicity.

The 3D shape functions used in our implementation can be classified into four groups: the nodal or vertex modes, the edge modes, the face modes, and the internal modes. The nodal or vertex modes are defined by the standard trilinear shape functions, well known from the isoparametric eight-noded brick element. The edge and face modes are non-zero on the edges and faces which they are associated to and vanish on all other edges and faces, whereas the internal modes are purely local being zero on all faces and edges of the hexahedral element.

Three different types of trial spaces can be defined: the *trunk space* $\mathcal{S}_{\text{ts}}^{p_\xi,p_\eta,p_\zeta}(\Omega_{\text{st}}^{\text{h}})$, the *tensor product space* $\mathcal{S}_{\text{ps}}^{p_\xi,p_\eta,p_\zeta}(\Omega_{\text{st}}^{\text{h}})$ and an *anisotropic tensor product space* $\mathcal{S}^{p,p,q}(\Omega_{\text{st}}^{\text{h}})$. A detailed description of these trial spaces can be found in [125, 32, 126]. The polynomial degree for the trial spaces $\mathcal{S}_{\text{ts}}^{p_\xi,p_\eta,p_\zeta}(\Omega_{\text{st}}^{\text{h}})$ and $\mathcal{S}_{\text{ps}}^{p_\xi,p_\eta,p_\zeta}(\Omega_{\text{st}}^{\text{h}})$ can be varied separately in each local direction. It is possible to construct discretizations where the polynomial degree for the in-plane and thickness direction of thin-walled structures can be treated differently. High order solid elements can therefore provide a fully three-dimensional solution also including arbitrary three-dimensional stress states and, nevertheless, can cope with high aspect ratios of thin-walled structures. It was shown e.g. in [5, 4, 63] that these elements are less prone to locking effects than classical low order elements. Additionally, only one element type for thin- as well as thick-walled structures is sufficient. Transition elements between thin-walled and massive parts of the structure are not needed. A detailed discussion of the advantages of high order solid elements for thin-walled (nonlinear) continua can be found in [118, 125, 32, 37, 35, 34, 126, 108, 38].

In the process of solving Equation (2.24) a solution is sought for $\mathbf{u}$ (i.e. $\bar{\mathbf{u}}$ in every iteration) whose weak form is described by Equation (2.19). The spatial discretization of Equation (2.19) leads to the following semidiscrete form:

$$\mathbf{R}(\mathbf{u}) + \mathbf{M}\ddot{\mathbf{u}} = \mathbf{F} \tag{2.26}$$

in which the time derivative has yet to be discretized. The mass matrix emerges from term (b) in Equation (2.19) as:

$$\mathbf{M} = \rho_0 \int_{\Omega} \mathbf{N}^T \mathbf{N} \mathrm{d}\,\Omega \tag{2.27}$$

The body forces constituting term (c) of Equation (2.19) and the external forces (d) form the two parts of the load vector:

$$\mathbf{F} = \underbrace{\int_{\Omega} \mathbf{N}^T \rho_0 \mathbf{b} \mathrm{d}\,\boldsymbol{\Omega}}_{\text{(c)}} + \underbrace{\int_{\Gamma} \mathbf{N}^T \mathbf{t}_0 \mathrm{d}\,\mathbf{A}}_{\text{(d)}} \tag{2.28}$$

[6] As in the standard Galerkin approach, the virtual displacements are discretized with the same shape functions. Their (arbitrary) coefficients , say $\delta\hat{\mathbf{v}}$, drop out in the derivation and are not considered here from the start.

The internal forces stem from the internal stresses due to strains as given in term (a) in Equation (2.19) are symbolized by $\mathbf{R}(\mathbf{u})$ and discretized according to:

$$\mathbf{R}(\mathbf{u}) = \underbrace{\int\limits_{\Omega_0} \mathbf{B}^T \mathbf{S} \, \mathrm{d}V}_{\text{(a)}} \tag{2.29}$$

where the symmetric tensor $\mathbf{S}$ can be written as a vector using the notation of Voigt[143]:

$$\mathbf{S}^T := \begin{bmatrix} S_{11} & S_{22} & S_{33} & S_{12} & S_{23} & S_{31} \end{bmatrix} \tag{2.30}$$

The strain-displacement matrix $\mathbf{B}$ is defined as

$$\mathbf{B} := \begin{bmatrix} F_{11}\frac{\partial N}{\partial X} & F_{21}\frac{\partial N}{\partial X} & F_{31}\frac{\partial N}{\partial X} \\ F_{12}\frac{\partial N}{\partial Y} & F_{22}\frac{\partial N}{\partial Y} & F_{32}\frac{\partial N}{\partial Y} \\ F_{13}\frac{\partial N}{\partial Z} & F_{23}\frac{\partial N}{\partial Z} & F_{33}\frac{\partial N}{\partial Z} \\ F_{11}\frac{\partial N}{\partial Y} + F_{12}\frac{\partial N}{\partial X} & F_{21}\frac{\partial N}{\partial Y} + F_{22}\frac{\partial N}{\partial X} & F_{31}\frac{\partial N}{\partial Y} + F_{32}\frac{\partial N}{\partial X} \\ F_{12}\frac{\partial N}{\partial Z} + F_{13}\frac{\partial N}{\partial Y} & F_{22}\frac{\partial N}{\partial Z} + F_{23}\frac{\partial N}{\partial Y} & F_{32}\frac{\partial N}{\partial Z} + F_{33}\frac{\partial N}{\partial Y} \\ F_{13}\frac{\partial N}{\partial X} + F_{11}\frac{\partial N}{\partial Z} & F_{23}\frac{\partial N}{\partial X} + F_{21}\frac{\partial N}{\partial Z} & F_{33}\frac{\partial N}{\partial X} + F_{31}\frac{\partial N}{\partial Z} \end{bmatrix} \tag{2.31}$$

In Equation (2.31) F_{ij} refers to the entries of the deformation gradient $\mathbf{F}$ according to its definition in Equation (2.3).
The directional derivative $\mathcal{D}\mathfrak{W}(\bar{\mathbf{u}}, \mathbf{v})[\Delta\mathbf{u}]$ utilized in Equations (2.24) whose continuous form is given in Equation (2.22) leads to a tangential stiffness matrix $\mathbf{K}_T\Delta\mathbf{u}$ composed of the initial stress and the geometrical component. Its discrete form is:

$$\mathcal{D}\mathfrak{W}(\bar{\mathbf{u}}, \mathbf{v})[\Delta\mathbf{u}] \approx \mathbf{K}_T\Delta\mathbf{u} = \left(\int\limits_{\Omega} \mathbf{G}\mathbf{I}\mathrm{d}\Omega + \int\limits_{\Omega} \bar{\mathbf{B}}^T : \mathbb{C} : \bar{\mathbf{B}}\mathrm{d}V \right) [\Delta\mathbf{u}] \tag{2.32}$$

where $\mathbf{G}$

$$\mathbf{G} = (\mathrm{Grad}\mathbf{N})^T \, \bar{\mathbf{S}} \, \mathrm{Grad}\mathbf{N} \tag{2.33}$$

2.2.3 Temporal discretization

The spatial discretization of the weak form results in the semidiscrete form given in Equation (2.26) which has to be discretized in time. A wide variety of methods may be utilized of which [68] provides a good overview. All of these methods may directly be applied to high order spatial discretizations. Herefrom, the generalized-α method has been selected for implementation in AdhoC. It was first presented in [19] and is a generalization of the Newmark method [96]. Implementational details are given in [20, 62]. For the generalized-α method in context with energy conserving algorithms see e.g. [81].

To include damping, Equation (2.26) is extended by a damping matrix $\mathbf{C}$ which is formed by a linear combination of the mass and the stiffness matrix according to $\mathbf{C} = a\mathbf{K} + b\mathbf{M}$. This type of damping is also termed Rayleigh damping[7].

The extension of Equation (2.26) with a damping term $\mathbf{C}\dot{\mathbf{u}}$ leads to the additional unknown velocity $\dot{\mathbf{u}}$. In the Newmark method, the velocity $\dot{\mathbf{u}}$ and acceleration $\ddot{\mathbf{u}}$ are approximated via a finite difference Ansatz in terms of the primary unknown variable, the displacement $\mathbf{u}$. The supporting points of this finite difference scheme are the values computed at the end of the previous time steps. The generalized-α method extends this principle. Here, the support points of the finite difference scheme are themselves linear interpolations between the values computed at the beginning and the end of the previous time steps. The semidiscrete form of the generalized-α method is thus given by:

$$\mathbf{M}\ddot{\mathbf{u}}_{n+1-\alpha_m} + \mathbf{C}\dot{\mathbf{u}}_{n+1-\alpha_f} + \mathbf{R}(\mathbf{u})_{n+1-\alpha_f} = \mathbf{F}_{t_{n+1-\alpha_f}} \tag{2.34}$$

where α_f and α_m are scalars. Essentially, acceleration terms are evaluated at a point in time linearly interpolated from $t_{n+1-\alpha_m} = (1-\alpha_m)t_{n+1} + \alpha_m t_n$ while velocities as well as internal and external forces are evaluated at the point $t_{n+1-\alpha_f} = (1-\alpha_f)t_{n+1} + \alpha_f t_n$. This interpolation offers the possibility of numerical damping while keeping the algorithm second order accurate w.r.t. the time step size. The Ansatz to reduce the velocity and acceleration terms to the primary unknown variable $\mathbf{u}$ reads:

$$\begin{aligned} \dot{\mathbf{u}}_{n+1-\alpha_f} &= \tfrac{1-\alpha_m}{\beta\Delta t^2}\left(\mathbf{u}_{n+1} - \mathbf{u}_n\right) - \tfrac{1-\alpha_m}{\beta\Delta t}\dot{\mathbf{u}}_n - \tfrac{1-\alpha_m-2\beta}{2\beta}\ddot{\mathbf{u}}_n \\ \ddot{\mathbf{u}}_{n+1-\alpha_m} &= \tfrac{(1-\alpha_f)\gamma}{\beta\Delta t}\left(\mathbf{u}_{n+1} - \mathbf{u}_n\right) - \tfrac{(1-\alpha_f)\gamma-\beta}{\beta}\dot{\mathbf{u}}_n - \tfrac{(\gamma-2\beta)(1-\alpha_f)}{2\beta}\Delta t\ddot{\mathbf{u}}_n \end{aligned} \tag{2.35}$$

where β and γ are scalars determining the properties of the scheme. They are defined in Equations (2.42) and (2.43). Insertion of Equations (2.35) into Equation (2.34) leads to the following effective structural equation system for the determination of $\mathbf{u}_{n+1}$:

$$\begin{aligned} \mathbf{G}\left(\mathbf{u}_{n+1}\right) &= \mathbf{M}\left[\frac{1-\alpha_m}{\beta\Delta t^2}\left(\mathbf{u}_{n+1} - \mathbf{u}_n\right) - \frac{1-\alpha_m}{\beta\Delta t}\dot{\mathbf{u}}_n - \frac{1-\alpha_m-2\beta}{2\beta}\ddot{\mathbf{u}}_n\right] \\ &+ \mathbf{C}\left[\frac{(1-\alpha_f)\gamma}{\beta\Delta t}\left(\mathbf{u}_{n+1} - \mathbf{u}_n\right) - \frac{(1-\alpha_f)\gamma-\beta}{\beta}\dot{\mathbf{u}}_n - \frac{(\gamma-2\beta)(1-\alpha_f)}{2\beta}\Delta t\ddot{\mathbf{u}}_n\right] \\ &+ \mathbf{R}\left(\mathbf{u}_{n+1-\alpha_f}\right) - \mathbf{F}_{n+1-\alpha_f} = \mathbf{0} \end{aligned} \tag{2.36}$$

In addition to boundary conditions defined for the static problem, Equation (2.36) requires the definition of velocities and displacements at time step t_0 as initial conditions.

Equation (2.36) is nonlinear in $\mathbf{u}_{n+1}$ and must be linearized to be accessible to a Newton-Raphson scheme. The linearization leads to the dynamic stiffness matrix $\mathbf{K}_{T_{dyn}}$:

$$\mathbf{K}_{T_{dyn}} = (1-\alpha_f)\left[\mathbf{K}_{T_{t+1}} + \frac{\gamma}{\beta\Delta t}\mathbf{C}\right] + \frac{1-\alpha_m}{\beta\Delta t^2}\mathbf{M} \tag{2.37}$$

[7] It offers computational simplicity, however describes a damping behaviour which is more suited for numerical reasons than for describing real physical damping behaviour. See e.g. [104]

where the tangential (static) stiffness matrix $\mathbf{K}_{T_{t+1}}$ is computed via Equation (2.32). The discrete Newton-Raphson procedure is given by:

$$\mathbf{K}^{i}_{T_{dyn}} \Delta \mathbf{u}^{i+1}_{n+1} = -\mathbf{G}\left(\mathbf{u}_{n+1}\right) \tag{2.38}$$

$$\mathbf{u}^{i+1}_{n+1} = \mathbf{u}^{i+1}_{n+1} + \Delta \mathbf{u}^{i+1}_{n+1} \tag{2.39}$$

For geometric linear computations, unconditional stability and second order accuracy with maximal algorithmic damping for high frequencies is obtained by (see [19]):

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1} \tag{2.40}$$

$$\alpha_f = \frac{\rho_\infty}{\rho_\infty + 1} \tag{2.41}$$

$$\beta = \frac{1}{4}\left(1 - \alpha_m + \alpha_f\right)^2 \tag{2.42}$$

$$\gamma = \frac{1}{2} - \alpha_m + \alpha_f \tag{2.43}$$

where ρ_∞ can be chosen to either maximize or eliminate the damping of frequencies that are high relative to the resolution level. If ρ_∞ is chosen to be zero the method is said to annihilate the highest frequency in one step (only for a linear problem). If ρ_∞ is chosen to be one then the highest frequency (as well as all other others) are preserved (in the linear problem). The method then corresponds to the midpoint rule which is equivalent to the trapezoidal rule for linear problems [71]. The second order accurate Newmark scheme with no numerical damping is obtained by setting $\alpha_m = \alpha_f = 0$[8]. For geometrical nonlinear computations, the Newmark method and the generalized alpha method both loose their unconditional stability. In the Newmark method, conditional stability may be controlled by introducing stronger damping of higher modes via selection of β according to:

$$\beta = \frac{\left(\gamma + \frac{1}{2}\right)^2}{4} \tag{2.44}$$

where $\frac{1}{2} \leq \gamma < 1$. An increasing γ increases the damping of the higher modes and explains the choice for the parameters $\gamma = 0.9$ and $\beta = 0.49$ in Chapter 8. The trade off is the loss of second order accuracy. A more profound discussion of these parameters can be found in [67]. Alternatively, a stiffness proportional damping is used to damp out the frequencies which are higher than those which can be resolved by the time step itself. This may be achieved by the Newmark Method with pure stiffness proportional damping i.e. $\mathbf{C} = a\mathbf{K}$. In this case, the modal damping ξ_n is given by:

$$\xi_n = \frac{1}{2} a\omega_n = \frac{\pi}{T_n} a \tag{2.45}$$

[8] Other schemes such as the Hilber-α Method [64] emerge via setting $\alpha_m = 0$ and the Bossak-α Method suggested in [142] is obtained by setting $\alpha_f = 0$

For a time step length Δt equal to the time duration of a period T_n a modal damping with a value of at least one for the specific frequency ω associated with T_n can be assigned by choosing a according to:

$$a = \xi_n \frac{\Delta t}{\pi} \tag{2.46}$$

The damping of the other frequencies is then given by the relationship in Equation (2.45). This type of damping was used e.g. in Chapter 7.2.1.

2.2.4 Remarks on p-FEM for fluid-structure interaction

One advantage of utilizing three-dimensional models for fluid-structure interaction problems is that a correct representation of the *wet surface* in contact with the fluid emerges directly from the formulation without having to reconstruct this skin surface from the middle surface and certain kinematic assumptions, as in the usual approach for dimensionally reduced plate and shell models. This is an issue which has long been recognized in FSI but still is described as unresolved (see e.g. [88]). It may be avoided altogether elegantly by using high order, spatially adapted discretizations as suggested in this thesis. Curved geometries are described by means of the blending function method. This method also works very well for three-dimensionally curved surfaces. A detailed description is given in [32, 126]. Moreover, by using an adaptive selection of the polynomial degree in the three directions, a very cost-efficient discretization can be found while maintaining control of the approximation and modelling errors. See [38, 90] for elastostatic problems and [117, 114, 116] for elastodynamic problems as further references.

Chapter 3

Computational fluid mechanics

3.1 Basic mechanics for incompressible fluids

Incompressible fluids are characterized by the incompressible form of the Navier-Stokes equations. They provide the basis for the formulation of the Finite Element Method (FEM) and the Finite Volume Method (FVM) alike. They can be written in conservative or non-conservative form. Integral and weak forms, index and vector notation contribute to the variations in their appearance. The Navier-Stokes equations have been derived in numerous textbooks in all forms and this shall not be repeated here. Instead, the appropriate formulation will be chosen from literature as a starting point to describe the basic layout of the discretization method. However, to introduce some important notions and concepts, a general layout of their derivation is attempted as given in more detail e.g. in [65].

The behaviour of an incompressible, isothermal Newtonian fluid is well described considering the conservation of mass and momentum in an arbitrary control volume as depicted in Figure 3.1.

U is a given scalar or vector valued quantity and F represents the flux of that quantity through the surface S enclosing a volume Ω. Within Ω, one may have sources or sinks of the quantity to be preserved either on the surface $\mathbf{Q}_s$ or within the domain $\mathbf{Q}_v$. The domain Ω can depend on time but is considered fixed for now. The conservation equation balancing gains and losses of U reads:

$$\partial_t \int\limits_{\Omega} \mathbf{U}\,\mathrm{d}\Omega = -\oint\limits_{S} \mathbf{F}\cdot \mathrm{d}\mathbf{S} + \int\limits_{\Omega} \mathbf{Q}_v\,\mathrm{d}\Omega + \oint\limits_{S} \mathbf{Q}_s \cdot \mathrm{d}\mathbf{S} \tag{3.1}$$

If the conservation of mass is considered, then $\mathbf{U}$ is equal to the scalar function ρ describing the variation of density (mass per volume) in space. The flux $\mathbf{F}$ of the mass ρ through the surface element $d\mathbf{S}$ is taken into account by the velocity of the mass $\mathbf{v}$ times the mass itself. Only the perpendicular portion of this flux through the surface is of importance. Neglecting mass sources in the interior and diffusion of mass through the surface Equation (3.1) transforms to:

$$\partial_t \int\limits_{\Omega} \rho\,\mathrm{d}\Omega + \oint\limits_{S} \rho\mathbf{v}\cdot \mathrm{d}\mathbf{S} = 0 \tag{3.2}$$

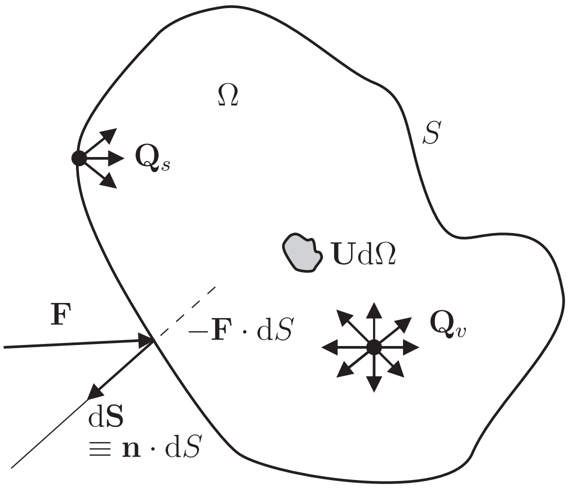

Ω
S
$\mathbf{Q}_s$
$\mathbf{U}\mathrm{d}\Omega$
$\mathbf{F}$
$-\mathbf{F}\cdot\mathrm{d}S$
$\mathbf{Q}_v$
$\mathrm{d}\mathbf{S}$
$\equiv \mathbf{n}\cdot\mathrm{d}S$

Figure 3.1: Conservation of quantity **U** at a finite fixed but arbitrarily shaped volume

Application of the divergence theorem of Gauss to Equation (3.2) transforms the surface integral to a volume integral of the divergence of the considered fluxes. The mass conservation then reads:

$$\partial_t \int\limits_\Omega \rho\,\mathrm{d}\Omega + \int\limits_\Omega \nabla\cdot(\rho\mathbf{v})\,\mathrm{d}\Omega = 0 \tag{3.3}$$

In incompressible fluids, the density is constant in time as well as in space such that the mass conservation simplifies to

$$\int\limits_\Omega \nabla\cdot\mathbf{v}\,\mathrm{d}\Omega = 0 \tag{3.4}$$

in integral form. As Equation (3.4) is valid for arbitrary volumes, including infinitesimally small ones, the integral may be dropped to give Equation (3.4) in differential form as:

$$\nabla\cdot\mathbf{v} = 0 \tag{3.5}$$

The conservation of momentum $\rho\mathbf{v}$ is accounted for by Equation (3.1) if $\mathbf{U} \equiv \rho\mathbf{v}$. The convective flux tensor is then given by $\rho\mathbf{v}\otimes\mathbf{v}$ where $\otimes$ symbolizes the dyadic product. The conservation of momentum stated in the integral form then reads:

$$\partial_t \int\limits_\Omega \rho\mathbf{v}\,\mathrm{d}\Omega + \oint\limits_S \rho\mathbf{v}\,(\mathbf{v}\cdot\mathrm{d}\mathbf{S}) = \int\limits_\Omega \rho\mathbf{F}_e\,\mathrm{d}\Omega + \oint\limits_S \boldsymbol{\tau}\cdot\mathrm{d}\mathbf{S} \tag{3.6}$$

The stresses inside the fluid are denoted by $\boldsymbol{\tau}$. Inside the control volume, they cancel each other and appear as stresses acting on the surface with an intensity of $\boldsymbol{\tau}\cdot\mathrm{d}S$. In a Newtonian fluid these stresses are split up into pressure and shear stresses:

$$\boldsymbol{\tau} = -p\mathbf{I} + \boldsymbol{\sigma} \tag{3.7}$$

where $\boldsymbol{\sigma}$ contains the constitutive relation. Inserting Equation (3.7) into Equation (3.6) gives:

$$\partial_t \int\limits_{\Omega} \rho \mathbf{v} \, \mathrm{d}\Omega + \oint\limits_{S} \rho \mathbf{v} \left(\mathbf{v} \cdot \mathrm{d}S \right) = \int\limits_{\Omega} \rho \mathbf{F}_e \, \mathrm{d}\Omega - \oint\limits_{S} p \mathbf{I} \cdot \mathrm{d}\mathbf{S} + \oint\limits_{S} \boldsymbol{\sigma} \cdot \mathrm{d}\mathbf{S} \tag{3.8}$$

Application of the Gauss Theorem[1] and neglecting volume forces leads to the integral form of the momentum conservation for a Newtonian fluid:

$$\partial_t \int\limits_{\Omega} \rho \mathbf{v} \, \mathrm{d}\Omega + \int\limits_{\Omega} \nabla \cdot \left(\rho \mathbf{v} \otimes \mathbf{v} \right) \mathrm{d}\Omega + \int\limits_{\Omega} \nabla \cdot \left(p \mathbf{I} - \boldsymbol{\sigma} \right) \, \mathrm{d}\Omega = \mathbf{0} \tag{3.9}$$

Equation (3.9) is valid for all volumes and the integration may be dropped to give the momentum conservation in differential form as:

$$\partial_t \rho \mathbf{v} + \nabla \cdot \left(\rho \mathbf{v} \otimes \mathbf{v} \right) + \nabla \cdot \left(p \mathbf{I} - \boldsymbol{\sigma} \right) = \mathbf{0} \tag{3.10}$$

The second and third term of Equation (3.10) can be written as:

$$\nabla \cdot \mathbf{\Pi} = \nabla \cdot \left(\rho \mathbf{v} \otimes \mathbf{v} + p \mathbf{I} - \boldsymbol{\sigma} \right) \tag{3.11}$$

where $\mathbf{\Pi}$ is termed momentum flux tensor, because it describes the transport of the momentum. Equations (3.4) and Equation (3.9) are termed Navier-Stokes equations for incompressible, Newtonian fluids in integral form and vector notation. The differential form is restated next for clarity:

$$\rho \partial_t \left(\mathbf{v} \right) + \nabla \cdot \left(\mathbf{\Pi} \right) = \mathbf{0} \tag{3.12}$$

$$\nabla \cdot \mathbf{v} = 0 \tag{3.13}$$

It is sometimes more illustrative to use index notation with the summation convention of Einstein. The Navier-Stokes equations read:

$$\rho v_{i,t} + \Pi_{ij,i} = \mathbf{0} \tag{3.14}$$

$$v_{i,i} = 0 \tag{3.15}$$

The momentum flux tensor Π_{ij} in index notation reads:

$$\Pi_{ij} = \rho v_j v_i + p \delta_{ij} - \mu \left(v_{i,j} + v_{j,i} \right) \tag{3.16}$$

The constitutive relations are given by:

[1] and thereby assuming differentiable fluxes

$$
\begin{aligned}
\tau_{ij} &= p\delta_{ij} - \mu\left(v_{i,j} + v_{j,i}\right) && (3.17)\\
p_i &= p\delta_{ij} && (3.18)\\
\sigma_{ij} &= \mu\left(v_{i,j} + v_{j,i}\right) && (3.19)
\end{aligned}
$$

Assuming that the density $\rho = 1$, introducing the kinematic viscosity:

$$\nu = \frac{\mu}{\rho} \tag{3.20}$$

and respecting the mass conservation described in Equation (3.5), Equation (3.14) may as well be written as:

$$v_{i,t} + v_j v_{i,j} = -(p\delta_{ij})_{,j} + \nu v_{i,jj} \tag{3.21}$$

Further, the Reynolds number has to be mentioned. It is a dimensionless number measuring the ratio between internal forces stemming from inertia (ρv) to viscous forces ($\frac{\mu}{L}$). Using Equation (3.20), the Reynolds number Re can be written as:

$$Re = \frac{Lv}{\nu} \tag{3.22}$$

where L is a characteristic length. This ratio is a decisive measure for characterising flow behaviour. Among other purposes, it is used to determine dynamic similitude between different numerical or experimental setups.

3.2 Discretization by the Finite Volume Method

Extensive literature exists on Finite Volume Methods (FVM) for fluid flows (see e.g. [65] and the literature cited therein). Implementational issues for FVM on deforming grids are nicely laid out e.g. in [121].

In this research work, the solver utilized to compute flows with the Finite Volume Method was the commercial solver CFX [2]. Unlike in the other two solvers, no changes were made to the kernel of the solver such that the formulation used is identical to the one given in the solver manual [2]. Only the basis of the method is presented here for completeness.

Starting point are the conservation equations for mass and momentum in their integral form as given in Equation (3.4) and Equation (3.9) respectively. In order to incorporate deflecting boundaries, CFX utilizes the Arbitrary Eulerian-Lagrangian (ALE) formulation of these conservation laws. The ALE formulation of Equation (3.4) and Equation (3.9) needs to account for an extra source term stemming from the deformation of Ω with respect to time $\frac{\partial \Omega}{\partial t}$.

A derivation may be found, e.g. in [2] and the final form restated here for completeness reads:

$$
\begin{aligned}
\partial_t \int\limits_{\Omega(t)} \rho \, \mathrm{d}\Omega + \oint\limits_{S(t)} \rho(\mathbf{v} - \mathbf{v}_g) \cdot \mathrm{d}\mathbf{S} &= 0 && (3.23)\\
\partial_t \int\limits_{\Omega(t)} \rho\mathbf{v} \, \mathrm{d}\Omega + \oint\limits_{S(t)} \rho\mathbf{v}\left((\mathbf{v} - \mathbf{v}_g) \cdot \mathrm{d}\mathbf{S}\right) &= \int\limits_{\Omega(t)} \rho\mathbf{F}_e \, \mathrm{d}\Omega + \oint\limits_{S(t)} \boldsymbol{\tau} \cdot \mathrm{d}\mathbf{S} && (3.24)
\end{aligned}
$$

where $\mathbf{v}_g$ is the velocity of the boundary of the control volume.
Additionally, the geometric conservation law (see e.g. [24, 25, 41]) needs to be obeyed. It prevents the appearance of artificial mass in a moving domain. In a fluid at rest with a constant density and the absence of sources and sinks, the change of mass in the volume must only stem from the change of volume[2]:

$$\rho \partial_t \int\limits_{\Omega(t)} \mathrm{d}\Omega + \rho \oint\limits_{S(t)} \mathbf{v}_g \cdot \mathrm{d}\mathbf{S} = 0 \tag{3.25}$$

The basic idea in the FVM is to directly discretize the integral form of the conservation laws expressed in Equations (3.23), (3.24) and (3.25) at finite control volumes (cells). The discrete form of the conservation laws is obtained by replacing the volume integrals with an averaged value over the considered cell and the surface integrals by the sum over all bounding faces of the considered volume.
CFX solves the pressure velocity coupling not by a pressure correction algorithm but by a coupled approach in which at any given timestep, the velocity and pressures are determined simultaneously.
The mesh deformation is solved at each time step by a Poisson equation which diffuses specified displacements homogenously throughout the mesh. A mesh stiffness proportional to the volume may be provided to improve performance w.r.t. overlapping meshes.

3.3 Discretization by the Spectral Element Method

The differential form of the Navier-Stokes Equations (3.21) and (3.13) are written in a fixed (Eulerian) reference frame. The state of the art concerning their spectral discretization was summarized in the books by Canuto [16] and further treated extensively by Karniadakis in [73].
Equations (3.21) and (3.13) serve as a starting point in deriving their weak form in an arbitrarily moving reference frame (ALE-form). The procedure of how to derive their weak form in an ALE reference frame was presented e.g. in [66, 99] and is not restated here but in turn serves as a starting point for the following description of the Spectral Element Method (SEM) in ALE-form:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \int\limits_{\Omega(t)} v_i \delta v_i \, \mathrm{d}\Omega + \int\limits_{\Omega(t)} \left[(v_j - w_j) v_{i,j} \delta v_i - v_i w_{j,j} \delta v_i \right] \mathrm{d}\Omega + & \\ \int\limits_{\Omega(t)} \left[\left(-p\delta_{ij} + \nu v_{i,j} \right) \delta v_{i,j} - f_i \delta v_i \right] \mathrm{d}\Omega &= 0 \qquad (3.26) \\ \int\limits_{\Omega(t)} v_{j,j} \, \delta q \, \mathrm{d}\Omega &= 0. \qquad (3.27) \end{aligned}$$

where it is assumed for simplicity that the fluid density $\rho = 1$. The kinematic viscosity is defined according to Equation (3.20). Fluid and mesh velocity are represented by v_i and w_i,

[2] It is interesting to note that obeying the geometric conservation law formally leads to a time independent mass conservation. Insertion of Equation (3.25) into Equation (3.24) leads to $\oint_S \mathbf{v} \cdot \mathrm{d}S = 0$.

internal fluid forces are denoted by f_i and pressures by p, where δ_{ij} is the Kroneker delta. δ denotes, that the following letter symbolizes a suitable test function. Equation (3.26) is transferred to the Eulerian or Lagrangian form by setting $w_i = 0$ or $w_i = v_i$, respectively. A straight forward Galerkin discretization of Equations (3.26) and (3.27) in which the velocity, pressure and mesh unknowns are expressed as $v_i = \mathbf{\Phi}\mathbf{v}_i$, $p = \mathbf{\Phi}\mathbf{p}$, $w_i = \mathbf{\Phi}\mathbf{w}_i$ and $x_i = \mathbf{\Phi}\mathbf{x}_i$, with $\mathbf{\Phi}$ containing the global shape functions $\mathbf{\Phi}$:

$$\mathbf{\Phi} := \left[\begin{array}{cccc} \phi_1(x,y) & \phi_2(x,y) & \ldots & \phi_G(x,y) \end{array} \right] \tag{3.28}$$

leads to:

$$\begin{aligned} \mathbf{M}\mathbf{v}_{i,t} + \mathbf{N}_i(\mathbf{v},\mathbf{w}) &= \mathbf{D}_i^T\mathbf{p} - \nu\mathbf{L}\mathbf{v}_i + \mathbf{F}_i && (3.29) \\ \mathbf{D}_i\mathbf{v}_i &= 0, && (3.30) \end{aligned}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{D}_i$ is the derivative matrix, $\mathbf{L}$ is the stiffness (Laplacian) matrix, and $\mathbf{F}_i$ is the vector of body forces. They are computed by:

$$\begin{aligned} \mathbf{M} &= \int\limits_\Omega \mathbf{\Phi}^T\mathbf{\Phi}\,\mathrm{d}\Omega && (3.31) \\ \mathbf{D}_i &= \int\limits_\Omega \mathbf{\Phi}^T\left(\nabla_i\mathbf{\Phi}\right)\,\mathrm{d}\Omega && (3.32) \\ \mathbf{L} &= \int\limits_\Omega \left(\nabla\mathbf{\Phi}\right)^T\left(\nabla\mathbf{\Phi}\right)\,\mathrm{d}\Omega && (3.33) \\ \mathbf{F}_i &= \int\limits_\Omega f_i\mathbf{\Phi}^T\,\mathrm{d}\Omega, && (3.34) \end{aligned}$$

where ∇ is the Nabla operator. The nonlinear convective term is denoted by $\mathbf{N}_i(\mathbf{v},\mathbf{w})$ to describe the dependency on both the fluid and mesh velocity. The mesh velocity is defined based on the mesh coordinates from

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{w}_i. \tag{3.35}$$

Again, the main difference compared to low order methods is the choice of the shape functions contained in $\mathbf{\Phi}$. They are chosen to improve the condition of the mass matrix and are based on Jacobi polynomials. For a detailed description of the corresponding suitable polynomial expansion basis, the reader is referred to [73].

The spectral Finite Element solver $N\epsilon\kappa\tau\alpha\rho$, described in [73] is freely available for download (including sources) at [1] and incorporates this discretization method. The time integration algorithm implemented in $N\epsilon\kappa\tau\alpha\rho$ is the high order stiffly stable time integration scheme discussed in [73, 72] where comments on the stability of such schemes may be found as well. An explicit version was further extended in [7] for the ALE-formulation. It is a high order *explicit* splitting scheme in which the nonlinear term, mesh velocity term and pressure boundary conditions are first treated explicitly by:

$$\frac{1}{\Delta t}\left(\tilde{\mathbf{v}}_i - \sum_{q=0}^{J_i-1}\alpha_q \mathbf{M}\mathbf{v}_i^{n-q}\right) = -\sum_{q=0}^{J_e-1}\beta_q \mathbf{N}(\mathbf{v}^{n-q},\mathbf{w}^{n-q}) + \mathbf{F}^{n+1}, \tag{3.36}$$

$$\frac{1}{\Delta t}\left(\gamma_0 \mathbf{x}_i^{n+1} - \sum_{q=0}^{J_x-1}\alpha_q \mathbf{x}_i^{n-q}\right) = \sum_{q=0}^{J_w-1}\beta_q \mathbf{w}_i^{n-q}, \tag{3.37}$$

$$\frac{\partial p^{n+1}}{\partial n} = -\mathbf{n}\cdot \left[\frac{\partial \mathbf{v}^{n+1}}{\partial t} + \sum_{q=0}^{J_e-1}\beta_q \mathbf{N}(\mathbf{v}^{n-q},\mathbf{w}^{n-q}) + \nu\sum_{q=0}^{J_e-1}\beta_q[\nabla\times(\nabla\times\mathbf{v}^{n-q})]\right] \tag{3.38}$$

The timestep is denoted by n. Equation (3.36) determines the non-divergence free velocity $\tilde{\mathbf{v}}_i$ from the divergence free velocities $\mathbf{v}_i$ of the previous time steps, the external forces and the nonlinear convective term. The orders of integration for the advection and the diffusion terms are denoted as J_e and J_i, respectively. Then, the new mesh position $\mathbf{x}_i$ at time step $n+1$ is computed via Equation (3.37) where J_x and J_w denote the orders of integration for the material position and the mesh velocity, respectively. All integration weights, including α, β and γ, are dependent on the order of time integration and may be found in [73] in Table 5.2. Equation (3.37) then computes the pressure p^{n+1} normal to the boundary. Next, the implicit terms are treated as follows:

$$\mathbf{L}\mathbf{p}^{n+1} = \frac{1}{\Delta t}\mathbf{D}_i\tilde{\mathbf{v}}_i, \tag{3.39}$$

$$\frac{1}{\Delta t}\left(\gamma_0\mathbf{M}\mathbf{v}_i^{n+1} - \tilde{\tilde{\mathbf{v}}}_i\right) = \nu\mathbf{L}\mathbf{v}_i^{n+1}, \tag{3.40}$$

$$\mathbf{L}\mathbf{w}_i^{n+1} = 0 \tag{3.41}$$

where the intermediate field $\tilde{\tilde{\mathbf{v}}}_i$ is computed by:

$$\tilde{\tilde{\mathbf{v}}}_i = \mathbf{M}\tilde{\mathbf{v}}_i + \mathbf{D}_i^T\mathbf{p}^{n+1}. \tag{3.42}$$

The pressure is computed in Equation (3.39) and the divergence free velocity is obtained via the viscous correction in Equation (3.40). The mesh velocity is obtained here from the Laplacian Equation (3.41). In principle, its choice may be arbitrary which is why it can be adjusted to minimize mesh deformation in an effort to avoid overlapping elements for large deformations.

In order to achieve a stable implicit coupling procedure for FSI, the explicit stiffly stable time integration scheme needed to be changed to an implicit version. The corresponding *implicit* stiffly stable time integration scheme was derived by modifying the right-hand side of Equations (3.36)-(3.38) as follows:

$$\frac{1}{\Delta t}\left(\tilde{\mathbf{v}}_i - \sum_{q=0}^{J_i-1}\alpha_q \mathbf{M}\mathbf{v}_i^{n-q}\right) = -\mathbf{N}(\mathbf{v}^{n+1},\mathbf{w}^{n+1}) + \mathbf{F}^{n+1}, \tag{3.43}$$

$$\frac{1}{\Delta t}\left(\gamma_0 \mathbf{x}_i^{n+1} - \sum_{q=0}^{J_x-1}\alpha_q \mathbf{x}_i^{n-q}\right) = \mathbf{w}_i^{n+1}. \tag{3.44}$$

$$\frac{\partial p^{n+1}}{\partial n} = -\mathbf{n}\cdot\left[\frac{\partial \mathbf{v}^{n+1}}{\partial t} + \mathbf{N}(\mathbf{v}^{n+1},\mathbf{w}^{n+1}) + \nu[\nabla\times(\nabla\times\mathbf{v}^{n+1})]\right] \tag{3.45}$$

Effectively, the right-hand side now only considers terms from the current timestep $n+1$. An iteration over Equations (3.43)-(3.45) and Equations (3.39)-(3.41) is now needed for the implicit solution of pressure and velocities. Beyond the necessity to have an implicit scheme for an overall implicit coupling procedure for FSI, the presented implicit scheme offers a (theoretical) increase of the stable region for an increase of the order of integration, as opposed to the explicit integration scheme. This has, however, not been validated for implementational reasons.

The stresses at the boundary are computed by Equation (3.17), which requires evaluating the derivatives of the velocity field.

3.4 The Lattice Boltzmann Method

All methods presented previously are based on differential equations which were derived from macroscopic continuum theory. Not so the Lattice Boltzmann Method (LBM). It is based on the Boltzmann equation and emerged from molecular gas dynamics, investigating collision and propagation of particles. The Lattice Boltzmann Method has been presented and reviewed numerous times in literature (see e.g. [18, 59, 78]). The implementation of the method on dynamically adaptive grids is described in [130]. All computations with LBM were carried out in close cooperation with Sebastian Geller from IRMB [49]. The following is an attempt to introduce the most important concepts and to put the LBM into context with the Finite Volume Method introduced in Chapter 3.2 and the Spectral Element Method introduced in Chapter 3.3.

An important measure in molecular gas dynamics is the Knudsen number:

$$Kn = \frac{l_f}{L} \tag{3.46}$$

where l_f is the mean free path of a particle flying in space without colliding with others and L is a characteristic, macroscopic length over which macroscopic variations of the flow situation can be noticed. Two extreme values of Kn classify gases. The presence of a continuum[3] may be assumed if $Kn \ll 1$. This assumption is violated at $Kn \geq 1$ classifying free molecular streams[4]. The Knudsen number is inversely proportional to the density of the gas.

In the sequel it is assumed that, at microlevel, molecules travel with an absolute speed $\boldsymbol{\xi}$ which may be decomposed according to:

$$\boldsymbol{\xi} = \boldsymbol{c} + \mathbf{v} \tag{3.47}$$

where $\boldsymbol{c}$ is a peculiar speed stemming from the motion of particles due to temperature and describes the speed of the particles relative to the fluid speed $\mathbf{v}$.

Another important definition is that of the Mach number. It describes the relation between the microscopic speed c to the macroscopic flow velocity U_0, in the form:

$$Ma = \frac{U_0}{c} \tag{3.48}$$

[3] e.g. air at atmospheric pressure

[4] e.g. satelites reentering the atmosphere

3.4.1 The Boltzmann equation and the Maxwell distribution

In vector notation and without external forces, the Boltzmann equation reads:

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial f}{\partial \mathbf{x}} = \Omega\left(f\right) \tag{3.49}$$

where $f(t, \mathbf{x}, \boldsymbol{\xi})$ is the unknown particle distribution function. The particle distribution function is defined as:

$$f\left(t, x, y, z, \xi_x, \xi_y, \xi_z\right) = \frac{dN}{dx\, dy\, dz\, d\xi_x\, d\xi_y\, d\xi_z} \tag{3.50}$$

The distribution function $f(t, \mathbf{x}, \boldsymbol{\xi})$ can be viewed deterministically as a density function describing the absolute number of particles N present in a volume of space and velocity. It can also be viewed probabilistically describing the probability to find a molecule at a specific time instance t at a specific location $\mathbf{x}$ with a specific velocity $\boldsymbol{\xi}$. Equation (3.49) therefore has seven dimensions.

The left hand side is the total derivative of the particle distribution function f w.r.t. time t and space $\mathbf{x}$. It is thus a linear advection equation, whose whole purpose is to transport the unknown "concentration" or "mass fractions" f along characteristic lines with a constant speed $\boldsymbol{\xi} = \partial_t \mathbf{x}$ without changing their value. The nonlinearity is hidden in the collision operator Ω on the right-hand side:

$$\Omega\left(f\right) = \frac{1}{m} \int U\left(f(\boldsymbol{\xi}')\right)\left(f(\boldsymbol{\xi}_1') - f(\boldsymbol{\xi})\right)\left(f(\boldsymbol{\xi_1})\right) \mathrm{d}\overline{\varpi\boldsymbol{\xi_1}} \tag{3.51}$$

It models the collision of the molecules (or mass fractions), whereby $'$ denotes their state after the collision and the index $_1$ is used to distinguish the two different mass fractions which are colliding. U is a potential and $\overline{\varpi}$ denotes the effective cross section on which U acts, thus describing the local interaction of the molecules.

The following interpretation of the Boltzmann Equation (3.49) is thus possible: The total change to the particle distribution function f is equal to its advection with the velocity $\boldsymbol{\xi}$ along characteristics plus (minus) the changes which emerge from the collision of particles symbolized by Ω. For gases at thermodynamic equilibrium, the net change of the particle distribution function in collisions is zero and thus the collision operator is equal to zero $\Omega = 0$. Necessary simplifications for the derivation of the collision operator Ω include:

- the gas contains only hard, elastic molecules
- the collision times are so short that the collision has no influence on the distribution function
- only binary collisions are allowed, three particles do not collide at the same time
- there is no correlation between molecules entering a collision (Boltzmann's closure assumption)

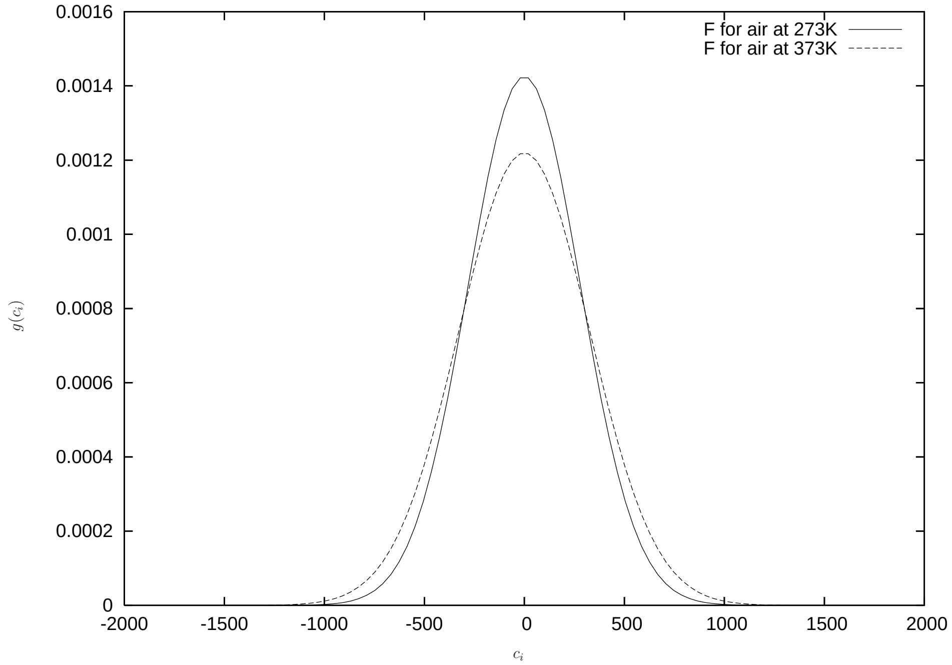


Figure 3.2: one dimensional Maxwell distribution for air at 273[K] and at 373[K]

among others[5]. Nevertheless, the Boltzmann Equation is the most general equation describing gas dynamics be it gases with Knudsen numbers $Kn \ll 1$ to free molecular streams with $Kn \gg 1$. It is therefore more general than the Naiver-Stokes Equations, which emerge from Equation (3.49) by a Chapman-Enskog expansion as a first order approximation of f for small Kn inserted in Equation (3.49).

An analytic solution for f in Equation (3.49) is only possible for special cases, one of which is a gas in equilibrium at small Kn. The basic procedure is outlined e.g. in [59] and leads to the Maxwell distribution:

$$f^{eq}(\mathbf{c}) = \frac{n}{(2\pi RT)^{\frac{D}{2}}} \cdot e^{\left(\frac{-\mathbf{c}^2}{2RT}\right)} \tag{3.52}$$

where D is the number of dimensions, R is the gas constant, T is the temperature and n is the number of molecules per considered volume[6]. It is illustrative to represent 3.52 graphically in two dimensions by decomposing it into each individual direction i by $f(\mathbf{c}^2)/n = g'(c_1)g'(c_2)g'(c_3)$ and to visualize only one component of $\mathbf{c}$ as depicted in Figure 3.2.

The area $\int g' \mathrm{d}c_i$ is constant and a measure of the density while the width can vary and is a measure for the temperature[7]. It should be noted that Equation (3.52) describes the state with

[5] It should be noted that the last assumption is violated e.g. for water, due to the hydrogen bonding. More sophisticated models exist with which even multi phase modelling is possible as presented for example in [119]. However, hydrogen bonding does seem to have only a minor effect on the results of the computed examples in this thesis.

[6] The exact definition of n is $n(\mathbf{x},t) = \lim_{\Delta V \to 0} \frac{\Delta N}{\Delta V} = \int f(t,\mathbf{x},\boldsymbol{\xi}) \cdot \mathrm{d}\boldsymbol{\xi}$

[7] The peculiar (scalar) speed c emerges as a root means square measure of the absolute values of $f^{eq}(\mathbf{c^2})$ as $c = \sqrt{3RT}$. It is directly related to the kinetic energy of a gas, see e.g. [59]

the highest probability. Therefore, without imposing external energy onto a closed system, the equilibrium state described by Equation (3.52) is reached by an enclosed gas after a sufficiently large time span. The gas then has a constant density and mass throughout space. Although in equilibrium, the gas may have a uniform flow velocity such that $\boldsymbol{c} = \boldsymbol{\xi} - \mathbf{v}$.
In Equation (3.52) $f^{eq}(\mathbf{c})$ is independent of location and time. In order to be able to describe macroscopically varying properties (e.g. density, temperature), while having local equilibrium, it is assumed as a first approximation that the equilibrium is reached instantly at the microlevel, but may vary at the macrolevel such that $f^{eq}(t, \boldsymbol{\xi}, \mathbf{x})$. This scale separation is justified when $\lim Kn \to 0$ since this implies:

$$Kn = \frac{l_f}{L} \approx \frac{\tau}{T} \to 0 \tag{3.53}$$

and therefore the collision time τ is much smaller than the timescale T relevant for the global solution of the flow field. This is the case in a wide range of flows and also for the examples discussed in this thesis. It is not the case, e.g. within shockwaves which are not considered here.

3.4.2 Macroscopic values via moments of microscopic quantities

The link between the microscopic and macroscopic quantities is given by the moments of the distribution functions. Specifically, the macroscopic density and momentum are obtained through:

$$\rho(t, \mathbf{x}) = \int_{\boldsymbol{\xi}=-\infty}^{\infty} f(t, \boldsymbol{\xi}, \mathbf{x}) \, \mathrm{d}\boldsymbol{\xi} \tag{3.54}$$

$$\rho(t, \mathbf{x}) \, \mathbf{v}(t, \mathbf{x}) = \int_{\boldsymbol{\xi}=-\infty}^{\infty} \boldsymbol{\xi} f(t, \boldsymbol{\xi}, \mathbf{x}) \, \mathrm{d}\boldsymbol{\xi} \tag{3.55}$$

In what follows, Greek indices are used for denoting a macroscopic spatial direction as opposed to the regular latin i which will later denote microscopic characteristics.
The pressure tensor $D_{\alpha,\beta}(t, \mathbf{x})$ is related to Equation (3.17) and emerges as a higher moment by considering the relative particle velocities $\boldsymbol{c} = \boldsymbol{\xi} - \mathbf{v}$:

$$D_{\alpha,\beta}(t, \mathbf{x}) = p\delta_{\alpha\beta} - \sigma_{\alpha\beta} = \int_{\boldsymbol{\xi}=-\infty}^{\infty} \boldsymbol{c}_\alpha \boldsymbol{c}_\beta \cdot f(t, \boldsymbol{\xi}, \mathbf{x}) \, \mathrm{d}\boldsymbol{\xi} \tag{3.56}$$

where $\sigma_{\alpha\beta}$ is a shear tensor stemming from the difference to the equilibrium distribution. It directly relates to Equation (3.19) and is computed by:

$$\sigma_{\alpha\beta} = \int_{\boldsymbol{\xi}=-\infty}^{\infty} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\beta \left(f^{eq} - f\right) \mathrm{d}\boldsymbol{\xi} \tag{3.57}$$

in a Boltzmann context. At equilibrium, $f \rightarrow f^{eq}$ and $\sigma \rightarrow 0$.
The momentum flux tensor $\Pi_{\alpha\beta}$ can be computed by:

$$\Pi_{\alpha\beta} = \rho v_\alpha v_\beta + D_{\alpha,\beta}\left(t, \mathbf{x}\right) = \rho v_\alpha v_\beta + p\delta_{\alpha\beta} - \sigma_{\alpha\beta} = \int\limits_{\boldsymbol{\xi}=-\infty}^{\infty} \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\beta f\left(t, \boldsymbol{\xi}, \mathbf{x}\right) \mathrm{d}\boldsymbol{\xi} \tag{3.58}$$

and is directly related to Equation (3.16). Only at $\mathbf{v} = \mathbf{0}$ is $D_{\alpha,\beta}\left(t, \mathbf{x}\right) \equiv \Pi_{\alpha\beta}$.
It can be shown, that the collision operator 3.51 has exactly five invariants, which satisfy the following property:

$$\int\limits_{\boldsymbol{\xi}=-\infty}^{\infty} \Omega\left(f\right) \cdot \psi_k \mathrm{d}\boldsymbol{\xi} = 0 \tag{3.59}$$

with $\{\psi_1 = 1, \psi_{1..3} = \boldsymbol{\xi}, \psi_4 = \boldsymbol{\xi}^2\}$. These invariants correspond to the conservation of mass, momentum and energy.

3.4.3 Simplification of the collision operator: BGK-Modell

The collision operator on the right hand side of Equation (3.49) converts the linear advection equation on the left hand side to a nonlinear integro differential equation. Although direct methods to solve this type of equations exist [3], it is computationally more advantageous to simplify the collision operator by means of an Ansatz which preserves the invariants of Equation (3.59). A widely used Ansatz is the BGK collision operator [9]:

$$\Omega_{BGK}\left(f\right) = -\frac{1}{\tau}\left(f - f^{eq}\right) \tag{3.60}$$

where τ is one global scalar parameter which determines the relaxation time at which f approaches f^{eq}. It is related to the collision rate ω as $\omega = \frac{1}{\tau}$.
The idea of modelling the collision via the "relaxation" ω to an equilibrium state can be illustrated via developing the Boltzmann equation with the BGK operator of Equation (3.60) by a Taylor series w.r.t. the time t. Truncated after the first term this gives:

$$\frac{df}{dt} = \omega\left(f - f^{eq}\right) \approx f(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t) = f(\mathbf{x}, t) + dt\, \omega\left(f(\mathbf{x}, t) - f^{eq}(\mathbf{x}, t)\right) \tag{3.61}$$

Where $\frac{df}{dt}$ symbolizes the total derivative of f w.r.t. time and ω clearly is a relaxation parameter.

3.4.4 Discretization of the BGK-Modell

The solution to the Boltzmann Equation (3.49) with the simplified collision operator given in Equation (3.60) is a distribution function $f\left(t, \boldsymbol{\xi}, \mathbf{x}\right)$ which depends on seven independent variables. For $\lim Kn \rightarrow 0$ i.e. for dense gases, it is possible to discretize the microscopic velocity $\boldsymbol{\xi}$ by a finite set of discrete velocities $\boldsymbol{e}_i$ at a set of N finite points. The Boltzmann Equation (3.49) then converts to a set of differential equations, the discrete Boltzmann equation:

$$\frac{\partial f_i}{\partial t} + e_{i\alpha} \cdot \frac{\partial f_i}{\partial x_\alpha} = -\frac{1}{\tau}\left(f_i - f_i^{eq}\right) \qquad i = 1, ..., N \tag{3.62}$$

The integrals in Equations (3.54) to (3.56) for the computation of the macroscopic flow variables are then converted to sums to perform a numerical integration via collocation points and their corresponding weights. For the density:

$$\rho\left(t, \mathbf{x}\right) = \int\limits_{\boldsymbol{\xi}=-\infty}^{\infty} f\left(t, \boldsymbol{\xi}, \mathbf{x}\right) \mathrm{d}\boldsymbol{\xi} = \sum_{i=0}^{N-1} w_i \tilde{f}_i = \sum_{i=0}^{N-1} f_i \tag{3.63}$$

where $f_i = w_i \tilde{f}_i$. Likewise, the momentum and the momentum flux are computed by:

$$\rho \mathbf{v} = \sum_i \boldsymbol{e}_i f_i \tag{3.64}$$

$$\Pi_{\alpha\beta} = \sum_i \boldsymbol{e}_{i\alpha} \boldsymbol{e}_{i\beta} f_i \tag{3.65}$$

With Equation (3.52) not being a polynominal, no exact integration is possible for a finite number of collocation points. However, the goal is to reflect the Navier-Stokes equations, which crop up as an approximation to the Boltzmann equation in the low mach number limit. The art then is to choose the collocation points, corresponding weights and the equilibrium distribution such that the invariants mass, impuls and energy correspond to Equation (3.59). A possible choice for f^{eq} is to expand Equation (3.52) in a Taylor series for small Mach numbers w.r.t. the macroscopic velocity. This leads to the following equilibrium distribution [107] :

$$f_i^{eq}\left(t, \mathbf{x}\right) = w_i \rho \left(1 + \frac{e_{i\alpha} u_\alpha}{c_s^2} + \frac{u_\alpha u_\beta}{2c_s^2}\left(\frac{e_{i\alpha} e_{i\beta}}{c_s^2} - \delta_{\alpha\beta}\right)\right) \tag{3.66}$$

where w_i are the corresponding weights.
Most of the computations in this thesis were carried out using a D2Q9 model [107] describing two dimensional flows discretized into nine discrete microscopic directions. The D2Q9 model is depicted in Figure 3.3. The coordinates of the collocation points are defined as:

$$\{\mathbf{e}_i, i = 0, \ldots, 8\} = \left\{ \begin{array}{ccccccccc} 0 & c & 0 & -c & 0 & c & -c & -c & c \\ 0 & 0 & c & 0 & -c & c & c & -c & -c \end{array} \right\} \tag{3.67}$$

They generate a space-filling lattice and mark end points of the vector valued microscopic propagation speed c. For the D2Q9 model the corresponding weights are $w_0 = \frac{4}{9}$,$w_1 = w_2 = w_3 = w_4 = \frac{1}{9}$ and $w_5 = w_6 = w_7 = w_8 = 1/36$. The speed of sound is constant within the lattice and given as $c_s = \frac{1}{\sqrt{3}} c$ and the viscosity is $\nu = \frac{\tau}{3} c^2 \rho$.

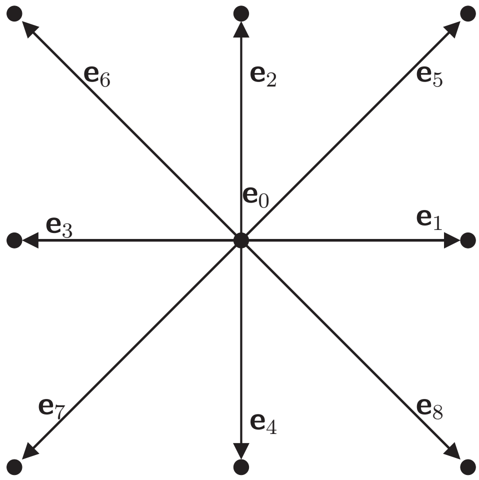


Figure 3.3: discrete velocities $\mathbf{e}_i$ of the D2Q9 model

The semi-discrete Boltzmann Equation (3.62) may be discretized by characteristic based finite elements [85], high oder finite elements using a discontinuous Galerkin formulation [33], finite volumes [101, 102, 144, 145, 134, 135] or finite differences. In this thesis the finite difference discretization according to [107] is utilized where a forward Euler finite different Ansatz is employed. This Ansatz leads to the following form of Equation (3.62):

$$\frac{f_i\left(t+\Delta t, \mathbf{x}\right)-f_i\left(t, \mathbf{x}\right)}{\Delta t}+c\frac{f_i\left(t+\Delta t, \mathbf{x}+\boldsymbol{e}_i \Delta x\right)-f_i\left(t+\Delta t, \mathbf{x}\right)}{\Delta x}=-\frac{1}{\tau}\left(f_i\left(t, \mathbf{x}\right)-f_i^{eq}\left(t, \mathbf{x}\right)\right) \tag{3.68}$$

The finite difference grid is chosen to coincide with the discretization of the microscopic velocity space such that the finite difference scheme resolves the transport along the characteristics of the right-hand side of 3.49.

Moreover, the grid spacing is chosen to be $\Delta x = c\Delta t$ such that by multiplication with Δt Equation (3.68) simplifies to:

$$f_i\left(t+\Delta t, \mathbf{x}+\boldsymbol{e}_i \Delta x\right)=f_i\left(t, \mathbf{x}\right)-\frac{\Delta t}{\tau}\left(f_i\left(t, \mathbf{x}\right)-f_i^{eq}\left(t, \mathbf{x}\right)\right) \tag{3.69}$$

The explicit forward Euler scheme introduces an artificial negative viscosity. Thus, Equation (3.69) results in an inherently instable scheme [129]. This artificial viscosity can be quantified to be $\nu = c^2\rho\left(\frac{\tau}{3}-\frac{\Delta t}{6}\right)$ and is substracted again to lead to a stable discretization. With $c = \Delta t = \Delta \mathbf{x} = 1$, the following, explicit scheme results:

$$f_i\left(t+\Delta t, \mathbf{x}+\boldsymbol{e}_i \Delta x\right)=f_i\left(t, \mathbf{x}\right)-\frac{1}{\tau}\left(f_i\left(t, \mathbf{x}\right)-f_i^{eq}\left(t, \mathbf{x}\right)\right) \tag{3.70}$$

with a viscosity of $\nu = \left(\frac{\tau}{3}-\frac{1}{6}\right)$. The pressure can be computed directly from the density as:

$$p=\frac{1}{3}\rho=\frac{1}{3}\sum_{i=0}^{8} f_i \tag{3.71}$$

The stress tensor emerges as a part of the momentum stream tensor. It is locally available and can be computed at each node from the non equilibrium distribution as:

$$S_{\alpha\beta} = -\left(1 - \frac{1}{2\tau}\right) \sum_{i=1}^{8} \boldsymbol{e}_{i\alpha} \boldsymbol{e}_{i\beta} \left(f_i - f_i^{eq}\right) \tag{3.72}$$

It directly relates to Equation (3.19). In the Lattice Boltzmann Method, the complete stress tensor τ_{ij} as given in Equation (3.17) is locally available and can be computed via a summation of Equation (3.71) and Equation (3.72) in the lattice Boltzmann context, without the need of computing derivatives of the velocity field.

It can be shown by a Chapman-Enskog analysis that the moments of Equations (3.63), (3.64) and (3.65) of the solution of Equation (3.70) converge to the incompressible Navier-Stokes equation with an error of $\mathcal{O}\left(\Delta x^2\right)$ and $\mathcal{O}\left(Ma\right)\Delta t$.

A typical scheme for implementation is to split Equation (3.70) into two steps, a collision step and a propagation step. The collision step computes the new post-collision values $\tilde{f}_i\left(t + \Delta t, \mathbf{x}\right)$ at the same grid points to be:

$$\tilde{f}_i\left(t + \Delta t, \mathbf{x}\right) = f_i\left(t, \mathbf{x}\right) - \frac{1}{\tau}\left(f_i\left(t, \mathbf{x}\right) - f_i^{eq}\left(t, \mathbf{x}\right)\right) \tag{3.73}$$

The propagation step then propagates these values to the neighboring nodes by a simple copy or memory shift operation.

3.4.5 The incompressible model

The presented formulation describes a compressible fluid where the equation of state is $p = \frac{1}{3}\rho$ and the resulting macroscopic velocities are not divergence-free. The computations performed here use an incompressible formulation first presented by [61]. This can be achieved by introducing a pressure variable as $p_i = c_s^2 f_i$. The incompressible equivalent of Equation (3.66) is:

$$f_i^{eq_{incompr}}\left(t, \mathbf{x}\right) = w_i \left(\rho + \rho_0 \left(3\frac{e_{i\alpha}u_\alpha}{c_s^2} + \frac{9u_\alpha u_\beta e_{i\alpha} e_{i\beta}}{2c_s^4} - \frac{3u_\alpha^2}{2c_s^2}\right)\right) \tag{3.74}$$

while the other parts of the method remain unchanged. The incompressible model is able to reconstruct a divergence-free velocity field for *stationary* problems. However, in practical computations, the Mach number is finite such that some compressibility remains in the model.

3.4.6 The multiple relaxation time model

The stability of the method is improved by use of the *M*ultible *R*elaxation *T*ime model MRT first introduced in [28, 29]. A very well readable contribution is also found in [78].

The basic idea is to transform the distributions f_i into an equivalent moment space before the collision is carried out. This separates moments which describe conservation quantities such as density and transport of momentum from others which can be tuned to suit numerical needs.

The moments are then relaxed by individually determined relaxation rates tuned to optimize the stability of the algorithm. This is where the MRT model derives its name from. After relaxation, the moments are transformed back to the space of the distributions. The imposition of boundary conditions as well as the propagation step, however, remain unchanged.

Chapter 4

Partitioned fluid-structure interaction

This Chapter describes the general framework of fluid-structure interaction and focuses on its partitioned solution.

4.1 Definitions

The entire physical domain termed Ω, is bounded by Γ and may be dependent on time. It is partitioned into the domain occupied by the fluid $\Omega_f(t)$ and the structural domain $\Omega_s(t)$ such that $\Omega = \Omega_f(t) \cup \Omega_s(t)$. An interface $\Gamma_{fs}(t)$ exists between $\Omega_f(t)$ and $\Omega_s(t)$ such that $\Gamma_{fs}(t) = \Omega_s(t) \cap \Omega_f(t)$ and whose position is time dependent. Γ_{fs} is termed wetted boundary. As it will be clear from context if time dependent (instationary) or time independent (stationary) problems are treated, the notation (t) is only used when it is stressed that instationary problems on deforming domains are addressed. The notation Γ_f and Γ_s is used when it is emphasized that only the portion of the wetted boundary of the fluid *or* structure is addressed.

4.2 Interface conditions

In short, two conditions need to be fulfilled at the interface. The structure is assumed impermeable for the fluid. Thus, at all times, the displacements, velocities and accelerations at Γ_{fs} of the structure must be equal to the displacements, velocities and accelerations of the fluid in contact with this boundary:

$$\mathbf{d}(t)_s = \mathbf{d}(t)_f \quad \wedge \quad \dot{\mathbf{d}}(t)_s = \dot{\mathbf{d}}(t)_f \quad \wedge \quad \ddot{\mathbf{d}}(t)_s = \ddot{\mathbf{d}}(t)_f \tag{4.1}$$

These conditions are termed the *kinematic* compatibility conditions. Similarly, the *dynamic* compatibility conditions ensure the equilibrium of tractions at all times at the boundary.

$$\mathbf{t}(t)_s = -\mathbf{t}(t)_f \tag{4.2}$$

The negative sign is due to the opposite direction of the normal vectors at the boundary.

4.2.1 Preliminary remarks on conservation of mass, momentum and energy

The consequences of the intuitive conditions presented in Equations (4.1) and (4.2) are the conservation of mass, momentum and energy at the interface. Mass is conserved at the interface if fluid and solid boundaries match at all times as required by Equation (4.1). Conservation of momentum requires that the change in momentum at the interface is to be zero as time advances:

$$\Delta\mathbf{m}_{\Gamma_{fs}}^{n\to n+1} = \int\limits_{\Gamma_{fs}} \int\limits_{t_n}^{t_{n+1}} \left(\mathbf{t}_f + \mathbf{t}_s\right) dt\, d\Gamma_{fs} = 0 \tag{4.3}$$

Equation (4.2) it is a necessary and sufficient condition in order to satisfy Equation (4.3).

The conservation of energy at the interface requires that energy is neither created nor dissipated by the interface.

$$\Delta\mathbf{E}_{\Gamma_{fs}}^{n\to n+1} = \int\limits_{\Gamma_{fs}} \int\limits_{t_n}^{t_{n+1}} \left(\mathbf{t}_f \cdot \dot{\mathbf{d}}_f + \mathbf{t}_s \cdot \dot{\mathbf{d}}_s\right) dt\, d\Gamma_{fs} = 0 \tag{4.4}$$

$\Delta\mathbf{E}_{\Gamma_{fs}}^{n\to n+1}$ is the artificial interface energy and $\mathbf{t}_s$ is the traction vector as seen from the structural point of view. $\mathbf{t}_f$ is the traction vector of the fluid on its boundary to the structure and $\dot{\mathbf{d}}_f$, $\dot{\mathbf{d}}_s$ are the corresponding interface velocities. Equation (4.4) is automatically satisfied if Equation (4.1) and Equation (4.3) are fulfilled. Since the conservation of energy contains the conservation of mass *and* momentum it will be used in Chapter 8 as first suggested by [105] as an indicator for the error introduced by the selected coupling algorithm and its influence on the stability of the computation.

It is stressed that Equation (4.4) contains an integration over time. Thus, not only does the discrete time integration method of the individual codes play a role. Additionally, the point in time at which the information is exchanged is equally important. The conservation of energy is a relevant issue for partitioned fluid structure interaction. It has been shown e.g. in [136, 93], that an incorrect transfer of energy can taint the benefits of the chosen coupling algorithm.

Two main considerations are therefore necessary in an effort to drive $\Delta\mathbf{E}_{\Gamma_{fs}}^{n\to n+1}$ to zero in partitioned, transient simulations.

A first step is to try to achieve mass and momentum conservation via an accurate transfer of forces and displacements (or velocities). This can be performed e.g. via the projection methods introduced in Chapter 4.2.2 through Chapter 4.2.4 which were employed throughout this thesis. At this stage only stationary problems are considered in which the increment of the displacements is zero as $t_n - > t_{n+1}$ and thus Equation (4.4) is always fulfilled.

In a second step instationary situations are considered where it is additionally necessary to try to reduce $\Delta\mathbf{E}_{\Gamma_{fs}}^{n\to n+1}$ either via the use of implicit methods for the overall iteration procedure or via utilizing more sophisticated explicit staggered solution procedures tailor made for this purpose. These are introduced in Chapter 4.3.

4.2.2 Preliminary remarks on transfer of variables at the boundary

The transfer of variables at the boundary may be achieved by a number of methods. The most simple method is to match the space (and time) discretization at the interface mesh. However, it is computationally more favourable to adapt the individual meshes to the needs of the discretization methods used for the fluid and the structure. Different spatial discretizations then result in non matching interface meshes. Not only do the element boundaries usually not match. In general, at the interface, the representation of $\mathbf{t}_f$, $\mathbf{t}_s$, $\mathbf{d}_f$, $\mathbf{d}_s$ may differ in their order of approximation as well such that the situation depicted in Figure 4.1 is present.

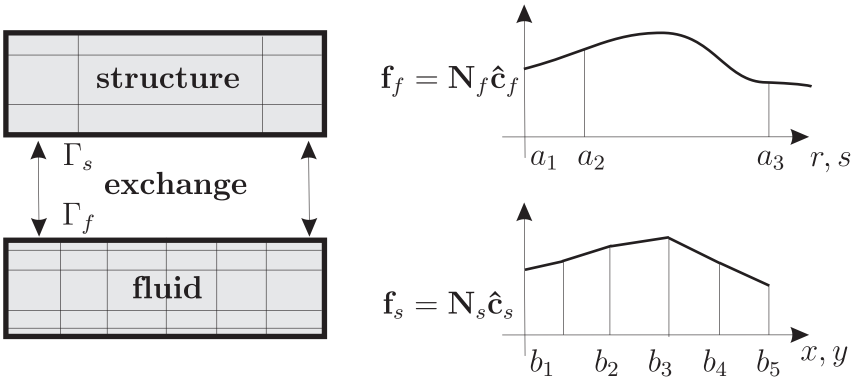

structure
Γ_s
exchange
Γ_f
fluid
$\mathbf{f}_f = \mathbf{N}_f\hat{\mathbf{c}}_f$
a_1 a_2 a_3 r, s
$\mathbf{f}_s = \mathbf{N}_s\hat{\mathbf{c}}_s$
b_1 b_2 b_3 b_4 b_5 x, y

Figure 4.1: Transfer of variables between the boundary Γ_f and Γ_s

In Figure 4.1, the fluid variables i.e. tractions and velocities are represented by a vector valued function $\mathbf{f}_f$ described by its Ansatz functions composed of coefficients $\hat{\mathbf{c}}_f$ and their basis $\mathbf{N}_f(r, s)$ defined in fluid innate coordinates, here r, s. The solid, on the other hand, must be able to "see" an equivalent function now termed $\mathbf{f}_s$, expressed in its own coordinate system, say (x, y), with its own coefficients $\hat{\mathbf{c}}_s$ in its own basis $\hat{\mathbf{N}}_s(x, y)$. A mapping must be defined to exchange these functions i.e. to be able to represent one and the same function on different grids with different bases. To achieve this, the main task is to tune the (free) coefficients of the Ansatz into which the given function is to be projected.

Force transfers between non-matching interface meshes have been investigated before. A famous approach was first introduced in [40], but unfortunately it implicitly requires the Ansatz functions to sum up to one at every point of the domain. This is *not* the case for the hierarchic Ansatz functions introduced in [125] and used in the structural discretization employed in this thesis. Coupling high order structural finite element methods to fluid solvers thus requires a more subtle approach. One remedy is to transfer forces by a *composed integration* scheme as utilized in this thesis. It was first used by [109, 80] in the context of the *hp-d* method[1]. Convergence studies of this method concerning fluid-structure interaction were investigated in [123] and published in [118]. Its application to coupling high-order solid discretizations with fluids can be found in [75]. An article by Xiangmin [146] also treats the subject in the broader setting of data transfer between non matching meshes in multiphysics simulations. Herein this method is called *common refinement*. The force transfer utilized in the thesis at hand can be viewed as a subset of the method described by [146]. A more recent investigation into the subject was carried out by [74] in the context of high order structural and fluid solvers loosely coupled in time.

[1] The *hp-d* method is further extended to model adaptive computations in [36]

A general comparison of the currently available methods and their performance w.r.t. errors may be found in e.g. [22, 23].
Among these, a $L2$ projection of the form:

$$\int\limits_{\Gamma_{fs}} (\mathbf{f}_f - \mathbf{f}_s)^2 \, d\Gamma_{fs} \rightarrow min \tag{4.5}$$

was chosen. The reasons for this choice are highlighted in the following:

- The $L2$ projection of the data is equivalent to a weighted residual approach forming the basis of the FEM. The error is thus in the same range as the one inherent to the methods used for the discretization of the fields themselves.
- By construction, an $L2$ projection is conservative in an integral sense and consequently preserves the momentum likewise.
- For the force transfer, a composed integration is equivalent to an $L2$ projection. No additional computational effort is needed.

The first point is fundamental to functional analysis and proven e.g. in [39]. It can easily be reconstructed by inserting the definitions of the functions $\mathbf{f}_s, \mathbf{f}_f$ into Equation (4.5). To find the minimum, the residual $\mathbf{R} = \mathbf{N}_f \hat{\mathbf{c}}_f - \mathbf{N}_s \hat{\mathbf{c}}_s$ must be differentiated w.r.t. the free coefficients. The common basis can then be factored. In case of a projection of $\mathbf{f}_f$ to $\mathbf{f}_s$ the final form reads:

$$\int\limits_{\Gamma_{fs}} \mathbf{N}_s^T \left(\mathbf{N}_f \hat{\mathbf{c}}_f - \mathbf{N}_s \hat{\mathbf{c}}_s \right) \mathrm{d}\Gamma_{fs} = \mathbf{0} \tag{4.6}$$

In Equation (4.6), the residual $\mathbf{R} = \mathbf{N}_f \hat{\mathbf{c}}_f - \mathbf{N}_s \hat{\mathbf{c}}_s$ is weighted by a weighting function $\mathbf{N}_s$. This is similar to a standard Galerkin procedure, in which the weighting function is chosen equal to the one of the discretization of the field variables.
The second point can be reproduced by considering that Equation (4.6) is valid for the basis $\mathbf{N}_s$ which must form a complete basis also containing $N_s = 1$. For a stationary problem, a comparison to Equation (4.2) then demonstrates the momentum conservation in an integral sense.
The third point is shown in Chapter 4.2.3.

4.2.3 Traction transfer

Specifically, projecting the forces from the fluid boundary to the structural boundary the following expression has to be minimized:

$$\int\limits_{\Gamma_{fs}} (\mathbf{t}_s(x,y) - \mathbf{t}_f(r,s))^2 \, d\Gamma_{fs} \rightarrow min \tag{4.7}$$

where $\mathbf{t}_s(x,y)$ is the load function as seen from the structural point of view. $\mathbf{t}_f(r,s)$ is the force function on the fluid boundary *computed by the fluid solver* and is therefore given. Clearly,

$\mathbf{t}_f(r,s)$ needs to be transferred to $\mathbf{t}_s(x,y)$ and the only way to minimize (4.7) is to adjust the representation of $\mathbf{t}_s(x,y)$ on the structural side. The integral (4.7) needs to be minimized over the entire wetted surface Γ_{fs}. $\mathbf{t}_s(x,y)$ as well as $\mathbf{t}_f(r,s)$ are defined on their own domain (x,y) and (r,s) and it is assumed that a proper mapping was defined such that (r,s) may be mapped onto (x,y) and vice versa. Therefore, in what follows the coordinates are not explicitly denoted. The function $\mathbf{t}_s$ is represented by piecewise polynomials defined by the vector of coefficients $\hat{\mathbf{t}}$ and the matrix of the shape functions $\mathbf{N}$. Expanding $\mathbf{t}_s(x,y)$ leads to:

$$\int\limits_{\Gamma_{fs}} \left(\mathbf{N}_s\hat{\mathbf{t}}_s - \mathbf{t}_f\right)^2 d\Gamma_{fs} \rightarrow min \tag{4.8}$$

$\mathbf{t}_f$ is prescribed, but the coefficients $\hat{\mathbf{t}}_s$ may be adjusted to solve the least squares problem. To minimize the integral (4.8) one needs to derive (4.8) w.r.t. to these free variables $\hat{\mathbf{t}}_s$. Solving for $\hat{\mathbf{t}}_s$ results in:

$$\hat{\mathbf{t}}_s = \mathbf{M}^{-1} \int\limits_{\Gamma_{fs}} \mathbf{N}_s^T \mathbf{t}_f d\Gamma_{fs} \tag{4.9}$$

In equation (4.9) $\mathbf{M}$ corresponds to a mass matrix defined as:

$$\mathbf{M} = \int\limits_{\Gamma_{fs}} \mathbf{N}_s^T \mathbf{N}_s d\Gamma_{fs} \tag{4.10}$$

The discretized load vector on the structural side $\mathbf{F}_s$ is defined as the integral over the structural shape functions $\mathbf{N}_s^T$ times the traction vector $\mathbf{t}_s$:

$$\mathbf{F}_s = \int\limits_{\Gamma_{fs}} \mathbf{N}_s^T \mathbf{t}_s d\Gamma_{fs} = \mathbf{M}\hat{\mathbf{t}}_s \tag{4.11}$$

It is important to remember that $\mathbf{t}_s$ is actually the load function imposed onto the structure by the fluid and that we may choose the coefficients $\hat{\mathbf{t}}_s$ to minimize (4.7). Hence, we can substitute the solution of the minimization (4.9) into the definition of the force vector (4.11):

$$\mathbf{F}_s = \int\limits_{\Gamma_{fs}} \mathbf{N}_s^T \mathbf{t}_f d\Gamma_{fs} \tag{4.12}$$

Now, (4.12) represents the force vector which needs to be integrated *exactly* to perform an accurate least squares projection of the fluid forces onto the structure. It is worth emphasizing the following points:

1. There is no need to explicitly invert the mass matrix or solve a system of equations in the process of minimizing (4.8) in this context. Exact integration of (4.9) guarantees a $L2$ projection.

2. This projection is equivalent to a Galerkin projection of the fluid function into the function space used by the structural solver.
3. This holds independently of the type and degree of the shape functions used.

The idea of the composed integration is very simple. For a computation of the left-hand side of Equation (4.12), the integral needs to be evaluated partially as a sum of sub integrals. For example in Figure 4.1, the right-hand side of Equation (4.12) is evaluated as:

$$\int\limits_{\Gamma_{fs}} \mathbf{N}_s^T \mathbf{t}_f d\Gamma_{fs} = \sum_{i=b_1}^{b_4} \int\limits_{i}^{i+1} \mathbf{N}_s^T \mathbf{t}_f d\Gamma_{\Delta i} \tag{4.13}$$

This process requires the knowledge of the footprint of the fluid discretization on the structural mesh, i.e. the location of b_i on the boundary Γ_s. This footprint is computed via mesh intersection, see e.g. [123].

4.2.4 Velocity and displacement transfer

Unfortunately, the $L2$ projection has to be computed explicitly for the transfer of the velocities. This is due to the fact that the mass matrix does not cancel out as conveniently as is the case for the force transfer. Instead the coefficients must be computed via an explicit solution of:

$$\mathbf{M}_f \dot{\mathbf{d}}_f = \int\limits_{\Gamma_{fs}} \mathbf{N}_f^T \dot{\mathbf{d}}_s d\Gamma_{fs} \tag{4.14}$$

in which the mass matrix is now composed of the fluid shape functions:

$$\mathbf{M}_f = \int\limits_{\Gamma_{fs}} \mathbf{N}_f^T \mathbf{N}_f d\Gamma_{fs} \tag{4.15}$$

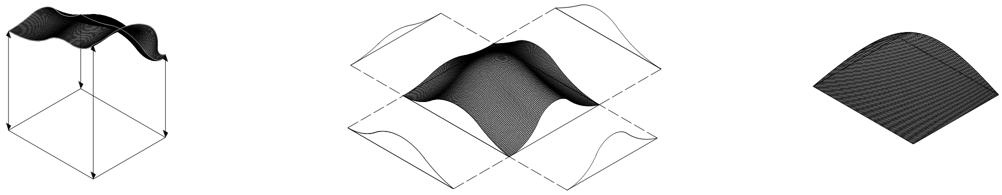
The evaluation of the right hand side of Equation (4.14) must again be carried out via a composed integration.

In principle, the projection defined in Equation (4.14) must be carried out globally over the entire wetted boundary to ensure global conservation of momentum. To avoid having to solve a global system, the hierarchy of the basis stored in $\mathbf{N}_f$ is exploited. First, the nodal modes are projected, i.e. the corresponding nodes are set to have the same value. This guarantees C^0 continuity. Next the edge modes are projected and last, if present, the face modes. The procedure is depicted in Figure 4.2.

This hierarchic type of projection is termed $H^{\frac{1}{2}}$ and layed out e.g. in [73][2] The $H^{\frac{1}{2}}$ projection is not momentum preserving but is a good compromise between accuracy and efficiency. For numerical studies and comparison to an $L2$ projection as well as an interpolation see [99].

The $H^{\frac{1}{2}}$ projection is carried out explicitly in the case the spatial resolution of the velocities or displacements at the boundary have an order of greater then one in the fluid solver. While

[2] The same projection is termed projection based interpolation in [26].



(a) vertex projection (b) edge projection (c) face projection

Figure 4.2: Visualization of a $H^{\frac{1}{2}}$ projection

this is the case for coupling to the spectral element method as in Chapter 7, the Finite Volume Method only represents the displacements with linear shape functions. For the coupling to the Finite Volume Method in Chapter 7, a simple evaluation of the displacements at the structural boundary at the footprint of the nodes of the fluid mesh is performed. The $H^{\frac{1}{2}}$ projection is then equivalent to an interpolation.

4.3 Algorithms for partitioned fluid-structure interaction

One of the first publications for a partitioned solution of fluid structure interaction was by Felippa [43]. Quite a number of different algorithms have been suggested in the literature since. A good review of the available methods can be found e.g. in [95, 82, 83, 27].

Most of the methods can be classified as either explicit or implicit in their nature. The general concept will be introduced and only those methods will be laid out in detail, which have actually been applied in the context of this thesis.

Algebraically, the coupled problem can be represented by (see e.g. [95]):

$$\begin{bmatrix} \mathbf{A}^f_{II} & \mathbf{A}^f_{I\Gamma} & \mathbf{0} \\ \mathbf{A}^f_{\Gamma I} & \mathbf{A}^f_{\Gamma\Gamma} + \mathbf{A}^s_{\Gamma\Gamma} & \mathbf{A}^s_{\Gamma I} \\ \mathbf{0} & \mathbf{A}^s_{I\Gamma} & \mathbf{A}^s_{II} \end{bmatrix} \begin{pmatrix} \mathbf{x}^f_I \\ \mathbf{x}_\Gamma \\ \mathbf{x}^s_I \end{pmatrix} = \begin{pmatrix} \mathbf{f}^f_I \\ \mathbf{f}_\Gamma \\ \mathbf{f}^s_I \end{pmatrix} \tag{4.16}$$

The matrix $\mathbf{A}$ is a coefficient matrix which is in general nonlinear and thus depends on the solutions $\mathbf{x}$. The indices $_I$ and $_\Gamma$ indicate with which part of the domain the coefficient matrix is associated with. In case it can be assembled by considering internal states only the double index in$_{II}$ is used whereas $_{\Gamma\Gamma}$ indicates that the coefficients emerge from the boundary only. A mixture of both is possible as well. The generic interface variable $\mathbf{x}$ is used to symbolize displacements, velocities and tractions alike.

The objective now is to solve for $\mathbf{x}$. On this abstract level, all solution methods for solving nonlinear systems of equations may in principle be employed: From block Gauss-Seidel techniques possibly accelerated by over- or stabilized by underrelaxation to Newton methods.

4.3.1 Explicit coupling schemes

Explicit coupling schemes attempt the solution of the coupled problem by exchanging the required data only once per timestep. They are thus also termed loosely coupled schemes. In an FSI context, they were first reported by [43] and have been used by a countless number of authors since. Review articles such as e.g. [100, 106, 105, 44] have summarized and analyzed a large number of possible algorithms for the partitioned solution of coupled mechanical systems. Their analysis extends far beyond the procedure presented in this chapter.

Generally, explicit coupling schemes are unable to obey the conservation conditions given by Equations (4.1), (4.2) and (4.4) as a one step solution to the coupled nonlinear system is only possible approximately. This eventually leads to instabilities in case of low solid to fluid density ratios [17, 45] especially in combination with incompressible fluids. However, explicit methods prove to be stable, efficient and accurate for most compressible flow simulations. Moreover, the error induced by these procedures is measurable, can be computed online and it is thus deducible if the instability stems from the explicit coupling algorithm itself or has other, possibly physical sources. For these reasons, explicit coupling methods are widely used in partitioned FSI.

The simplest way of solving a coupled system iteratively in an explicit manner is to follow these intuitive steps:

1. advance the fluid system from t_n to t_{n+1} and compute the forces at the wetted boundary
2. apply the loads on the structural boundary and advance the structural system from t_n to t_{n+1}
3. compute the structural displacements at the boundary and apply them to the fluid boundary

and hope the computation remains stable. A generalization of this scheme has been analyzed in detail in [106] and is termed *c*onventional *s*erial *s*taggered (CSS) procedure. It reads:

1. predict the structural displacement of the wetted boundary at time t_{n+1} e.g. using

   $$\mathbf{d}^{(n+1),p} = \mathbf{d}^n + \alpha_0 \Delta t_s \dot{\mathbf{d}}^n + \alpha_1 \Delta t_s (\dot{\mathbf{d}}^n - \dot{\mathbf{d}}^{n-1}) \tag{4.17}$$

   where $\mathbf{d}^{(n+1),p}$ is the predicted position of the structural wetted boundary and α_0, α_1 are two real constants determining the order of the prediction. First order accuracy in space is obtained by setting: $\alpha_0 = 1 \wedge \alpha_1 = 0$. Second order spatial accuracy is obtained by setting $\alpha_1 = 1 \wedge \alpha_1 = \frac{1}{2}$.

2. update the position of Γ_f to the position $\mathbf{d}^{(n+1)_p}$. Then, advance the fluid-subsystem from time step t^n to t^{n+1}. Subcycle the flow solver if $\Delta t_s > \Delta t_f$

3. compute the fluid tractions $\mathbf{t}_f^{(n+1)}$ at Γ_f. These are the fluid tractions which would have resulted if the structure had been advanced to $\mathbf{d}^{(n+1),p}$. From this, compute a corrected traction vector $\mathbf{t}_s^{(n+1),c}$

4. apply $\mathbf{t}_s^{(n+1),c}$ to Γ_s and advance the structural system from t_n to t_{n+1}

An energy based, analytical evaluation of the CSS procedure has been performed for a simplified setting in [105][3]. A detailed presentation is given in [95] as well. The principle idea is as follows:
The artificial energy created at the interface is composed of the energy created by fluid solver $\Delta\mathbf{E}_{\Gamma_f}^{n\to n+1}$ as its interface moves from $\mathbf{d}^{(n)}$ to $\mathbf{d}^{(n+1)}$ and the energy created by the structural solver $\mathbf{E}_{\Gamma_s}^{n\to n+1}$:

$$\Delta\mathbf{E}_{\Gamma_{fs}}^{n\to n+1} = \Delta\mathbf{E}_{\Gamma_f}^{n\to n+1} + \Delta\mathbf{E}_{\Gamma_s}^{n\to n+1} \tag{4.18}$$

The individual energies can be computed by:

$$\begin{aligned} \Delta\mathbf{E}_{\Gamma_f}^{n\to n+1} &= -\left(\theta\,\mathbf{t}_f^{n+1} + (1-\theta)\mathbf{t}_f^{n}\right)\cdot\left(\mathbf{d}_f^{(n+1),p} - \mathbf{d}_f^{n,p}\right) & (4.19)\\ \Delta\mathbf{E}_{\Gamma_s}^{n\to n+1} &= \left((1-\alpha_f)\mathbf{t}_s^{(n+1),c} + \alpha_f\mathbf{t}_s^{n,c}\right)\cdot\left(\mathbf{d}_s^{n+1} - \mathbf{d}_s^{n}\right) & (4.20) \end{aligned}$$

Explicit coupling procedures *always* produce artificial energy at the interface [105]. This can easily be comprehended by considering the favourable case of integrating the structure by the trapezoidal rule ($\alpha_f = 0$) and the fluid by a backward Euler method ($\theta = 1$). Then, the interface Energy would be:

$$\Delta\mathbf{E}_{\Gamma_{fs}}^{n\to n+1} = \mathbf{t}_s^{(n+1),c}\cdot\left(\mathbf{d}_s^{n+1} - \mathbf{d}_s^{n}\right) - \mathbf{t}_f^{n+1}\cdot\left(\mathbf{d}_f^{(n+1),p} - \mathbf{d}_f^{n,p}\right) \neq 0 \tag{4.21}$$

The "corrector" can easily be chosen to lead $\mathbf{t}_s^{(n+1),c} \equiv \mathbf{t}_f^{n+1}$. However, the "predictor" is inherently wrong and can only give $\mathbf{d}_f^{(n+1),p} \approx \mathbf{d}_s^{n+1}$ such that $\Delta\mathbf{E}_{\Gamma_{fs}}^{n\to n+1} \neq 0$
Thus, one can merely try to minimize the energy artificially produced at the interface at each time step by choosing an appropriate combination of $\mathbf{d}_f^{(n+1),p}$ and $\mathbf{t}_s^{(n+1),c}$. Each choice gives a particular instance of the CSS procedure.
The interface energy $\Delta\mathbf{E}_{\Gamma_{fs}}^{n\to n+1}$ may be computed discretely at the interface for each time step and summed up to give the total artificial interface energy $\mathbf{E}_{tot}$.

$$\mathbf{E}_{tot} = \sum_{n=1}^{n=n_{current}} \mathbf{E}_{\Gamma}^{n\to n+1}. \tag{4.22}$$

By monitoring $\mathbf{E}_{tot}$ it is possible to assess the influence of the explicit coupling algorithm on the stability of the coupled computation. A sudden increase of $\mathbf{E}_{tot}$ hints a loss of stability due to the partitioned, staggered solution. If a loss of stability is encountered without a sudden increase of $\mathbf{E}_{tot}$, then either an instable integration inside the fluid- or structural solver is the culprit or the system is physically instable. A negative $\mathbf{E}_{tot}$ hints artificial damping.
This criterion has been used to asses the stability of the coupling to the lattice Boltzmann solver in Chapter 8.

[3] The most important simplifications include: 1st consideration of a single point on the fluid-structure interaction interface, and surrounding patch of unit length in two dimensions, and unit area in three dimensions 2nd invicit flow, 3rd integration of the structural subsystem via the 2nd order accurate version of the Newmark method

Other more stable algorithms have been developed. These are usually catered for the use in ALE formulations and not utilized in this thesis but are mentioned for completeness. Among these is the *I*mproved *S*erial *S*taggered *P*rocedure (ISS), whose main goal is to meet yet another important condition which has been neglected so far: the *G*eometric *C*onservation *L*aw (GCL).

The GCL [127, 137] basically states, that the change in area (volume) of each control volume between t^n and t^{n+1} must be equal to the area (volume) swept by the cell boundary during $\Delta t = t^n - t^{n+1}$.

A carefully designed algorithm to achieve second order temporal accuracy while at the same time showing a very stable behaviour is laid out in [42].

Despite the advantages of these more sophisticated procedures solely the CSS procedure was used in this thesis as it proved to be sufficiently accurate and stable for computing the examples in Chapter 8.

4.3.2 Implicit coupling schemes

Unlike explicit partitioned approaches, implicit coupling schemes attempt to obey Equations (4.1), (4.2) and (4.4) by iterating within the same timestep over both fields until convergence is obtained. Provided this inter-field iteration converges they are stable w.r.t. advancing the coupled system in time also for incompressible flows and for unfavourable fluid to structural density ratios. This stability comes at the cost of efficiency.

The approach of iterating over both fields until convergence can be considered as a fixed point iteration scheme applied to the usually large non-linear system of equations as represented by Equation (4.16). Symbolically:

$$\mathbf{A}(\mathbf{x}) = \mathbf{x} \tag{4.23}$$

in which $\mathbf{x}$ is the fixed point of $\mathbf{A}$. The system described by Equation (4.23), can be solved efficiently by condensing it to the interface degrees of freedom $\mathbf{x}_\Gamma$, solving for these iteratively and recovering the internal solutions via backward substitution. The fixed point iterations can therefore be considered as being carried out on the interface degrees of freedom only. An interface operator $\mathbf{C}$ can be defined as:

$$\mathbf{C} : \bar{\mathbf{x}}_{i+1}^{n+1} = \mathbf{A}_\Gamma^s \left(\mathbf{A}_\Gamma^f \left(\mathbf{x}_i^{n+1} \right) \right) \tag{4.24}$$

In Equation (4.24) $\mathbf{A}_\Gamma^{f,s}$ now symbolizes the coefficients related to the concerned fluid (f) or solid (s) boundary. The interface operator $\mathbf{C}$ maps the generic variables $\mathbf{x}$, implicitly containing only variables on the interface $\mathbf{x} = \mathbf{x}_\Gamma$, from iteration i to iteration $i + 1$. The time step is fixed to the current time step t^{n+1} (for implicit solvers) and may be formulated using either the displacements, velocities, or the interface tractions. The bar in $\bar{\mathbf{x}}$ hints that the current solution was only obtained by a crude application of the operator $\mathbf{C}$ and has not been postprocessed in any way. Assuming for a minute that $\mathbf{x}$ represents the displacements $\mathbf{d}$ at the interface, then the solution $\mathbf{d}$ at the interface at $\mathbf{d}_{i+1}^{t+1}$ is found as follows. Application of the displacements at iteration i to the boundary of the structural solver via $\mathbf{t}_{\mathbf{i+1}}^{\mathbf{n+1}} = \mathbf{A}_\Gamma^f \left(\mathbf{d}_i^{n+1} \right)$

gives the tractions $\mathbf{t_{i+1}^{n+1}}$ on the boundary at iteration step $i+1$. These are directly applied to the structural boundary $\mathbf{d}_{i+1}^{n+1} = \mathbf{A}_{\Gamma}^{s}\left(\mathbf{t_{i+1}^{n+1}}\right)$ to lead to the new displacement solution $\mathbf{d}_{i+1}^{n+1}$. The convergence of the iteration is usually measured by the discrete, length scaled $L2$ norm $\|\cdot\|$ of the residual $\mathbf{r}$ between two consecutive applications of $\mathbf{C}$ as:

$$r_{i+1}^{n+1} = \frac{1}{n_{eq}} \|\mathbf{x}_{i+1}^{n+1} - \mathbf{x}_{i}^{n+1}\| = \frac{1}{n_{eq}} \|\mathbf{r}_{i+1}^{n+1}\| \tag{4.25}$$

If r_{i+1}^{n+1} is smaller than some chosen tolerance ϵ, the iteration is considered to have converged. The repeated application of Equation (4.24) leads to a sequence of solution vectors

$$\{\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_i, \mathbf{x}_{i+1}\} \tag{4.26}$$

whose convergence properties are determined by the fixed point theorem of Banach.

At this point it is worth highlighting an important prerequisite for the sequence (4.26) to converge: Both field solvers need to utilize implicit formulations in which they seek equilibrium at time step t^{n+1} using boundary values at time step t^{n+1}. If one of the field solvers utilized an explicit formulation[4] then the variables at t^{n+1} would never be changed. As a consequence, the same results would be obtained for all iterations $0, 1, ..., i, i+1$ at t^{n+1} and no convergence of the global series could be obtained with any of the implicit (i.e. strongly coupled) algorithms discussed in this thesis.

Strongly coupled field iterations with field solvers employing explicit algorithms are not discussed in this thesis. Such an algorithm was proposed by the author in cooperation with Geller (see [52]) to couple the p-FEM to the Lattice Boltzmann Method. Yet, the algorithm suggested in Chapter 8.2 turned out to be more efficient such that there was no need to pursue that path.

A further detail regarding the implementation of implicit coupling schemes into commercial solvers such as CFX or FLUENT is mentioned. In order to fix the timestep i.e. not to advance the computation in time, the state variables at state t^n must only be overwritten by the field solvers, if the next timestep is to be computed. The aforementioned commercial fluid solvers, however, internally overwrite this state in their current implementation at every coupling step or there values are not accessible without profound insider knowledge. The solution within the FSI-cycles might then converge to a wrong value or not converge at all. For now, this could only be solved by an expensive restart. A more intelligent solution is surely available for the programmers of the commercial software, but for external users this imposes a severe limitation causing serious difficulties for the implementation of truely implicit (i.e. strongly coupled) iteration schemes into CFX or FLUENT.

4.3.2.1 Block Gauss-Seidel Iteration

The rate of convergence and stability of the series 4.26 may be influenced by relaxing the next trial solution according to:

$$\mathbf{x}_{i+1} = \omega_i \bar{\mathbf{x}}_{i+1} + (1 - \omega_i)\, \mathbf{x}_i \tag{4.27}$$

[4] In the sense that it only used information from time step t^n to advance to t^{n+1}.

where ω_i is termed relaxation parameter. The convergence can be accelerated by choosing $1 < \omega_i \leq 2$ at the cost of stability or the stability may be improved by choosing $0 < \omega < 1$ with the effect of decreasing the rate of convergence. In linear algebra this method is known as successive over- or underrelaxation. An optimal relaxation parameter can be computed via the eigenvalues of the coefficient matrix $\mathbf{A}$ in Equation (4.23). This is, however, only possible for linear "toy" problems and in general senseless since the knowledge of the eigenvalues of $\mathbf{A}$ already provides the solution of Equation (4.23) directly.

As for realistic applications, either a predefined (non optimal) ω_i is chosen which is kept constant throughout all iterations or ω_i is determined per iteration by a vector acceleration method (i.e. by "advanced guessing"). Such an adaptive choice possibly increases the rate of convergence and the stability. In fluid-structure interaction, the vector iteration acceleration method proposed by Irons and Tuck [70] has shown to lead to good convergence and stability of the overall implicit iteration procedure. Its application to FSI was suggested by [95] and compared to other methods in [95, 82, 83]. It starts with two known pairs of interface variables $\{\bar{\mathsf{x}}_i, \mathsf{x}_{i-1}\}$, $\{\bar{\mathsf{x}}_{i+1}, \mathsf{x}_i\}$ and determines the relaxation factor ω_i for each new iteration by:

$$\omega_i = -\omega_{i-1} \frac{\mathbf{r}_i^T \left(\mathbf{r}_{i+1} - \mathbf{r}_i\right)}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|^2} \tag{4.28}$$

Equation (4.28) does not guarantee that $0 < \omega_i < 2$ which is necessary for the stability of the overall iteration process. Therefore, ω_i is usually (manually) limited to $\epsilon < \omega_i < 2$ where ϵ is a small number. In all examples computed in this thesis $\epsilon > 0.1$ was chosen. This method was applied e.g. in Chapter 7.

4.3.2.2 Interface-GMRES

At the time of investigating convergence acceleration techniques a promising method seemed to be the Interface-GMRES method suggested by Michler [92]. While its application to compressible flow solvers leads to good results, its transfer to *in*compressible flows disclosed deficiencies of the method. Recently, the method has also received attention in [83] where it was classified as a vector extrapolation scheme for the lack of explicitly using the interface Jacobian. This interesting insight is, however, not regarded here and the method will be presented in the sequel as originally introduced in [92]. Its application to incompressible flows is demonstrated in Chapter 6 where its drawbacks w.r.t. other methods such as presented in Chapter 4.3.2.1 are discussed.

The idea is to solve Equation (4.23) by a hybrid Newton-Krylov method as presented generically in [15] for systems of nonlinear equations. Starting point is the nonlinear system (4.23) which can be rewritten as follows:

$$\mathbf{R}\mathsf{x} = \mathbf{0} \tag{4.29}$$

The residual operator is defined as $\mathbf{R} = \mathbf{C} - \mathbf{I}$ such that the residual $\mathbf{r}$ can be computed by:

$$\mathbf{r}_i = \mathbf{C}\mathsf{x}_i - \mathbf{I}\mathsf{x}_i = \mathsf{x}_{i+1} - \mathsf{x}_i \tag{4.30}$$

where bar $\bar{\mathsf{x}}$ is now neglected for simpler notation. It will be clear from the context in the sequel if the solution x has been "postprocessed".

The Newton method involves a linearization step and an update of the linearized state $\mathbf{x}_0 \leftarrow \mathbf{x}_0 + \Delta\mathbf{x}$ according to:

$$\mathbf{R}\mathbf{x}_0 + \mathbf{J}\mathbf{x}' = \mathbf{0} \quad \Longleftrightarrow \quad \mathbf{J}\mathbf{x}' = -\mathbf{R}\mathbf{x}_0 \quad \Longrightarrow \quad \mathbf{x}' = -\mathbf{J}^{-1}\mathbf{R}\,\mathbf{x}_0 \tag{4.31}$$

$$\mathbf{x}_0 = \mathbf{x}_0 + \mathbf{x}' \tag{4.32}$$

where $\mathbf{J} = \frac{\partial \mathbf{R}}{\partial \mathbf{x}}|_{\mathbf{x}_i}$ denotes the Jacobian at that linearization state and the notation $'$ instead of Δ is employed. At each Newton step, one needs to solve the linear system (4.31). The initial guess $\mathbf{x}_0$ is then updated at each Newton iteration according to Equation (4.32) to form the next approximation at which another linearization is carried out.
The procedure thus generates the following series of vectors from an initial guess $\mathbf{x}_0$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{x}_i' = \mathbf{x}_i - \mathbf{J}^{-1}\mathbf{R}\,\mathbf{x}_i \tag{4.33}$$

where $\mathbf{x}_i{}' = \mathbf{J}^{-1}\mathbf{R}\mathbf{x}_i$ is a perturbation around the current linearization state $\mathbf{x}_i$
In FSI, neither the Jacobian $\mathbf{J}$ is known nor its inverse[5]. It is shown in [15] that, if a Krylov method is employed for the solution of the linear subproblem, then $\mathbf{J}$ is only required in form of a matrix-vector product. The update $\mathbf{x}_i'$ is therefore approximated in a Krylov space of order m associated with the linear problem of Equation (4.31).

$$\mathbf{x}_0^{'} \in \mathcal{K}^m := span\{\mathbf{x}_j - \mathbf{x}_0\}_{j=1}^{j=m} \tag{4.34}$$

Substitution of this Ansatz into the linear problem Equation (4.31) gives:

$$\mathbf{R}\mathbf{x}_0 + \mathbf{J}\sum_{j=1}^{m} \alpha_j \left(\mathbf{x}_j - \mathbf{x}_0\right) = \mathbf{R}\mathbf{x}_0 + \sum_{j=1}^{m} \alpha_j \mathbf{J}\left(\mathbf{x}_j - \mathbf{x}_0\right) = \mathbf{0} \tag{4.35}$$

Further, the matrix vector product $\mathbf{J}\left(\mathbf{x}_j - \mathbf{x}_0\right)$ is approximated by finite differences:

$$\mathbf{J}\left(\mathbf{x}_j - \mathbf{x}_0\right) \approx \mathbf{r}_j - \mathbf{r}_0 \equiv \mathbf{x}_{j+1} - \mathbf{x}_j - \mathbf{r}_i \tag{4.36}$$

where an extra approximation error occurs explaining the choice for ≈ instead of =. Substitution of Equation (4.36) into Equation (4.35) leads to:

$$\mathbf{r}_i + \sum_{j=i}^{m} \alpha_j \left(\mathbf{r}_j - \mathbf{r}_i\right) = \mathbf{0} \tag{4.37}$$

In this expression, a residual sensitivity space $span\{\mathbf{r}_j - \mathbf{r}_0\}$ appears which corresponds to the one chosen for $\mathbf{x}_0^{'}$ in Equation (4.34)

[5] Crude computation of $\mathbf{J}$ via finite differences is prohibitively expensive.

As is typical for a *G*eneralized *M*inimum *RES*sidual (GMRES) algorithm, the coefficients α_i necessary for the redefinition of the linear reference state $\mathbf{x}_i$ are determined by solving Equation (4.37) in a least squares sense according to:

$$\boldsymbol{\alpha} = arg\ min \|\mathbf{r}_i + \sum_{j=1}^{m} \alpha_j \left(\mathbf{r}_j - \mathbf{r}_i\right)\| \tag{4.38}$$

The residual $\mathbf{r}_j$ is a matrix and has the dimension of the interface degrees of freedom times the Krylov subspace currently approximating the linearization state j. The minimization in Equation (4.38) is equivalent to solving the following system for $\boldsymbol{\alpha}$:

$$\mathbf{r}_j^T \mathbf{r}_j \boldsymbol{\alpha} = -\left(\mathbf{r}_i^T \mathbf{r}_j\right)^T \tag{4.39}$$

The coefficient matrix $\mathbf{r}_j^T \mathbf{r}_j$ is of dimension $j \times j$. With j typically being of magnitude $j \leq 10$, the minimization in Equation (4.38) needs almost no computing time.
The norm of the residual is used as an estimate of the norm of the nonlinear problem:

$$\xi = \|\mathbf{r}_i + \sum_{j=1}^{m} \alpha_j \left(\mathbf{r}_j - \mathbf{r}_i\right)\| \tag{4.40}$$

The complete algorithm as presented in [92] is given in Algorithm A-1.

Algorithm A-1 Interface GMRES

```
 1: $i = 0$; $\mathbf{x}_1 = \mathbf{C}\mathbf{x}_0$; $\mathbf{r}_0 = \mathbf{x}_1 - \mathbf{x}_0$
 2: while $\|\mathbf{r}_i\| > \epsilon_0$ do
 3:     $j = 0$, $\xi = \|\mathbf{r}_i\|$
 4:     while $\xi > \epsilon_1$ do
 5:         $j = j + 1$
 6:         $\mathbf{x}_j^{'} = \mathbf{x}_j - \mathbf{x}_0$
 7:         for $k = 1$ to $j - 1$ do
 8:             $\mathbf{x}_j^{'} = \mathbf{x}_j^{'} - \mathbf{x}_k^{'} \left(\mathbf{x}_j^{'} \cdot \mathbf{x}_k^{'}\right) / \|\mathbf{x}_k^{'}\|^2$
 9:         end for
10:         $\mathbf{x}_j^{'} = \nu \mathbf{x}_j^{'} / \|\mathbf{x}_j^{'}\|$
11:         $\mathbf{x}_j = \mathbf{x}_0 + \mathbf{x}_j^{'}$
12:         $\mathbf{x}_{j+1} = \mathbf{C}\mathbf{x}_j$
13:         $\mathbf{r}_j^{'} = (\mathbf{x}_{j+1} - \mathbf{x}_j) - \mathbf{r}_i$
14:         $\boldsymbol{\alpha} = arg\ min \|\mathbf{r}_i + \sum_{k=1}^{k=j} \alpha_k \mathbf{r}_k^{'}\ \|$
15:         $\xi = \|\mathbf{r}_i + \sum_{k=1}^{k=j} \alpha_k \mathbf{r}_k^{'}\|$
16:     end while
17:     $\mathbf{x}_0 = \mathbf{x}_0 + \sum_{k=1}^{k=j} \alpha_k \mathbf{x}_k^{'}$
18:     $i = i + 1$, $\mathbf{x}_1 = \mathbf{C}\mathbf{x}_0$, $\mathbf{r}_i = \mathbf{x}_1 - \mathbf{x}_0$
19: end while
```

The algorithm consists of two main loops, the inner of which is the GMRES loop trying to minimize the linear residual computed in line 15 according to Equation (4.40). The outer loop constitutes the nonlinear update performed in line 17 and given in Equation (4.33).

The first line forms the first residual $\mathbf{r}_0$ from the first guess $\mathbf{x}_0$ via the operator $\mathbf{C}$. As given in Equation (4.24), the evaluation of this operator constitutes one entire FSI-cycle. The nonlinear residuals are controlled in line 2 and tested against a given ϵ_0. Line 3 estimates the first linear residual which is tested against ϵ_1 in line 4. It is recomputed every linear step in line 15. The Krylov subspace is formed in line 6 and orthonormalized via a Gram–Schmidt procedure in lines 7 to 10. A test vector $\mathbf{x}_j$ is now constructed by line 11 against which the FSI-cycle is tested in line 12. The approximation of the matrix vector product by finite differences is computed in line 13 (see Equation (4.36)) and those α_i are found in line 14 which minimize the residual according to Equation (4.38). Line 17 constructs the update from a linear combination of the Krylov vectors and the α_i's.

Chapter 5

Benchmarks

All coupling setups presented in this thesis are compared with the Benchmarks described in this chapter. Two main benchmark tests were defined. The first one is of purely numerical nature and closely resembles the well known benchmark of a cylinder in a cross flow published in [133]. All members of the research group were asked to verify their coupling setups against this numerical benchmark and the results are about to be published.

The second benchmark is an experiment, against which a validation is attempted.

Both Benchmarks were defined as part of the efforts of the Research Unit 493 supported by the Deutsche Forschungsgemeinschaft to provide a database for verification and validation of existing numerical codes and newly developed methods.

The question of verification, validation, confirmation and benchmarking is more diverse than expected at first sight. For terminological details also concerning philosophical questions it is referred to the article of Oreskes et. al in Science [98]. In this thesis, the term verification is used to state that the developed code can reproduce results computed by other (and different) numerical methods or available analytical solutions of test problems. The term validation is used if the developed code can reproduce the results of physical experiments.

5.1 Numerical benchmark

5.1.1 Setup

The numerical benchmark is a two-dimensional setup and consists of a rigid cylinder in the flow of a Newtonian fluid with a flexible flag attached to its downstream side. It was already published in [132] but the configuration is restated here for completeness. The geometry of the setup is depicted in Figure 5.1 and Figure 5.2.

No-slip boundary conditions are applied to the top and bottom walls. The center of the cylinder is located at $C = (0.2[m], 0.2[m])$, its radius is defined as $r = 0.05[m]$, the elastic flag has a length of $l = 0.35[m]$ and a height of $h = 0.02[m]$, the bottom right-hand corner of the flag is positioned at $(0.6[m], 0.19[m])$ and the left end is clamped tight against the rigid cylinder. The deflection of the structure is measured at point A located on the right end of the symmetry axis of the elastic flag.

Note that the cylinder is positioned slightly off the symmetry axis of the flow domain. The Newtonian fluid has a density of $\varrho^f = 10^3[\frac{kg}{m^3}]$ and a kinematic viscosity of $\nu^f = 10^{-3}[\frac{m^2}{s}]$. The

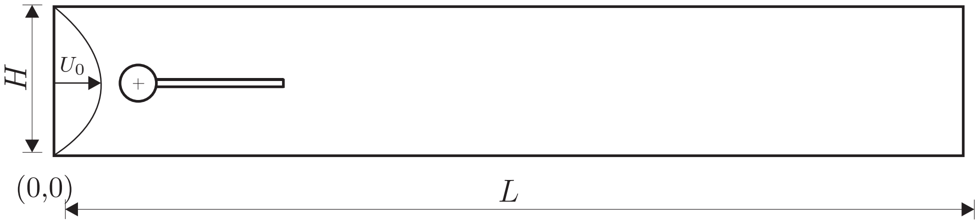


Figure 5.1: Computational domain with $H = 0.41[m]$ and $L = 2.5[m]$

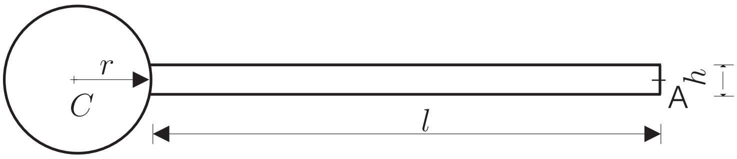


Figure 5.2: Detail: Cylinder with flag

inflow velocity is parabolic with a mean velocity $\bar{U}$ of either $0.2[\frac{m}{s}]$, $1.0[\frac{m}{s}]$ or $2.0[\frac{m}{s}]$, defining three different sub-Benchmarks FSI1, FSI2 and FSI3 with Reynolds numbers 20, 100 and 200. The density of the flag is $\varrho^s = 10^3[\frac{kg}{m^3}]$ in the cases of FSI1 and FSI3 and $\varrho^s = 10^4[\frac{kg}{m^3}]$ in the case of FSI2. Young's moduli of the St. Venant-Kirchhoff materials are $E^s = 5.6 \times 10^6[\frac{N}{m^2}]$ for case FSI3 and $E^s = 1.4 \times 10^6[\frac{N}{m^2}]$ for FSI1 and FSI2.

5.1.2 Principle behaviour and values of comparison

FSI1 leads to a stationary upward deflection of the flag tip. Stationary, because at $Re = 20$ no vortices develop in the lee of the cylinder and upward due to the asymmetry of the inflow profile w.r.t. the flag.
FSI2 leads to a large, periodic deflection of the flag. The periodic motion is mainly triggered by vortexes developing in the lee of the cylinder causing high and low pressure fields travelling along the flags surface. The movement is large (i.e. around 4.1 times the height of the flag), because the density of the flag is around ten times higher than that of the fluid. It is interesting to note that, were the flag stiff, it would act as a flow separator preventing the development of vortexes at this Reynold number.
Similar to FSI2, FSI3 leads to a periodic motion of the flag but with a smaller amplitude and higher frequency. FSI3 leads to vortex shedding even for a fixed flag.
The difficulties of computing these Benchmarks of course depend on the method chosen for its discretization but generally, the main hassle associated with each individual benchmark is:

- FSI1: to get a converged, steady state solution for the solid to fluid mass ratio of one
- FSI2: to be able to compute large deflections at the end point of the flag
- FSI3: to be able to resolve the boundary layers correctly at this higher Reynolds number.

In order to save computational time, all computations were started from a converged steady state fluid solution with a fixed flag and the flag was set free in the first step of the coupled computation.

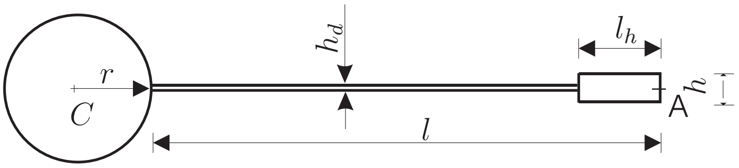


Figure 5.3: Numerical benchmark: Cylinder with flag

The quantities of comparison are the deflection at point A in amplitude and frequency. Additionally, drag and lift forces are evaluated by means of Equation (5.1)

$$(F_d, F_l)^T = \int_{\partial A} \mathbf{t}_s \, dl \tag{5.1}$$

where ∂A represents the wetted boundary.

5.2 Experimental benchmark

The exact definitions of the experimental benchmark were published in [57]. The most important data is recited here for completeness. The experimental benchmark is similar to the numerical benchmark as it consists of a cylinder with a flag attached to its downstream side. Additionally, the flag has a rectangular mass mounted at its tip in order to introduce more inertia into the system. Without it, the flag would not start to swivel. The flag is very thin and the cylinder is not clamped but only simply supported at its center in all directions. The cylinder can thus rotate freely around its axis. The structure is depicted in Figure 5.3. The cylinder is composed of aluminum with a density of $2828[\frac{kg}{m^3}]$, the membrane like flag and the rear mass is made of stainless steel with a density of $7855[\frac{kg}{m^3}]$ and $7800[\frac{kg}{m^3}]$, respectively. While flag and cylinder can be considered rigid, the Young's modulus of the flag was measured to be $E = 2.0 \times 10^{11}[\frac{N}{m^2}]$.

The fluid domain is depicted in Figure 5.4. Its dimensions are: $L = 0.338[m]$, $l = 0.272[m]$, $H = 0.240[m]$, $h = 0.170[m]$. The solid box is the fluid domain with its physical boundaries, while the dashed box represents the domain at which measurements were taken. Much like in the numerical benchmark, the fluid is a viscous, incompressible polyethylene glycol syrup with a high kinematic viscosity of $\nu = 1.64 \times 10^{-4}\frac{m^2}{s}$ resulting in low Reynolds numbers (i.e. laminar flows) at fairly high flow speeds. The fluid density was measured to be $1050[\frac{kg}{m^3}]$. Two benchmark configurations were provided which differ only in the magnitude of the inflow velocity. "Expemiment 1" has a inflow velocity of $U = 1.07[\frac{m}{s}]$ and "Expemiment 2" of $U = 1.45[\frac{m}{s}]$ resulting in Reynolds numbers of $Re = 140$ and $Re = 190$. The inflow profile is mainly constant across the inflow domain, however the results are sensitive to variations in U, which is why it is recommended to directly impose the measured values available from [57]. The walls in the positive and negative y-direction can be considered as no slip walls and the outflow as a "do nothing" boundary.

Gravity acts on fluid and structure alike in the direction of the positive x-axis. Difficulties in computing this benchmark again depend on the method chosen for its discretization but the main points are:

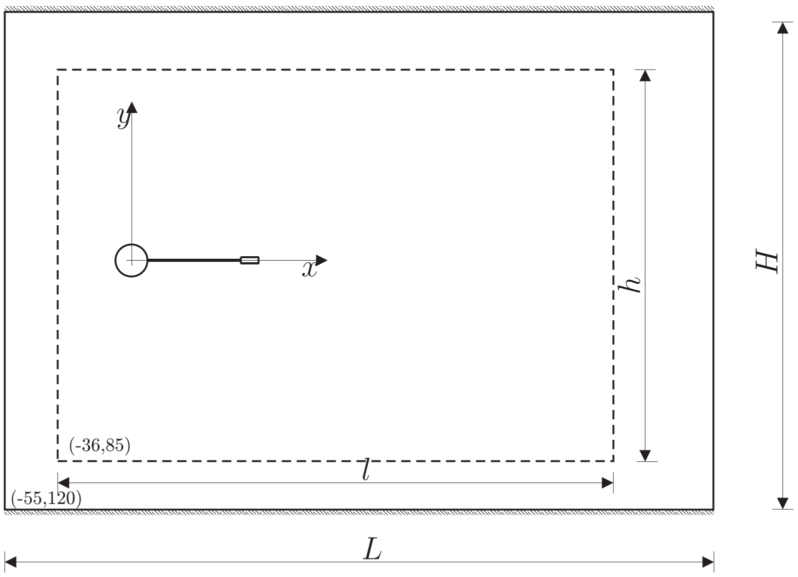


Figure 5.4: Numerical benchmark: Fluid domains, the dashed domain denotes the region in which measurements were conducted.

- the large deformations at the end of the flag and its rapid change in angle as it reverses its general direction.
- the thin structure and the boundary layers, which need to be resolved for an accurate computation.

Chapter 6

Coupling to the Finite Volume Method

6.1 Motivation

The main goal of the work documented in this Chapter was to test the suitability and accuracy of structural p-FEM in FSI against the Benchmarks presented in Chapter 5.1. This is why CFX[2] was chosen as a well known, verified and validated commercial fluid solver. Moreover, the accuracy of the force and velocity transfer methods suggested in Chapter 4.2.2 were to be investigated in a FSI computation. The force transfer was performed via composed integration. For the intersection of the meshes the tool developed in [123] was used. The displacements were evaluated at the footprint of the nodes of the fluid mesh on the structural boundary and directly transferred without performing a global L_2 projection. The results are presented in Chapter 6.2.

Once these goals were achieved, the convergence within the FSI cycles was to be accelerated. Explicit methods diverged after only a few time steps. For this reason, two implicit coupling procedures were employed, a Block Gauss-Seidel procedure with Aitken underrelaxation as presented in Chapter 4.3.2.1 and the Interface-GMRES Method presented in Chapter 4.3.2.2. For the Interface-GMRES method, results have only been published for small "toy" examples and compressible fluid solvers. The main concern was that the method would have drawbacks for larger systems with more interface degrees of freedoms. It was found that this is not the case. Surprisingly, the drawbacks lie elsewhere. They are layed out in detail in Chapter 6.2.1.

6.2 Testing against Benchmarks

The setup was tested against the benchmark configurations presented in Chapter 5.1. The benchmark proposed in [132] is a two-dimensional configuration. The coupling environment created for coupling to CFX [2], however, is realized in a three dimensional setting. Therefore, the two-dimensional setup was imitated by expanding the fluid domain by 0.005[m] (corresponding to $\frac{1}{4}^{th}$ of the height of the flag) into the z-direction and application of symmetry boundary conditions on the resulting front and back slices.

The fluid was discretized with 58 368 hexahedral elements as implemented in CFX 10 [2]. The time domain was resolved with a backward Euler method, while the structural discretization in time was realized via the Newmark method. The extremely soft material of the structure forced the use of a parameter set of $\beta = 0.49$, $\gamma = 0.9$ to damp out numerical instabilities

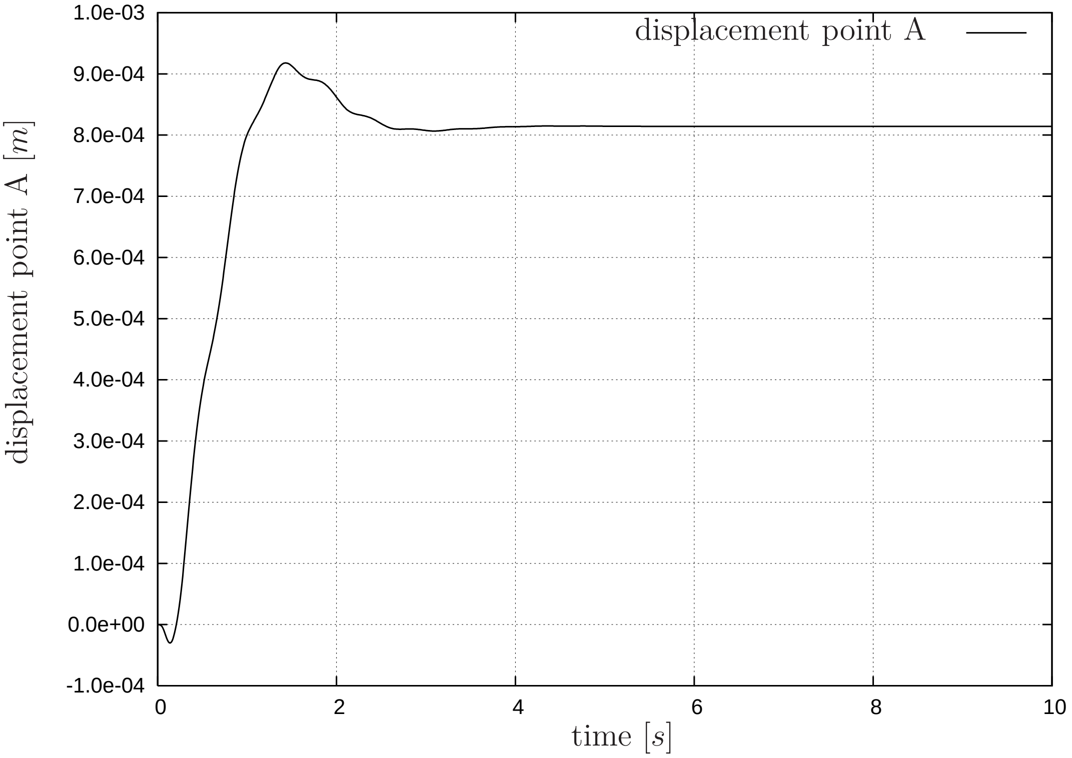


Figure 6.1: Displacement of point A

of the structural time integration. The structure was computed geometrically nonlinear. For fluid and solid alike, a timestep of $0.01[s]$ was chosen.

The deflection over time is depicted in Figure 6.1.

Figure 6.2 depicts the resulting pressure field at the steady state (time $t = 10[s]$) within the domain and shows the 3-D setup of the thin domain. The mass ratio $\frac{fluid}{structure}$ is equal to one. In combination with the relatively low Young's modulus, this leads to divergence of the sub iterations within the implicit coupling used in this setup. Therefore, a Gauss-Seidel type scheme as described in Chapter 4.3.2.1 was chosen in which the forces needed to be relaxed by a constant factor of $\omega_i = 0.15$ to achieve convergence at every time step.

Figure 6.3 shows the structural domain. It is pointed out that only three high order hexahedral elements as described in Chapter 2.2 are needed to obtain an accurate result. The polynomial degrees were chosen to be 7, 4 and 1 for all displacement components in x,y and z-direction respectively. This results in an overall discretization of 252 degrees of freedom when applying the *trunk space* $\mathcal{S}_{\mathrm{ts}}^{p_\xi,p_\eta,p_\zeta}(\Omega_{\mathrm{st}}^{\mathrm{h}})$. For a detailed view, the upper half of the structural domain and the intersection with the fluid mesh is shown in Figure 6.4.

Figure 6.5 depicts the relative and absolute force error of each time step. The limiting factor here is the precision within the commercial finite volume fluid solver CFX. The computation of the boundary forces only has single precision accuracy. The relative force error is below $1.0 \times 10^{-3}[\%]$ while the absolute force error is below $1.0\text{x}10^{-5}[N]$. Thus, the force transfer is as accurate as possible within this framework.

The asymmetric setting of the cylinder with respect to the inflow profile leads to a lift force on the flag. After a settling time of about 6 seconds, a steady state displacement of $8.141 \times 10^{-4}[m]$

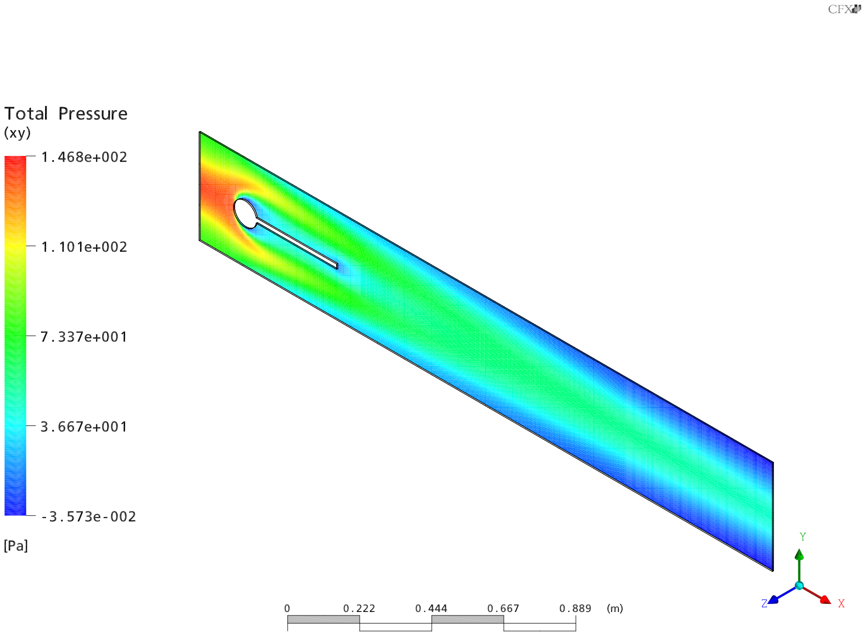


Figure 6.2: Pressure distribution at cutting plane xy

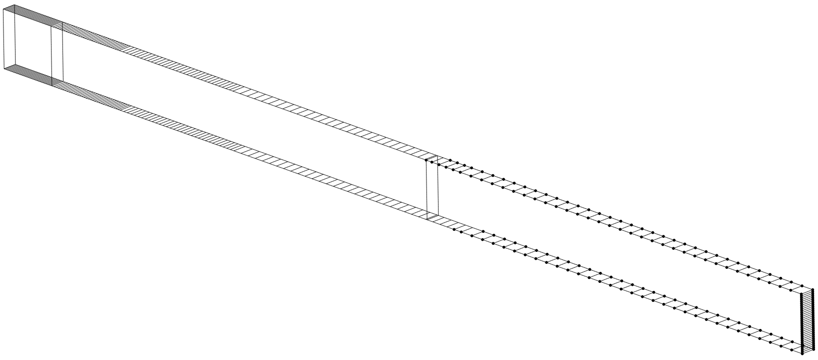


Figure 6.3: Intersection of fluid and structural meshes

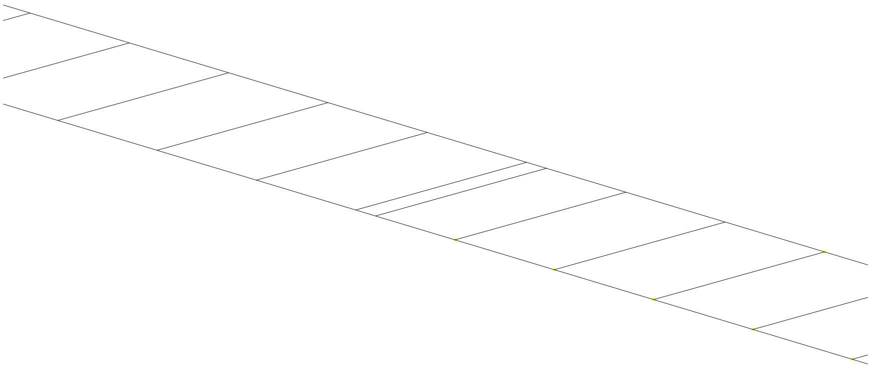


Figure 6.4: Detail of the intersected meshes

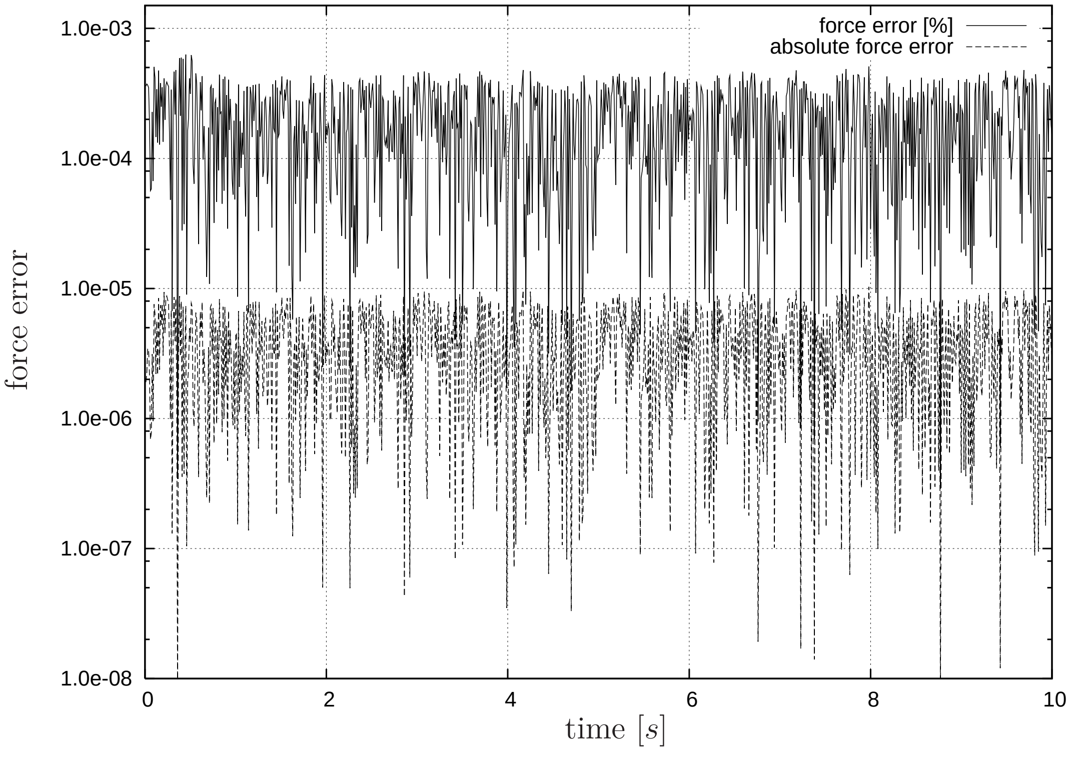


Figure 6.5: Force transfer error

developed at point A. This is only 0.8[%] off the results obtained by [132] which state a deflection of $8.209 \times 10^{-4}[m]$ as their reference value. The results are summarized in Table 6.1

Case	$u_x(A)[m]$	$u_y(A)[m]$	F_d *[N]*	F_l *[N]*
FSI1	2.15×10^{-5}	8.141×10^{-4}	14.287	0.7640
FSI1$_{ref}$	2.27×10^{-5}	8.209×10^{-4}	14.295	0.7638
[%] deviation	5.28	0.82	0.056	0.445

Table 6.1: FSI1: Results obtained and comparison with reference [132]. F_d and F_l are computed according to Equation (5.1) whereby ∂A is the wetted boundary of the flag plus the wetted boundary of the cylinder.

These results were encouraging at the time, however, neither test cases FSI1 nor FSI2 could be computed since the demand on the mesh deformation was too high. Overlapping meshes resulted in a premature breakdown of the computations. It is referred to Chapter 8.1 at this point for further discussion on mesh deformation problems.

6.2.1 A critical view on Interface-GMRES

The Interface-Generalized Minimal RESidual algorithm (I-GMRES) by Michler [92] was layed out in Chapter 4.3.2.2 and showed promising performance for the examples discussed in [92]. It was thus chosen to be implemented in CFX and tested against the FSI1 Benchmark. In order to be able to compare the performance of I-GMRES to the widely used Gauss-Seidel method all results in this Chapter were computed with identical initial conditions. The initial condition was a converged stationary fluid solution with a fixed flag and no prestress or predefined deflection in the structure. In all cases, the time step size was chosen to be $\Delta t = 0.01[s]$, and measurements were made at the first timestep.
First, the performance of I-GMRES with compressible fluids was tested where the fluid was given a speed of sound of $c = 10\frac{m}{s}$. The convergence history of the first timestep is depicted in Figure 6.6. The discrete L_2 norm of the residual is plotted against the number of interfield iterations[1]. The line named “Gauss-Seidel” denotes the results obtained by a Gauss-Seidel iteration according to Chapter 4.3.2.1 with a constant relaxation factor of $\omega_i = 0.3$ which has been chosen in advance by trial and error to lead to the best possible convergence rate for this timestep. After 21 iterations, an error reduction at the interface of approximately three orders, namely from 1.0×10^{-5} to 1.0×10^{-8} is obtained.
The curves of the convergence of I-GMRES in Figure 6.6 are termed “I-GMRES: reduction of linear residual by $< X >$ order” where “$< X >$” is either one, two or three and refers to how much the linear residual ξ (see Algorithm A-1 Line 15) is reduced before the update of the linearization state $\mathbf{x}_0$ is performed in line 17 of Algorithm A-1 (i.e. $\mathbf{x}_0 = \mathbf{x}_0 + \sum_{k=1}^{k=j} \alpha_k \mathbf{x}_k^{'}$). Following these curves from the left to the right in Figure 6.6, this update manifests itself by

[1] i.e the number of complete FSI cycles equivalent to the number of evaluations of the operator $\mathbf{C}$ defined in Equation (4.24).

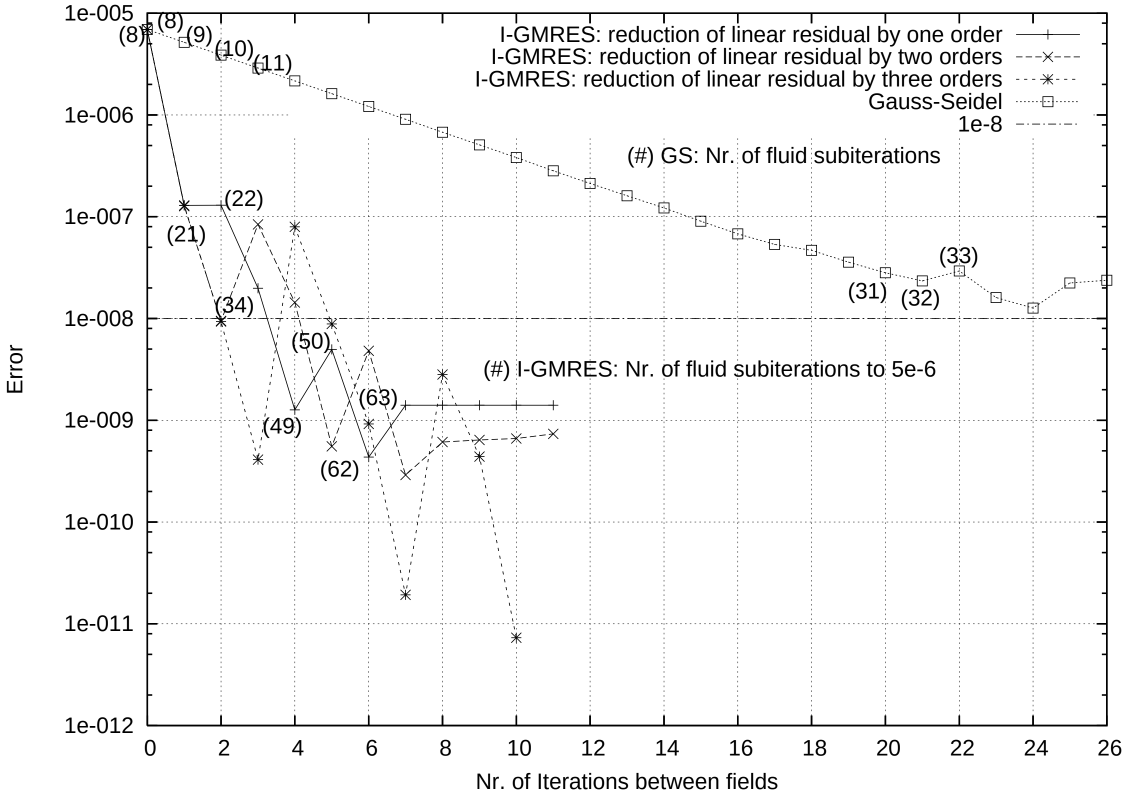


Figure 6.6: Compressible FSI1: Gauss-Seidel vs. I-GMRES: reduction of interface residual

the sudden jump to a larger error. The hull of these jumps corresponds to the true, nonlinear residual comparable to the ones obtained by a Gauss-Seidel iteration, while the intermediate lower values represent the linear residuals. As depicted in Figure 6.6, the error at the interface can be reduced by approximately four orders with I-GMRES; one order more than with the GS iterations.

At first glance, the performance w.r.t. the FSI-cycles needed shows an astonishing improvement, since a reduction of the interface error by three orders costs 24 iterations[2] with Gauss-Seidel compared to only five iterations between the fields with I-GMRES (see line “I-GMRES: reduction of linear residual by one order”). This, however, is an unfair comparison since the number of evaluations of the operator **C** only counts part of the effort. A more concise measure is to count the amount of cycles needed for the field solvers to seek equilibrium within the overall iteration process. The structural solver required around three cycles for both Gauss-Seidel and I-GMRES which are of minor importance in a direct comparison of efficiency. Therefore, the number of iterations the fluid solver itself needs to converge is considered a good measure and is thus given in brackets in Figure 6.6 for the case causing the least effort. The fluid solver needs 35 iterations if coupled to the structure by the Gauss-Seidel method and 50 for I-GMRES. It is noteworthy that, although at first glance I-GMRES is faster, a closer look reveils that I-GMRES is around 1.5 times slower for this configuration.

Since most of the effort is spent in the fluid solver, the next question to answer is if the fluid solver really needs to iterate down to an error of 5.0×10^{-6}. The study in Figure 6.7 shows

[2] i.e. evaluations of the operator **C**.

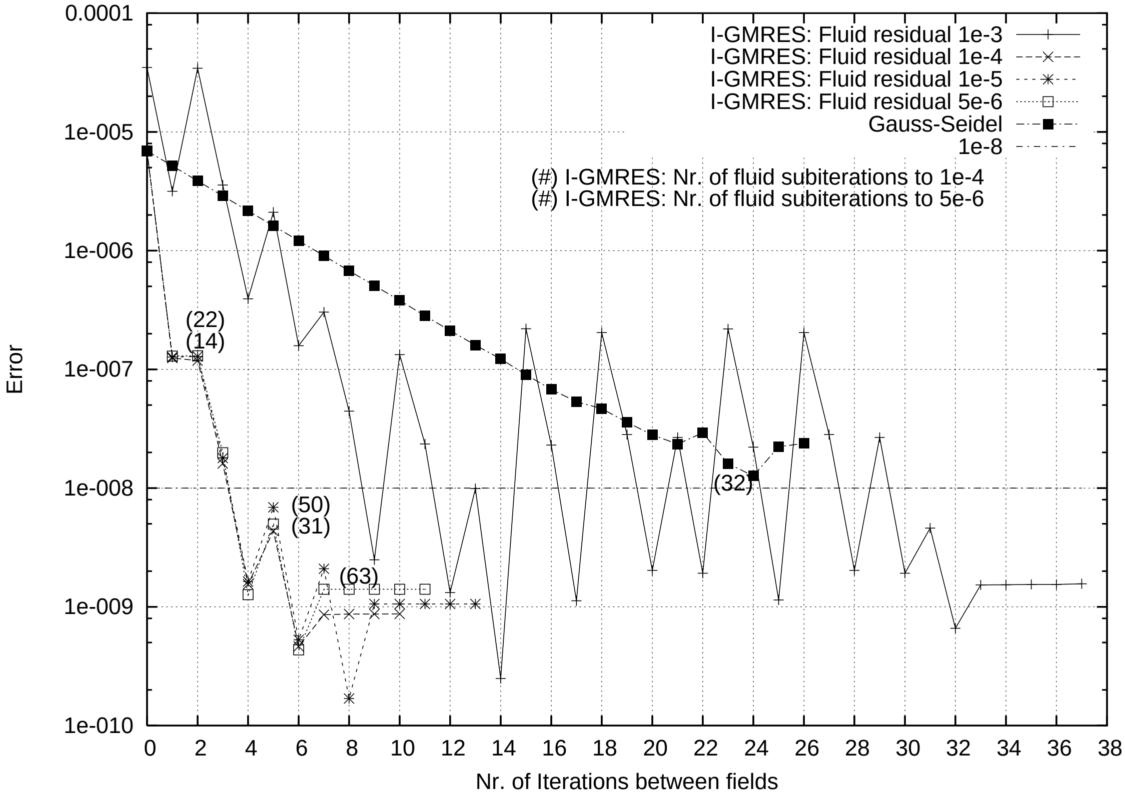


Figure 6.7: Compressible FSI1: Gauss-Seidel vs. I-GMRES: Different fluid residuals

that the fluid solver needs to compute its subproblem to an accuracy of at least 1.0×10^{-4} as otherwise the convergence of the overall algorithm deteriorates overproportionally. This brings the necessary iterations within the fluid field solver down to 31 as indicated by the numbers beside the corresponding curve. This is one iteration less than in the Gauss-Seidel approach.

The reason for the difficulty of the flow solver to converge lies in the orthogonalization of the Krylov base vectors $\mathbf{x}_j'$ in lines 7-10 of Algorithm A-1. These Krylov base vectors have a clear interpretation. They represent the boundary configuration against which the fluid solver is tested in line 12 of Algorithm A-1[3]. Figure 6.8 shows these vectors constituting the first two steps of a second order reduction of the linear residual of Figure 6.6. The two Krylov vectors correspond to the strongly bent curved boundaries. The third vector, corresponding to curve “update” depicts the linear combination of these vectors as needed for the update of the linearization state in line 17, namely $\mathbf{x}_0 = \mathbf{x}_0 + \sum_{k=1}^{k=j} \alpha_k \mathbf{x}_k'$. All three vectors have been magnified by a factor of ten for better visualization. The crucial point is, that the fluid solver is tested against orthogonal boundary conditions, i.e. boundary conditions which have as little in common as possible. This makes it extremely difficult for the fluid solver to converge as it needs to run into completely opposite directions each time.

The factor ν in line ten of Algorithm A-1 can be chosen to scale the magnitude of the Krylov vectors. As a matter of fact, it must be chosen smaller than one (in this case $\nu = 0.005$) for the inner iterations to converge. Choosing the value of ν is much like choosing a proper

[3] shifted by the current linearization state $\mathbf{x}_0$ in line 11.

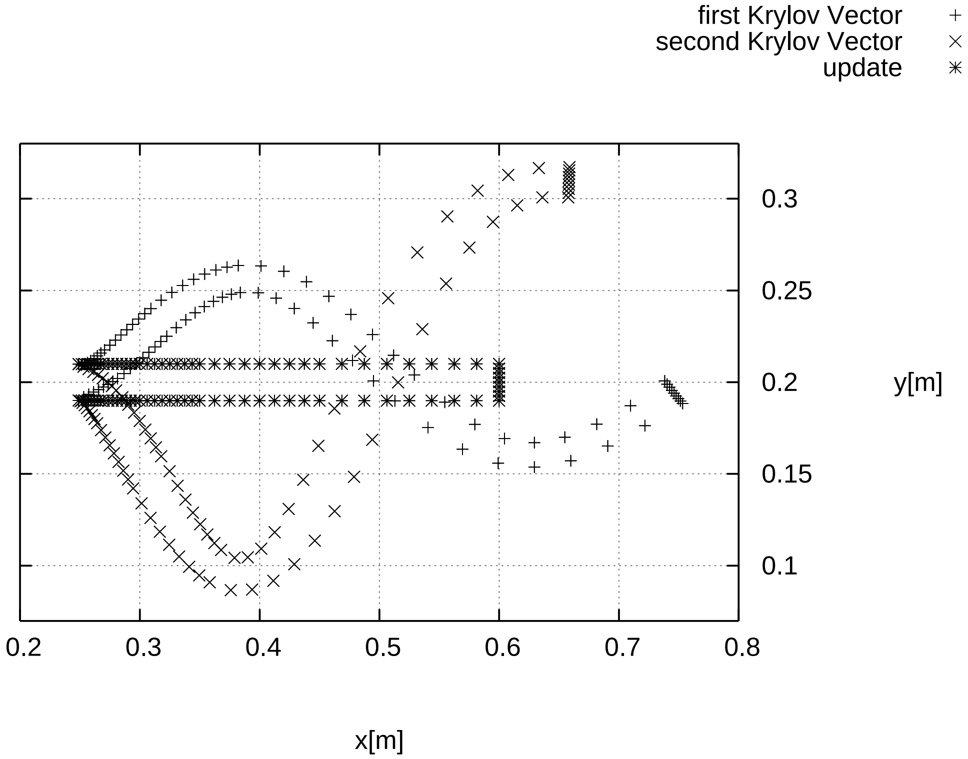


Figure 6.8: Compressible FSI1: I-GMRES: Krylov Vectors forming the second linear update in Figure 6.7 corresponding to fluid residual 1.0×10^{-5}. All vectors scaled by factor 10 for better visibility.

epsilon for a numerical differentiation. The value may not be too small as not to run into numerical round off errors and not too large as to capture the true tangent in a highly nonlinear problem. In this case, the lower bound is around $\nu = 0.005$ as CFX only computes with single precision numbers[4] and a further reduction leads to round off errors causing a breakdown of the algorithm.

The situation is completely different in the Gauss-Seidel method, where the interface position (i.e. vectors) slowly approaches the equilibrium position from one side instead of being orthogonal to each other. A selection is depicted in Figure 6.9. This is the reason why the fluid solver needs to perform only one iteration to converge for each FSI cycle.

One might think of dropping the orthogonalization performed in lines 7-10 altogether, but this is not possible as otherwise the basis vectors $\mathbf{x}'$ spanning this space were not orthogonal. Deleting lines 7-10 would quickly lead to a breakdown of the method, as no new linearly independent vector could be found by the minimalization procedure carried out in line 14. Then (a) the linear residual ξ will not diminish further at some point and (b) a premature

[4] even though there are double precision versions of it, but that is only related to a compiler option and does not lead to true double precision in all routines.

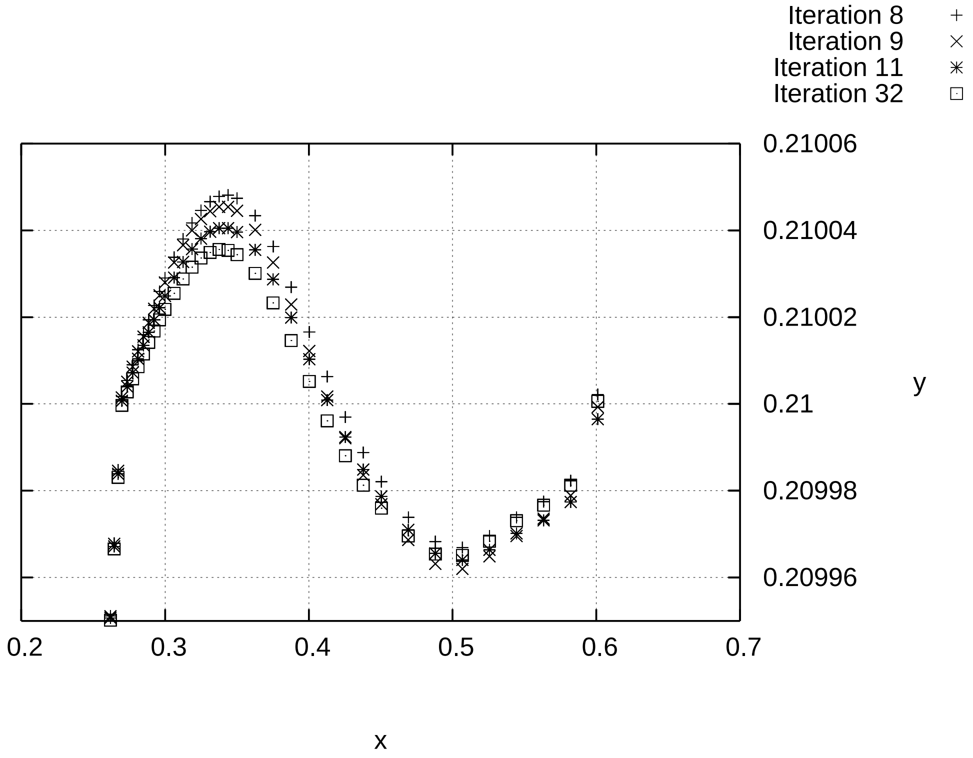


Figure 6.9: Compressible FSI1: Gauss-Seidel: Convergence history of the interface positions of the upper boundary of the flag.

update performed in line 17 would lead to no improvement of the linearization state $\mathbf{x}_0$.

Up to this point, all results were obtained by coupling to a compressible fluid. Sadly, the situation worsens when coupling to *in*compressible fluids because it is much harder for the incompressible solver to satisfy the orthogonalized boundary conditions. Additionally, the bandwidth of numerically stable choices for ν becomes very small. At some point, either the I-GMRES breaks down for lack of orthogonality of the boundary conditions or the fluid solver can not converge. This makes it virtually impossible to choose a value of ν in advance. Figure 6.10 shows the convergence curves obtained by coupling to the incompressible FSI1 configuration.

Interface-GMRES, however, does perform much better than a plain underrelaxed Gauss-Seidel iteration for small structure/fluid density ratios. The convergence history depicted in Figure 6.11 shows the convergence history of FSI1 where the structural mass has been lowered by a factor of ten leading to a mass ration of solid/fluid of 0.1. While Gauss-Seidel almost diverges and causes the fluid solver to perform more than 250 iterations, I-GMRES only demands around 85 iterations from the fluid solver. These results were obtained with the compressible configuration, though. The incompressible configuration lead to divergence for both iteration methods.

These examples require no extra fonts:

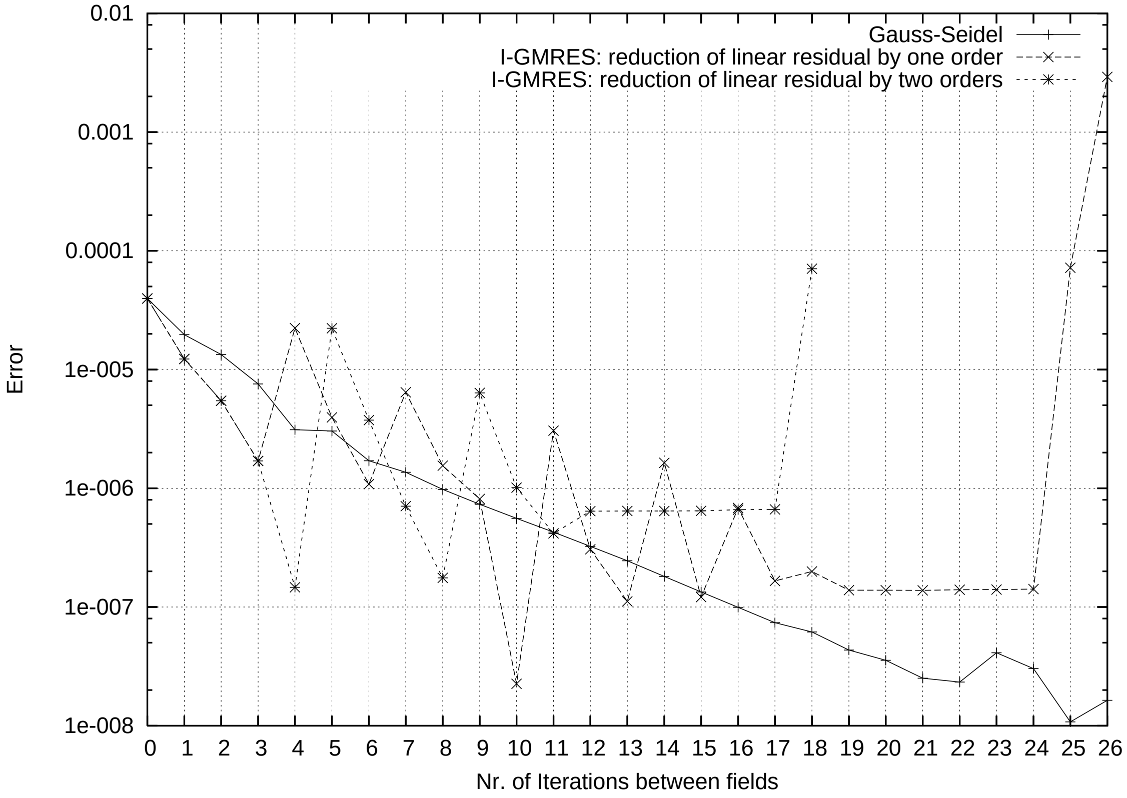


Figure 6.10: Incompressible FSI1: Gauss-Seidel vs. I-GMRES: reduction of interface residual

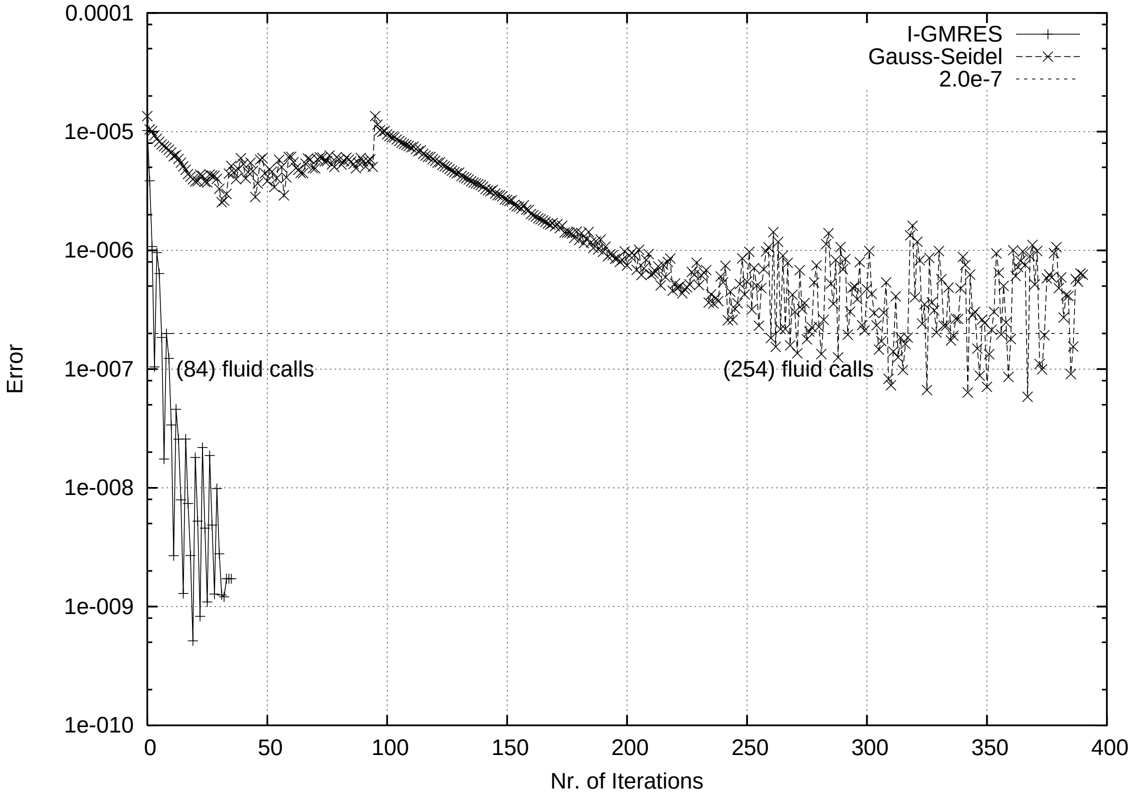


Figure 6.11: FSI1: Gauss-Seidel vs. I-GMRES, mass ratio 0.1: reduction of interface residual

In [92] it is pointed out that an increase in efficiency can be obtained by reusing the Krylov-vectors for the next timestep. This was implemented but deteriorated the convergence properties so severely in the subsequent time steps that no further studies were undertaken.

Chapter 7

Coupling to the Spectral Element Method

7.1 Motivation

The Spectral Element Method as discussed in Chapter 3.3 and the p-FEM layed out in Chapter 2.2 have the same roots. Actually, these methods are identical. Differences are only of terminological importance as "spectral" implies the use of *very* high orders and smoothness. This difference merely has historical interest. Both attempt to approximate the unknown function and its derivatives using piecewise polynomials of high order. These methods can, therefore, provide exponential convergence rates under the premise of the existence of sufficiently smooth solutions to the considered problem. For structures, this has been shown e.g. in [125, 126] and for fluids in [16, 73]. The goal to carry these benefits over to a partitioned FSI computation, is yet difficult to achieve.

Only few previous publications exist using a partitioned coupling of these discretizations in an FSI context [8, 112, 74, 10]. They are concerned with explicit, i.e. loosely coupled schemes or they suggest suitable traction and displacement transfer methods for this type of discretizations.

However, due to the incompressible nature of the benchmark problems, loose coupling methods diverged after only a few time steps. Therefore, an overall implicit coupling algorithm needed to be employed. For this purpose, the implicit stiffly stable time integration scheme for incompressible fluids layed out in Chapter 3.3 was utilized. The overall coupling algorithm including this scheme is presented in Chapter 7.1.1. It has not been published before but is the core result of the work jointly carried out in the framework of the master thesis of Papaioannou[99].

The feasibility of the method was tested against a driven cavity problem in Chapter 7.2.1 and the methods showed good results. The limits of the method became visible when tested against the Benchmarks defined in Chapter 5.1.

7.1.1 Coupling algorithm

The coupling is achieved by extending the fluids implicit time integrator to account for the structural solution, where the boundary conditions for both solvers are updated in each subiteration. The iterative time-stepping scenario for advancing the entire FSI system from time level n to time level $n+1$ is as follows (For notations see Chapter 3.3):

Start with a predictor for the fluid solution (e.g. of zeroth order):

$$(\mathbf{v}^{n+1}, \mathbf{w}^{n+1}, \mathbf{p}^{n+1})^0 = (\mathbf{v}^n, \mathbf{w}^n, \mathbf{p}^n) \tag{7.1}$$

Begin sub-iteration step $j+1$:

Update the fluid solution for the current iteration:

$$\frac{1}{\Delta t}\left(\tilde{\mathbf{v}}_i - \sum_{q=0}^{J_i-1} \alpha_q \mathbf{M}\mathbf{v}_i^{n-q}\right) = -\mathbf{N}(\mathbf{v}^{(n+1,j)}, \mathbf{w}^{(n+1,j)}) + \mathbf{F}^{n+1} \tag{7.2}$$

$$\frac{1}{\Delta t}\left(\gamma_0 \mathbf{x}_i^{(n+1,j+1)} - \sum_{q=0}^{J_x-1} \alpha_q \mathbf{x}_i^{n-q}\right) = \mathbf{w}_i^{(n+1,j)} \tag{7.3}$$

$$\begin{aligned}\frac{\partial p^{(n+1,j+1)}}{\partial n} &= -\mathbf{n}\cdot \\ &\left[\frac{\partial \mathbf{v}^{(n+1,j+1)}}{\partial t} + \mathbf{N}(\mathbf{v}^{(n+1,j)}, \mathbf{w}^{(n+1,j)}) + \nu[\nabla \times (\nabla \times \mathbf{v}^{(n+1,j)})]\right]\end{aligned} \tag{7.4}$$

$$\mathbf{L}\mathbf{p}^{(n+1,j+1)} = \frac{1}{\Delta t}\mathbf{D}_i \tilde{\mathbf{v}}_i \tag{7.5}$$

$$(\gamma_0 \mathbf{M} - \Delta t \nu \mathbf{L})\, \mathbf{v}_i^{(n+1,j+1)} = \mathbf{M}\tilde{\mathbf{v}}_i + \mathbf{D}_i^T \mathbf{p}^{(n+1,j+1)} \tag{7.6}$$

$$\mathbf{L}\mathbf{w}_i^{(n+1,j+1)} = 0 \tag{7.7}$$

Compute the structural force vector $\mathbf{F}^{(n+1,j+1)}$ by integrating the fluid tractions $t_j|_S^{(n+1,j+1)} = (\tau_{jk}|_F n_k)^{(n+1,j+1)}$ on the wetted surface:

$$\mathbf{F}^{(n+1,j+1)} = \int\limits_{\Gamma_{FS}^S} (\mathbf{t}|_S^{(n+1,j+1)})^T \mathbf{N} \mathrm{d}\Gamma \tag{7.8}$$

Update the structural solution for the current iteration, by solving:

$$\mathbf{M}\ddot{\mathbf{u}}^{(n+1,j+1)} + \mathbf{C}\dot{\mathbf{u}}^{(n+1,j+1)} + \mathbf{K}\mathbf{u}^{(n+1,j+1)} = \mathbf{F}^{(n+1,j+1)} \tag{7.9}$$

applying the Newmark method.

Compute the new fluid and mesh velocity at the interface for the next iteration, by solving the following weighted residual equation:

$$\int\limits_{\Gamma_{FS}^F} (\mathbf{v}_F - \dot{\mathbf{u}}|_S)\delta \mathbf{v} \mathrm{d}\Gamma = 0 \tag{7.10}$$

If the variance of the solution is sufficiently small, proceed to the next time step. Otherwise, continue the sub-iterative procedure.

7.1.2 Transfer of tractions

The right-hand side of Equation (7.8) involves the integration of the product of two functions defined in different functional spaces. The fluid traction vector is defined on Γ_{FS}^{F}, i.e. the fluid side of Γ_{FS}, in a space spanned by the spectral element shape functions $\mathbf{\Phi}$, which are locally defined at the elements of the fluid mesh. On the other hand, the matrix $\mathbf{N}$ contains the structural shape functions defined on the structural mesh footprint on Γ_{FS}^{S}, i.e. the structural side of Γ_{FS}.

There are several concepts of how to deal with this problem in the general case, i.e. when fluid and structural meshes do not match. Traction transfer methods and relevant literature concerned with this situation are discussed in Chapter 4.2.2 and Chapter 4.2.3. For the examples computed in Chapter 7.2 difficulties are avoided by matching the fluid and the solid meshes at their interface. The tractions at the fluid boundary can then be directly evaluated at the Gaussian points of the structure. This is equivalent to an L_2 projection of the fluid forces onto the structural boundary.

7.1.3 Transfer of Velocities

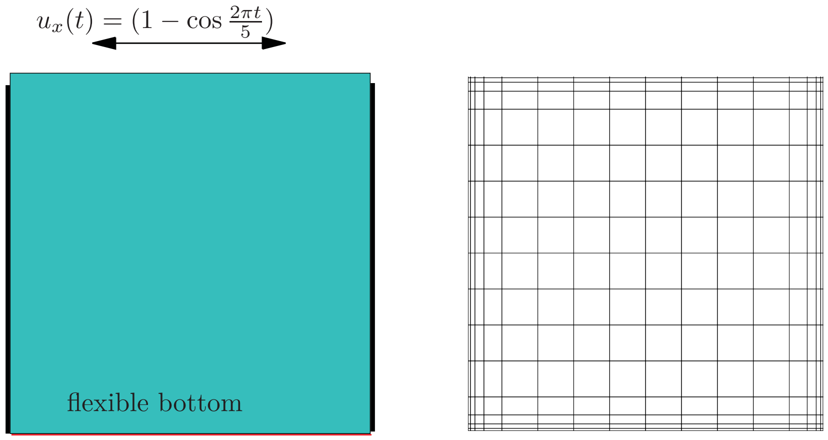
Similarly, the structural solver evaluates the velocities at the Gauss-Lobatto points of the fluid solver. The fluid solver then solves Equation (7.10) by applying a $H^{1/2}$ projection as described e.g. in Chapter 4.2.4, in order to compute the modal coefficients of the velocity on the interface.

7.2 Testing against Benchmarks

7.2.1 Driven Cavity

Before testing the setup against the Benchmarks proposed in Chapter 5 it was tested against the incompressible driven cavity problem equipped with a flexible bottom. It is depicted in Figure 7.1a.

This problem was first proposed in [140] and further investigated in [45]. Although it is a relatively simple setup in terms of its geometry, it is prone to the artificial added mass effect. It leads to catastrophic instabilities if the mass ratio is equal to one for any weakly coupled, staggered solution procedures regardless of how small the time step is chosen [45]. This qualifies it for testing the suitability of the implicit coupling setup proposed in this thesis. The fluid domain is a square of dimensions $1\,m \times 1\,m$ and the structural thickness is with $t = 0.002\,m$ very thin compared to the height of the entire domain. The oscillation of the flexible bottom is enforced by imposing a periodic horizontal velocity with a period of $5\,sec$ at the top boundary of the fluid domain, as shown in Figure 7.1a. The structure is clamped only at its edges with no support elsewhere allowing for a free movement throughout its span width due to the tractions applied by the fluid. The material properties for both structure and fluid are given in Table 7.1. The fluid mesh was constructed by applying a constant element size and refining near the edges of the domain, to cope with the singularities occurring due to the boundary conditions. The fluid mesh consisting of 16×16 elements is shown in Figure 7.1b. The structural mesh consists of 16 2D plain strain elements, such that its footprint on the interface coincides with the footprint of the fluid mesh. The structural as well as the fluid



(a) Geometry and boundary conditions

(b) Fluid spectral element mesh

Figure 7.1: Geometry, boundary conditions and discretization of the driven cavity with a flexible bottom

fluid	structure
density $\rho^F = 1.0\,[kg/m^3]$	density $\rho^S = 500\,[kg/m^3]$
viscosity $\nu^F = 0.01[m^2/s]$	stiffness $E = 2.5 \times 10^8\,[N/m^2]$
	Poisson's ratio $\nu^S = 0.0$

Table 7.1: Material parameters of driven cavity example with flexible bottom

discretization where chosen with a polynomial order of five.

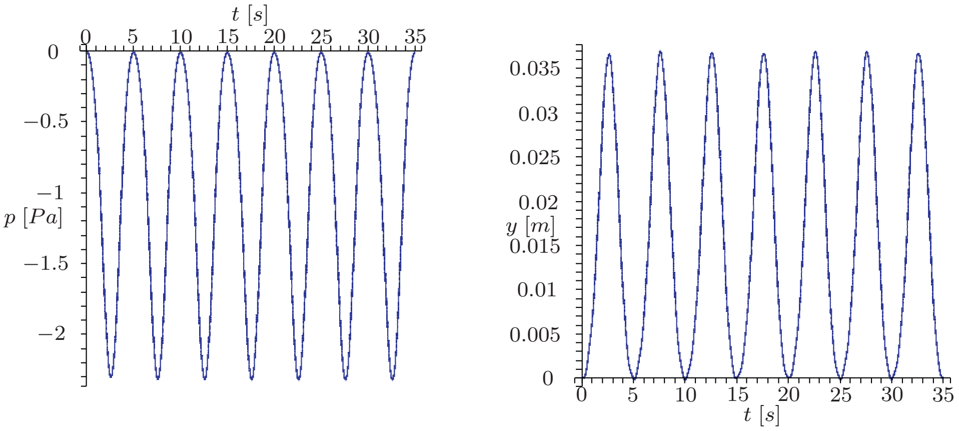
The coupling was carried out by implementing the scheme presented in Chapter 7.1.1, with a first order integration of the diffusion terms and mesh position ($J_i = J_x = 1$). It was assumed for simplicity that the fluid tractions on the interface consist only of the pressure. As the displacements are very small, the pressure is assumed to act entirely in the vertical direction. Hence, the structural loading on the interface is computed at each coupling step applying the resulting tractions directly at the Gauss points of the structure in the y-direction. The structural computation was performed by applying the average acceleration method, i.e. the special case of the Newmark method for $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$.

To achieve a compromise between high accuracy and a low Peclet number[1], the time integration step was chosen to be $\Delta t = 0.001[s]$. To avoid instabilities at the structure due to the excitement of modes with a period smaller than the time-step, a stiffness proportional damping with $\delta = 0.00032$ was applied resulting in a damping of frequencies lower than the time step. In order to achieve convergence acceleration and thus decrease the computational cost, the Aitken-like iteration method presented in Chapter 4.3.2.1 was applied to the transfer of the pressure from the fluid to the structural side of the FSI interface.

The imposed periodic horizontal velocity at the top boundary of the fluid domain causes the flexible bottom to oscillate. To monitor this oscillation, the pressure and vertical position at

[1] The Peclet number characterizes the relative importance of convective to diffusive effects in a given flow problem. Its definition is given e.g. in [30, 65]. It is nicely shown in [30] that a high Peclet number is unfavourable for pure Galerkin approximations of which the SEM used here is a representative.

the middle node of the flexible bottom were measured. The results are depicted in Figures 7.2a and 7.2b, respectively.



(a) Pressure (b) Vertical position

Figure 7.2: Pressure and vertical position at the middle of the flexible bottom

As may be observed in both graphs, the obtained response is periodic with a *constant* amplitude[2]. The attempt to solve the specific example applying a weak coupling scheme led to an unstable solution within the first 100 time-steps, thus not being able to complete the first period of oscillation. In Figure 7.3, a contour plot of the Euclidean velocity norm: $\|\mathbf{v}\|_2 = \sqrt{v_x^2 + v_y^2}$ is shown at different time instants, describing the first period of the systems time history. The plots are asymmetric because of the small vortices developing due to the nonlinear convection at the two sides of the top part of the fluid domain, close to the boundaries. In addition, the movement of the flexible bottom can also be seen. The fluid is applying a negative pressure to the structure, causing it to move in an inward direction.

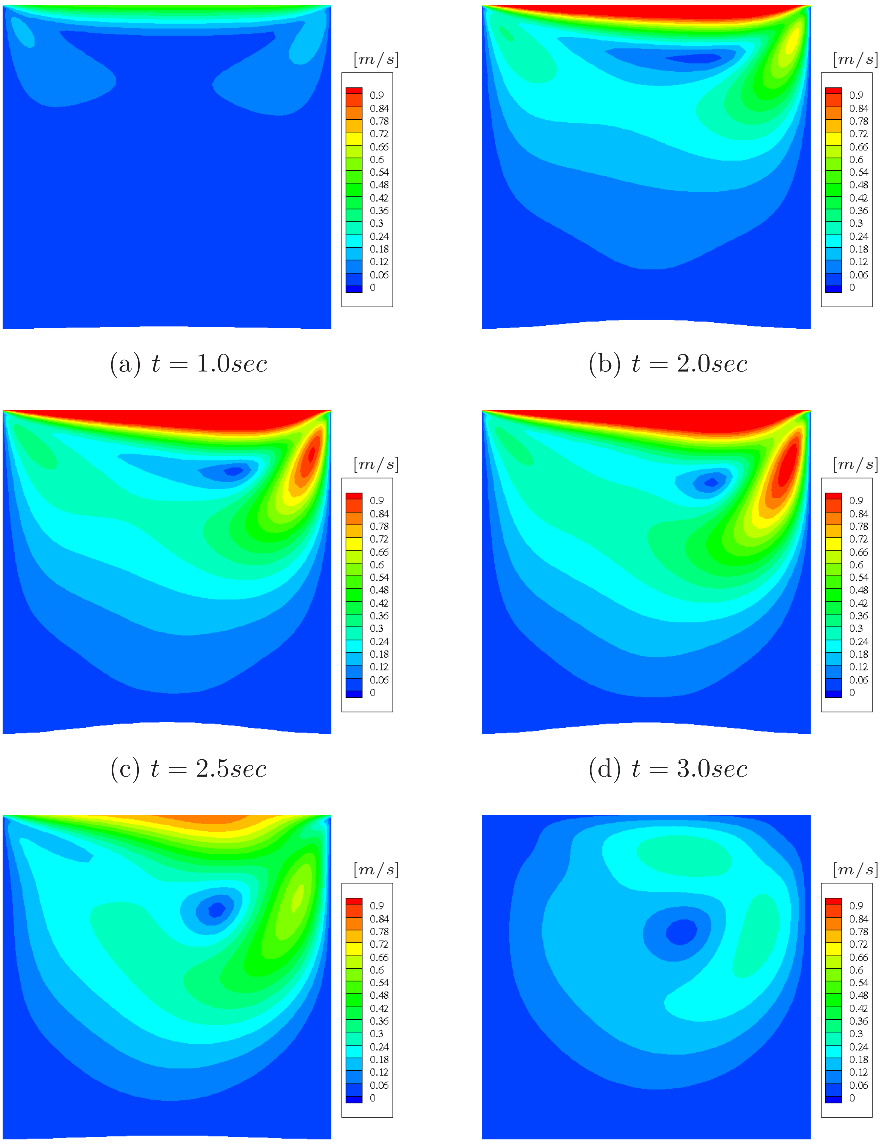
7.2.2 DFG Benchmark

The performance was tested against FSI1 as defined in Chapter 5.1. Only pressures at the boundary are transferred to the structural boundary and the structure is computed geometrically linear[3]. The fluid and structural mesh is depicted in Figure 7.4. The fluid mesh footprint at the wetted boundary matches the structural one. A polynomial degree of three was chosen for the structure and fluid alike. The Euclidean norm of the velocity and a pressure contour plot is given in Figure 7.5 and Figure 7.6 respectively. The deflection at point A of the flag was computed to be $8.58 \times 10^{-4}[m]$ and is therefore approximately 4.5[%] larger than the reference value.

It was expected, at first, that the correct evaluation of the extra shear terms can eliminate

[2] It is remarked that the magnitude and time history of the pressure and velocity can not be compared directly to the ones published by [45] because the parameters and properties of the discretization methods are different. For example, the structure is computed geometrically linear in the present thesis, the solid elements used in [45] exhibit a strong locking effect for the parameters chosen, time steps are different et cetera.

[3] geometrically linear here means that the contribution to the stiffness matrix stemming from geometrically nonlinear strains is neglected



(a) $t = 1.0sec$

(b) $t = 2.0sec$

(c) $t = 2.5sec$

(d) $t = 3.0sec$

(e) $t = 4.0sec$

(f) $t = 5.0sec$

Figure 7.3: Contour plot of the Euclidean velocity norm $\|\mathbf{v}\|_2$ at different time instants.

this inaccuracy. The shear at the boundary can be computed by Equation (3.19):

$$\sigma_{ij} = -\mu \left(v_{i,j} + v_{j,i} \right) \tag{3.19}$$

However, transferring the shear terms leads to a deflection of the flag of $1.1 \times 10^{-3} [m]$. A comparison with a CFX solution highlights the source of the difference. Figure 7.7 shows the pressures at the top of the flag as computed by CFX to the ones computed by $N\epsilon\kappa\tau\alpha\rho$.
The components of the traction vector $t_i = \sigma_{ij} n_j$ are given in comparison to the computation by CFX in Figure 7.8 and Figure 7.10 respectively. For the range $0.248 < x < 0.59$ these components are in good agreement with CFX, however, for the range $0.59 < x < 0.6$ the tractions computed by $N\epsilon\kappa\tau\alpha\rho$ seem to follow a developing singularity. The tractions computed by CFX do not show this behaviour.

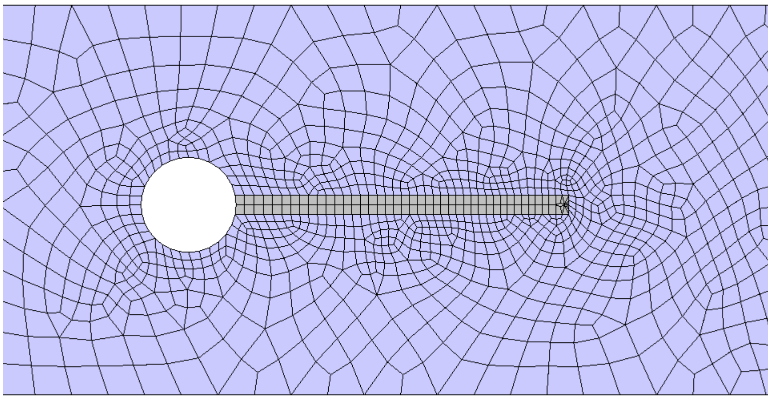


Figure 7.4: FSI1: Fluid spectral- and structural p-element mesh

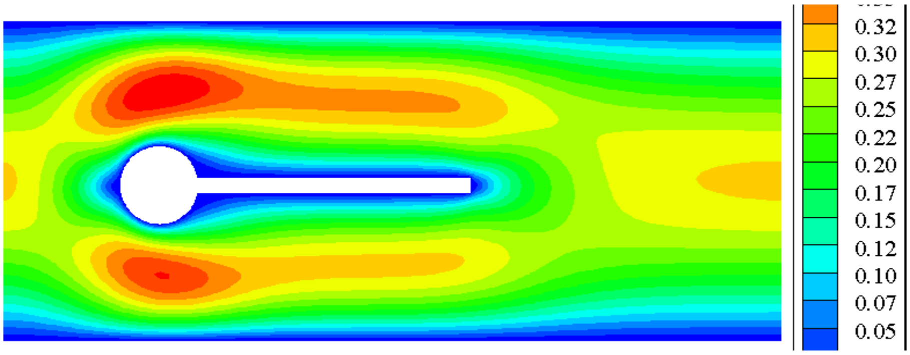


Figure 7.5: FSI1: Contour plot of the Euclidean velocity norm $\|\mathbf{v}\|_2$

Figure 7.6: FSI1: Contour plot of the pressures

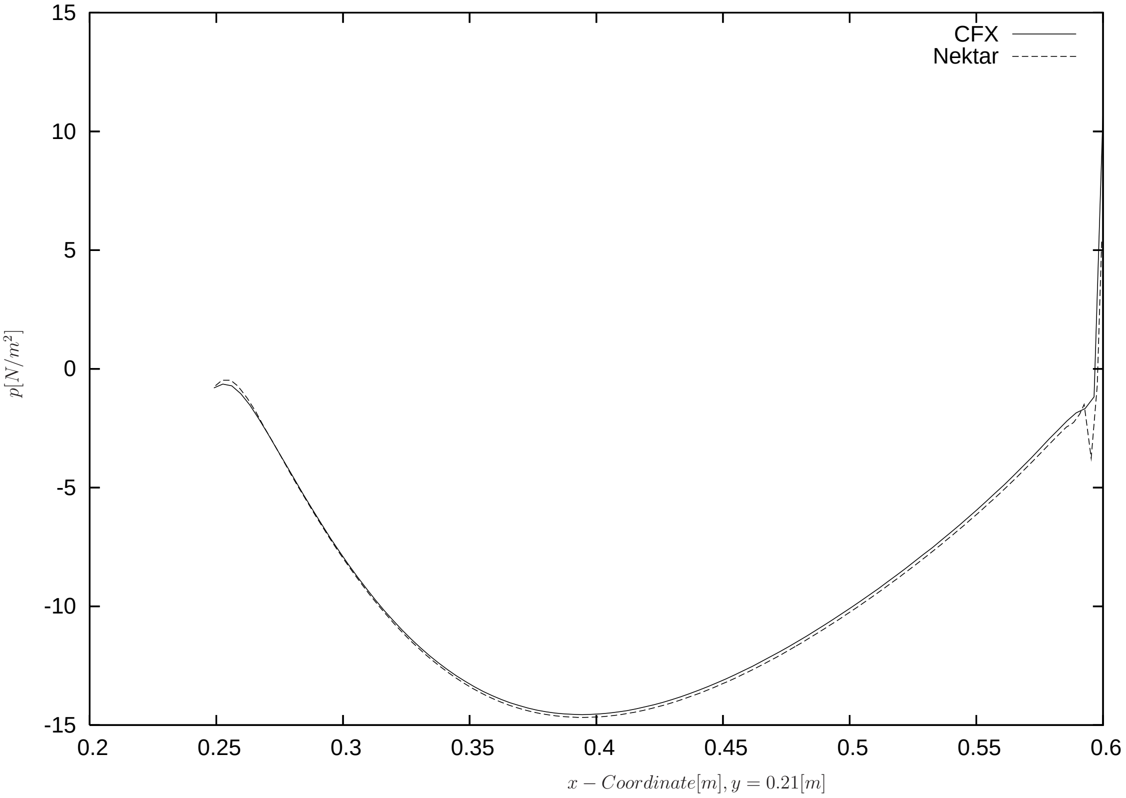


Figure 7.7: FSI1: pressure at top of the tail, left: x=0.248...0.6

The corners at the end of the flag result in sharp gradients of the pressure and stress fields. There are two possibilities: (a) The sharp gradients impose more difficulties to the spectral element discretization then low order methods do. (b) There is a singularity in the solution solution which is not seen so clearly by other methods. In this case there might not be a solution at all. Parallels might then be drawn to linear elasticity, where point loads provoke a singularity in the solution and where a refinement does not lead to convergence. In both cases, a simple remedy could be to round off the corners. However, then the definition of the geometry would have to be altered and would not represent the benchmark problems defined in Chapter 5 any more.

At this point, only the following can be stated: The proposed algorithm leads to results that are in the range of but different from the results of the computations of other codes on the benchmark problems presented in Chapter 5. This does not mean the results are incorrect, merely that they are different.

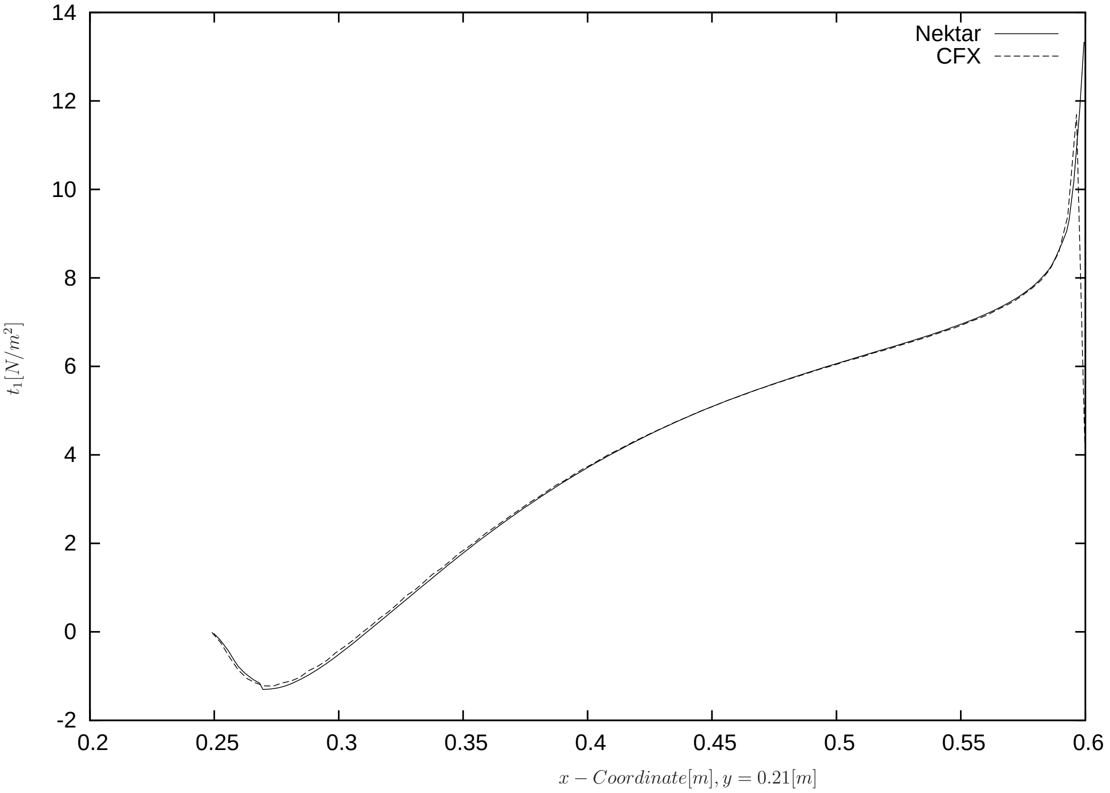


Figure 7.8: FSI1: t_1 at y=0.21[m] (top of the tail): x=0.248...0.6[m]

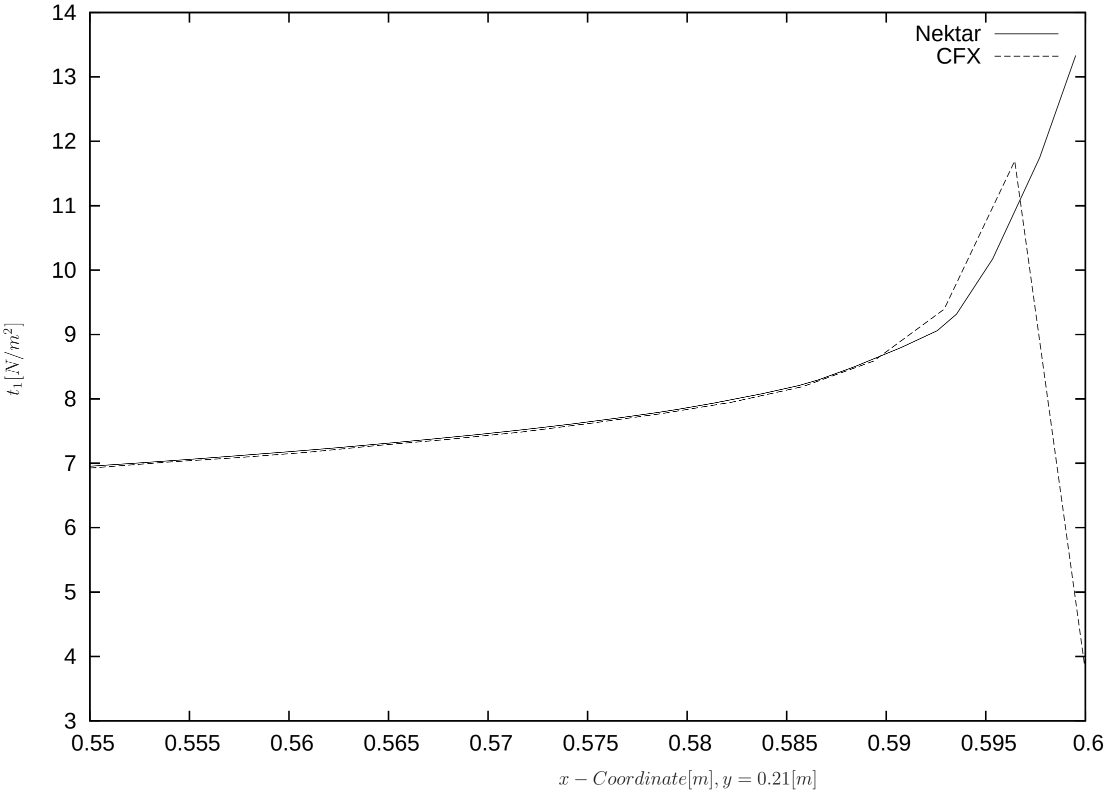


Figure 7.9: FSI1: t_1 at y=0.21[m] (top of the tail): zoom to tail

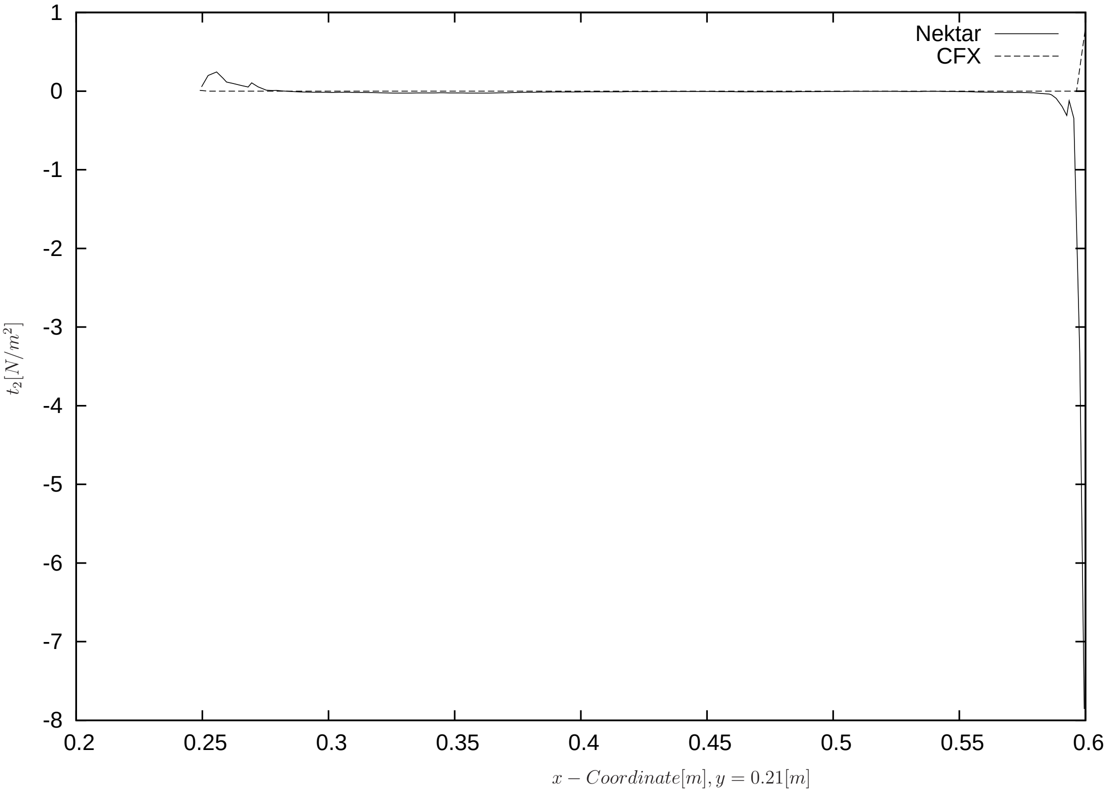


Figure 7.10: FSI1: t_2 at y=0.21[m] (top of the tail): left: x=0.248...0.6

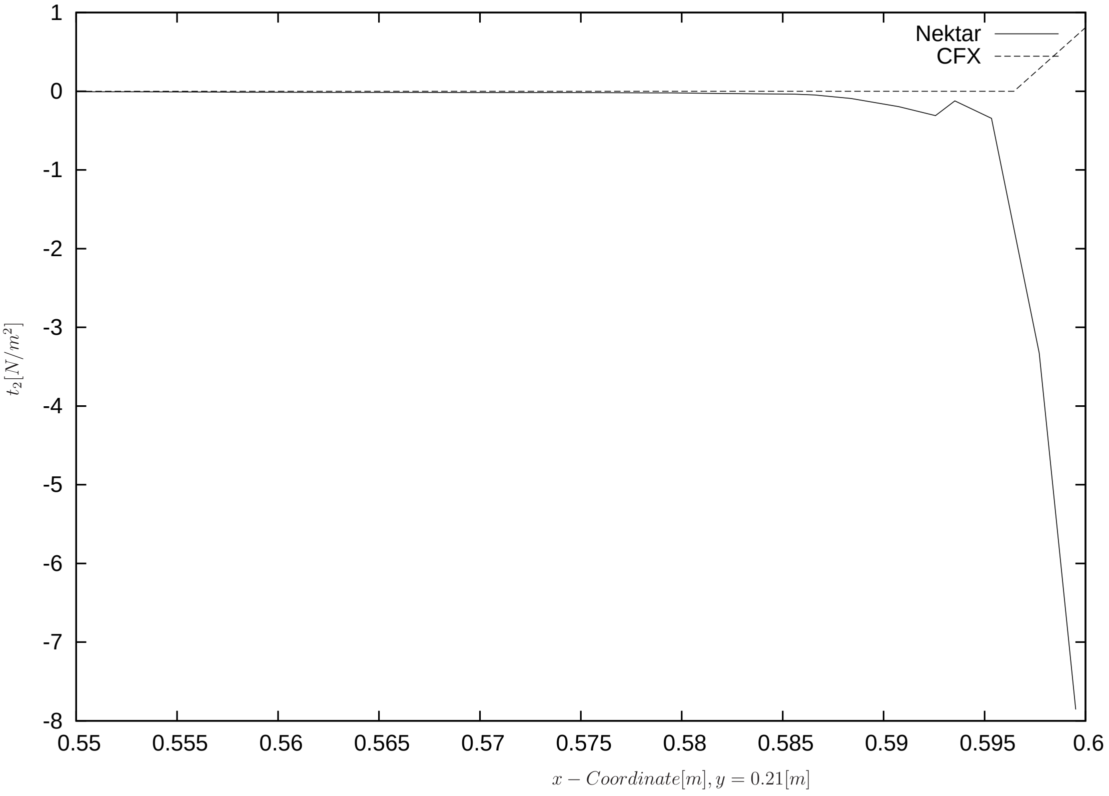


Figure 7.11: FSI1: t_2 at y=0.21[m] (top of the tail): right: zoom to tail

Chapter 8

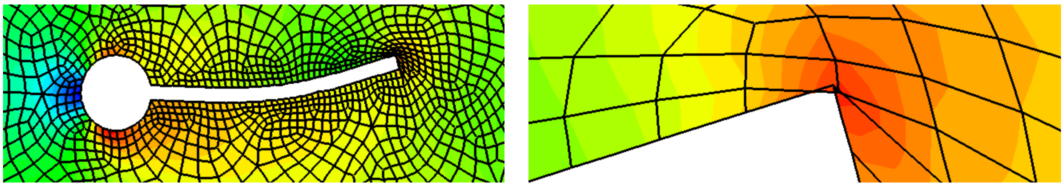
Coupling to the Lattice Boltzmann Method

8.1 Recapitulation, Motivation and Questions to answer

This Chapter proposes and investigates new methods for coupling high order structural Finite Elements to the Lattice Boltzmann Method in order to compute large displacement, transient fluid-structure interaction. Efficiency and accuracy studies are carried out for benchmark problems. The results are compared to ones obtained by completely different methods. Parts of this Chapter have been published in [76].

Among the definitions given in Chapter 1, the previously presented approaches classify as a type (a) coupling, in which the fluid is modelled via an ALE-formulation. As already pointed out, these methods were not able to compute the Benchmarks with large deflections. The end of the computation was forced once the mesh elements overlapped. Figure 8.1 depicts this situation.

That does not mean that these cases can not be computed via a coupling of type (a) in general. There are several remedies. One option is to remesh the domain once the element deformation becomes too large. As remeshing changes the topology of the mesh and therefore the discretization, the state variables need to be transferred from the old mesh to the new mesh, preferably by a projection. Another option is to use a better mesh deformation method. The method implemented in CFX is based on solving a Poisson system of equations where the



(a) pressure field and mesh (b) zoom to tail tip

Figure 8.1: Breakdown of ALE formulation due to large mesh deformation, FSI2 $N\epsilon\kappa\tau\alpha\rho$

left-hand side is a force term indirectly proportional to the volume of the finite volumes. This helps in some situations, however the sharp corners at the tip of the flag combined with the large deflections depict a strong difficulty for this approach. $\mathcal{N}\epsilon\kappa\tau\alpha\rho$ works with the simpler Laplace Equation (7.7). Methods which are capable of enduring the large mesh deformations present in cases FSI2 and FSI3 do exist. The group of Schäfer [113], for example, uses an elliptic grid movement technique as proposed in [124]. Besides the collection of methods published in [128], grid movement techniques and their limits are topic of the book by Hansen et. al. [60]. Interestingly a horseshoe shaped region with sharp concave corners similar to the one present at the tip of the flag serves therein as a benchmark test case for their performance. Some of these methods have been implemented in the course of this thesis as a stand alone program but the implementation of them into the commercial solver CFX or the code $\mathcal{N}\epsilon\kappa\tau\alpha\rho$ is not trivial. Additionally, no matter which grid deformation method is chosen, at some point, all of these must lead to unacceptably deformed discretizations, if the boundary motion is only large enough. This encourages the use of fixed grid approaches of type (b) for FSI.

The approach taken in this Chapter classifies as a special type (b) coupling since a Lattice Boltzmann Method is used to compute the fluid. It discretizes its unknowns on a fixed, Cartesian grid. For further clarification the approach for fluid-structure interaction undertaken in this Chapter is classified according to the work on Immersed Boundary Methods as presented in the review article by Peskin [103] or more recently by Mittal and Iaccarino [94] and put into the context of previous work.

The ideas of Peskin, Mittal and Iaccarino have been adopted in [120]. There the combination of a structural solver incorporating a high order spatial resolution of the structure and a Lattice Boltzmann Method for the computation of the fluid has been proposed in the context of a fictitious domain method, where the interface conditions are imposed by means of a discrete Lagrange multiplier method. The fluid nodes remain active inside the structure.

The approach taken in this Chapter is more similar to a Discrete Forcing approach of an Immersed Boundary Method, where the fluid-structure interface is represented by a sharp interface and the boundary conditions are directly imposed onto the structure and on the fluid at this discrete interface. Thus, the proposed method is most closely related to the *Ghost-Cell Finite-Difference* approach mentioned in [94].

There have been previous contributions to this field. The numerical simulation described in [79] deals with fluid-structure interaction studying a two-dimensional heart-valve with the structure only performing rigid body motions. In order to incorporate flexible structures, the structure was computed by a modal expansion using only the low-frequency modes [118, 51] while the coupled FSI problem was solved by a time-marching scheme. The forces were exchanged on a Gaussian point mesh which was neither adapted to the needs of the fluid solver nor correctly mapped to the structural discretization. The displacements obtained were in the range of the displacements published by Hron and Turek [132] as long as the error of computing the structure *geometrically linearly* remained small enough[1].

In this chapter, a time domain approach is adopted for the structure in order to be able to compute geometrically nonlinear displacements. An interface mesh is introduced which adapts the two completely different discretizations at the interface and it is demonstrated that the Benchmarks proposed in [132] can be accurately and stably computed using the explicit coupling algorithm with the force and displacement transfer methods described in

[1] i.e for FSI1 and FSI3.

this chapter.
Specifically, following issues are addressed:

- How can one deal with the problem of the moving boundary in the fluid solver within this setting?
- How can tractions and displacements be exchanged within this setting?
- Is it possible to solve the given Benchmarks accurately enough using the chosen setup?
- Is it possible to maintain an explicit overall coupling procedure?
- How efficient is the proposed approach?

At this stage it is worth pointing out the motivation of asking the last question. In order to compute the solution, the worst case scenario for an implicit, partitioned solution makes it necessary to:

- solve a non-linear fluid problem usually resulting in solving a linear system i times
- solve a non-linear structural problem usually resulting in solving a linear system j times
- possibly iterate over the fluid and solid k times
- do the above steps l times

which leads to $(i + j) * k * l$ solutions of large linear systems for one overall calculation. This does not include the extra effort needed to solve a system for the mesh movement and/or the effort needed to transfer the interface information from the fluid to the solid. The entire process is a computationally intensive task.
In order to reduce the effort, it is therefore desirable to use

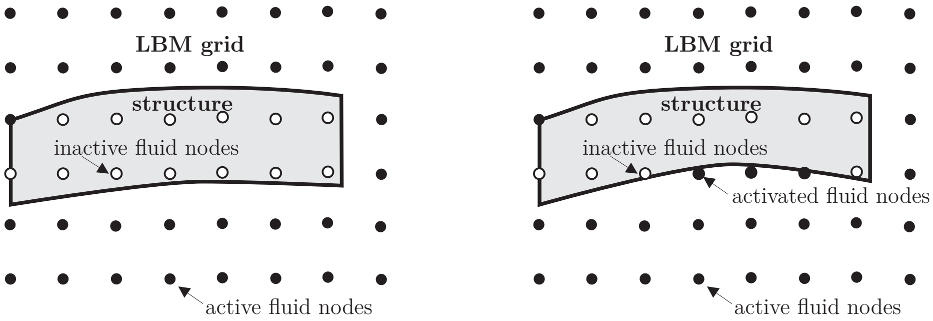
1. a very efficient fluid solver, possibly incorporating an explicit time-stepping scheme on the fluid side, thereby setting $i = 1$
2. a very efficient structural solver which, if not explicit, should have a high convergence rate, if possible
3. an explicit or staggered scheme, in which the information at the interface is only exchanged once per time-step. In this case k reduces to 1.

These goals are difficult to achieve. The most vital restriction is the third goal, i.e. the desire for an explicit coupling algorithm. Whereas explicit, staggered coupling algorithms can be applied in cases where the fluid is *compressible*, it proved to be impossible to design a stable, explicit staggered scheme if the fluid is *incompressible* [45]. Since the instability stems from the incompressibility condition of the fluid, a possible, simple remedy is to relax this condition by using a fluid solver which models weakly compressible flows. A further motivation is that the timescale of the fluid problem is dictated by the smaller time scales on the fluid side as compared to the structural side. This often results in having to choose a very small time-step for the fluid anyway. It would therefore be very convenient if one could employ an explicit overall coupling procedure, preferably one in which the fluid performs smaller time steps than the structure.

The lattice Boltzmann fluid solver is inherently explicit and describes weakly compressible fluids which will help in trying to maintain an explicit overall-coupling algorithm. It discretizes the fluid on an adaptive, fixed Cartesian grid so we do not have to worry about fluid-mesh deformations and the associated problems. In addition, it provides fast and accurate solutions for fluid problems. Moreover, the efficiency of this CFD code *Virtual Fluids* has already been demonstrated in [50], for instance, for non FSI problems. It has been shown that, for the weakly compressible case, the LB approach has a significant wall clock time advantage compared to more classical methods like finite volume or finite element methods discretizing the Navier-Stokes equations. Specifically, *Virtual Fluids* was as fast as Feat-Flow[2] and faster than CFX.

8.1.1 General considerations

The moving boundary between the fluid and the structure is considered as depicted in Figure 8.2. With the lattice Boltzmann fluid solver being a finite difference like discretization on a regular fixed grid and the structural solver based on high order finite elements, the following situation presents itself on the boundary: the lattice Boltzmann solver describes its unknowns $f_i(t, \mathbf{x})$ at specific grid points and propagates them through this lattice. In general, the boundary does not coincide with these grid points. In our implementation, the fluid boundary is described by a polygon which is used for inter- or extrapolation of the unknowns $f_i(t, \mathbf{x})$ [13]. The structural solver, on the other hand, utilizes an explicit geometric description of its boundary. In two dimensions it consists of piecewise continuous polynomials, which typically span much greater distances than those between two fluid nodes. The boundary description at the interface must therefore be controlled by the structural solver.



(a) Situation at time-step t_n

(b) Situation at time-step t_{n+1}

Figure 8.2: Situation at the boundary: curved p-elements in a lattice Boltzmann grid. Fluid nodes have to be activated and / or deactivated, as required

Transient fluid structure interaction with large structural displacements are the goal of the investigation. This implies that fluid nodes, which were positioned inside the fluid domain at time-step t_n and therefore actively take part in the fluid calculation, may at the next time-step t_{n+1} be located inside the structural domain. Since the fluid does not penetrate the structural

[2] `http://www.mathematik.uni-dortmund.de/~featflow/`

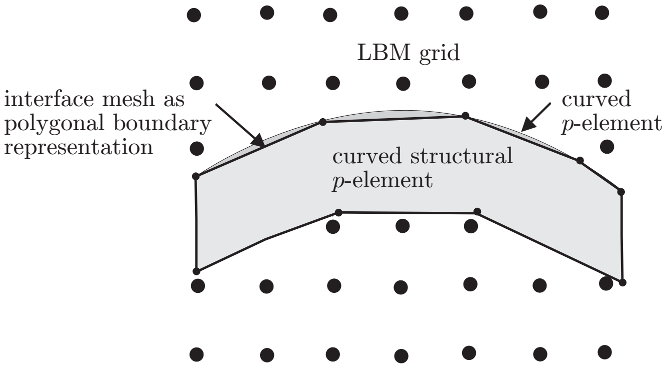

LBM grid
interface mesh as polygonal boundary representation
curved p-element
curved structural p-element

Figure 8.3: Interface mesh as the polygonal representation of the curved boundary of a curved p-quadrilateral element. The nodes of the Interface mesh coincide with the curved boundary.

boundary, these nodes have to be deactivated. Vice versa, solid points which may happen to move to the fluid domain have to be activated by initializing them with reasonable values for the unknowns $f_i(t, \mathbf{x})$ to become fluid nodes contributing to the fluid solution. This situation is depicted in Figure 8.2a and Figure 8.2b.

Within this setting, the overall accuracy of the data transfer on the interface is mainly influenced by:

1. the discretization of the boundary and its proximity

2. the initialization of fluid points with reasonable values

3. the evaluation of tractions at the fluid boundary

4. the computation of the structural force vector

The first point is briefly addressed in Chapter 8.1.2 and further elaborated in Chapter 8.4.1.2 while the second and third point are investigated in Chapter 8.3. The calculation of the structural force vector is treated in Chapter 8.4.

8.1.2 Boundary discretization

In contrast to the FEM with high order Ansatz functions and relatively large elements, the grid used within the LBM is Cartesian and in general very fine. The interface description should be polygonal with a spacing in the order of the grid size. At the same time, the structure needs to be able to integrate tractions on this mesh into its structural load vector. Therefore, the boundary is discretized in two dimensions by a polygonal interpolation of the higher order boundary provided by the structural solver, as depicted in Figure 8.3.

This polygonal representation is used by both codes to exchange data at the interface and is therefore referred to as *interface mesh.* The choice of the interface mesh is further elaborated in Chapter 8.4.1.2.

8.2 Coupling algorithm

8.2.1 Setup process

The framework (for details see [13]) is based on a client-server concept and realized through MPI [46]. A master process manages the exchange of data between the structural and fluid code. Tractions and displacements are exchanged on an interface mesh common to both codes. The data to be transferred is represented by double values organized in an array structure situated on the nodes of the polygonal interface mesh and is assumed to vary linearly between the nodes of the interface mesh (see also assumptions on tractions in Chapter 8.3.2 and displacements in Chapter 8.4.2). As part of the setup, the structural solver provides the supervisor process with the interface mesh in its initial configuration. This mesh is transferred to the fluid solver.

8.2.2 Core algorithm

To understand the time-stepping procedure of the FSI algorithm (Figure 8.4) it is important to note that a *nested* subcycling is used for the fluid solver. While n fluid steps are performed corresponding to one structural step, further sub-steps (called fluid-internal steps) need to be performed in each of these fluid steps, in order to propagate the fluid unknowns $f_i(t, \mathbf{x})$ through the hierarchical, non-uniform grid defined by the LBM solver (see [21]). After the setup process (Chapter 8.2.1), the algorithm follows these steps:

1. The fluid solver computes the traction vector on the interface mesh points according to the description in Chapter 8.3.2.

2. The tractions are handed through the interface mesh to the structural solver.

3. The structure integrates the loads with the load conservative scheme, as described in Chapter 8.4.1.

4. The structural solver computes the displacements described in Chapter 8.4.2, which are then exchanged through the interface mesh.

5. The fluid solver performs an interpolation of the positions of the interface mesh in time and computes the solution.

6. Step 5 is repeated for all fluid-internal steps.

7. Step 6 is repeated for the number n of fluid-subcycling steps. The fluid stresses of the subcycling steps plus the fluid internal steps are averaged.

Thus, the algorithm utilized in this Chapter is a staggered coupling algorithm with a subiteration for the fluid with the specialty that the fluid-solver needs to perform fluid-internal steps on its subgrids. As pointed out in Chapter 4.3.1 such an algorithm is a member of the family of the *c*onventional *s*erial *s*taggered (CSS) procedures and is only conditionally stable. However, and although subject to improvement, the simple version utilized in this thesis suffices to compute the benchmark cases discussed in Chapter 8.5 within the chosen setup.

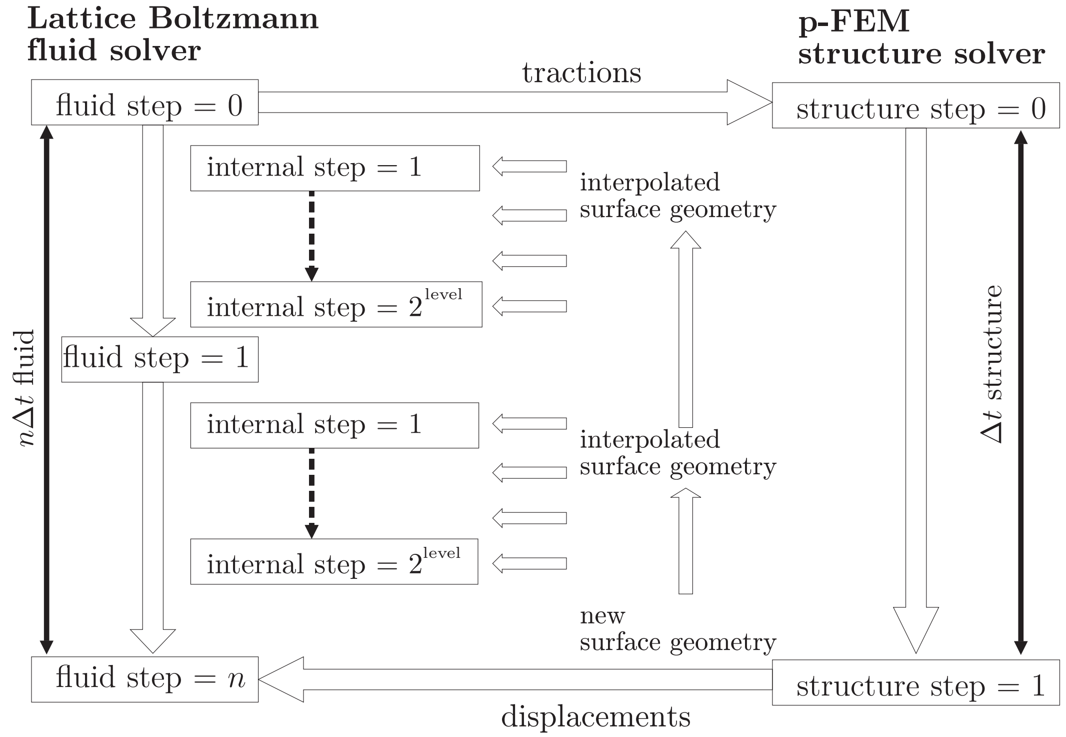


Figure 8.4: Explicit coupling algorithm with multilevel nested time-stepping

The stage is now set for an exact description of how the tractions and displacements are evaluated at the boundaries of the fluid and solid codes. This is explained in the following two Sections.

8.3 Fluid Boundary Conditions

8.3.1 Kinetic Boundary Conditions

The kinematic boundary conditions are discussed in more detail in [51], and restated here for completeness. The linear interpolated bounce back scheme developed in [12, 84] is utilized to impose velocity boundary conditions. In Figure 8.5 two cases along a link i are defined, which intersect the boundary at $\mathbf{r}_w = \mathbf{r}_b + q_i\,\hat{\mathbf{e}}_i$, where $\hat{\mathbf{e}}_i = \mathbf{e}_i\,\Delta t$:

(a) wall-node distance $q_i < 0.5$ and

(b) wall-node distance $q_i \geq 0.5$.

The modified bounce back scheme is

$$f_I(t+1,\mathbf{r}_b) = (1-2q_i)\tilde{f}_i(t,\mathbf{r}_b-\hat{\mathbf{e}}_i) + 2q_i\,\tilde{f}_i(t,\mathbf{r}_b) - \frac{6\,w_i\,\rho_0}{c^2}\,\mathbf{e}_i\cdot\mathbf{u}(t,\mathbf{r}_w), \quad 0.0 < q_i < 0.5 \tag{8.1}$$

$$f_I(t+1,\mathbf{r}_b) = \frac{(2q_i-1)}{2q_i}\,\tilde{f}_I(t,\mathbf{r}_b) + \frac{1}{2q_i}\,\tilde{f}_i(t,\mathbf{r}_b) - \frac{3\,w_i\,\rho_0}{q_i\,c^2}\,\mathbf{e}_i\cdot\mathbf{u}(t,\mathbf{r}_w), \quad 0.5 \leq q_i \leq 1.0, \tag{8.2}$$

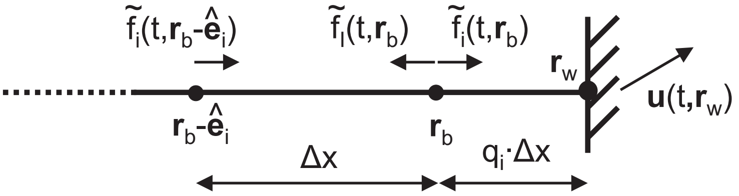

$\tilde{f}_i(t,\mathbf{r}_b-\hat{\mathbf{e}}_i)$
$\tilde{f}_I(t,\mathbf{r}_b)$
$\tilde{f}_i(t,\mathbf{r}_b)$
$\mathbf{r}_w$
$\mathbf{u}(t,\mathbf{r}_w)$
$\mathbf{r}_b-\hat{\mathbf{e}}_i$
$\mathbf{r}_b$
Δx
$q_i\cdot\Delta x$

Figure 8.5: Situation at the boundary: kinetic boundary conditions.

where $\mathbf{u}(t,\mathbf{r}_w)$ is the velocity of the moving wall. With this scheme second order accurate results in space even for curved geometries are obtained [50]. For a detailed discussion of LB boundary conditions including pressure boundary conditions it is also referred to [56].
As indicated in Chapter 8.1.1, fluid nodes have to be created and deleted due to the *moving* structure. Newly activated fluid nodes have to be initialized. Depending on the geometrical configuration, velocities are linearly inter- or extrapolated to the new nodes. A local Poisson type iteration, as described in [111], is used at those nodes to compute a consistent pressure and higher order moments.

8.3.2 Traction evaluation at the fluid boundary

There are two possibilities for evaluating forces on boundaries using the LB method: (a) the momentum exchange-based method and (b) the pressure/stress integration-based method. A comparison of both methods can be found in [91].
The momentum exchange works well for forces acting on large structural elements when only the integral of the force is relevant, but it is less feasible for the calculation of tractions when the extension of the interface line segments is in the order of the fluid mesh resolution. In the latter case, the stress integration method is preferable.
As stated in Chapter 3.4.4, the complete stress tensor τ_{ij} as given in Equation (3.17) can be computed at every node via a summation of Equation (3.71) and Equation (3.72) without the need of computing derivatives of the velocity field. The stresses at the boundary nodes are computed by inter/extrapolation.
Assume that, for a spacing h_b of interface line segments and grid size h_g of the LBM grid, $h_b > \sqrt{2}h_g$. There are 16 possible cases of how the solid boundary may be positioned with respect to the fixed fluid lattice. Figure 8.6 shows all 16 cases. They correspond to the Marching Squares (2D) and Marching Cubes (3D) algorithms [89] generating isolines and isosurfaces on fixed Cartesian grids. The index in Figure 8.6 is the sum of the bits of the fluid nodes, thereby identifying the corresponding case number.
The basic procedure of transferring stresses to the structural boundary is explained by means of an example. Consider Figure 8.7 which depicts two typical situations at a solid boundary. As indicated in the caption of Figure 8.7, the case depicted in Figure 8.7a corresponds to case (14) of Figure 8.6 and the case Figure 8.7b to case (12) of Figure 8.6. The grey area symbolizes the solid structure moving over the fluid lattice. In Figure 8.7a node *sw* lies inside the structure and is deactivated from the fluid point of view. The fluid stresses $\boldsymbol{\tau}_{sw}$ at this point are therefore zero. The procedure to compute the stresses at the boundary points $\boldsymbol{\tau}_a$, $\boldsymbol{\tau}_b$ and $\boldsymbol{\tau}_c$ is described below for $\boldsymbol{\tau}_b$:

1. Reconstruction of nodal values by extrapolation:

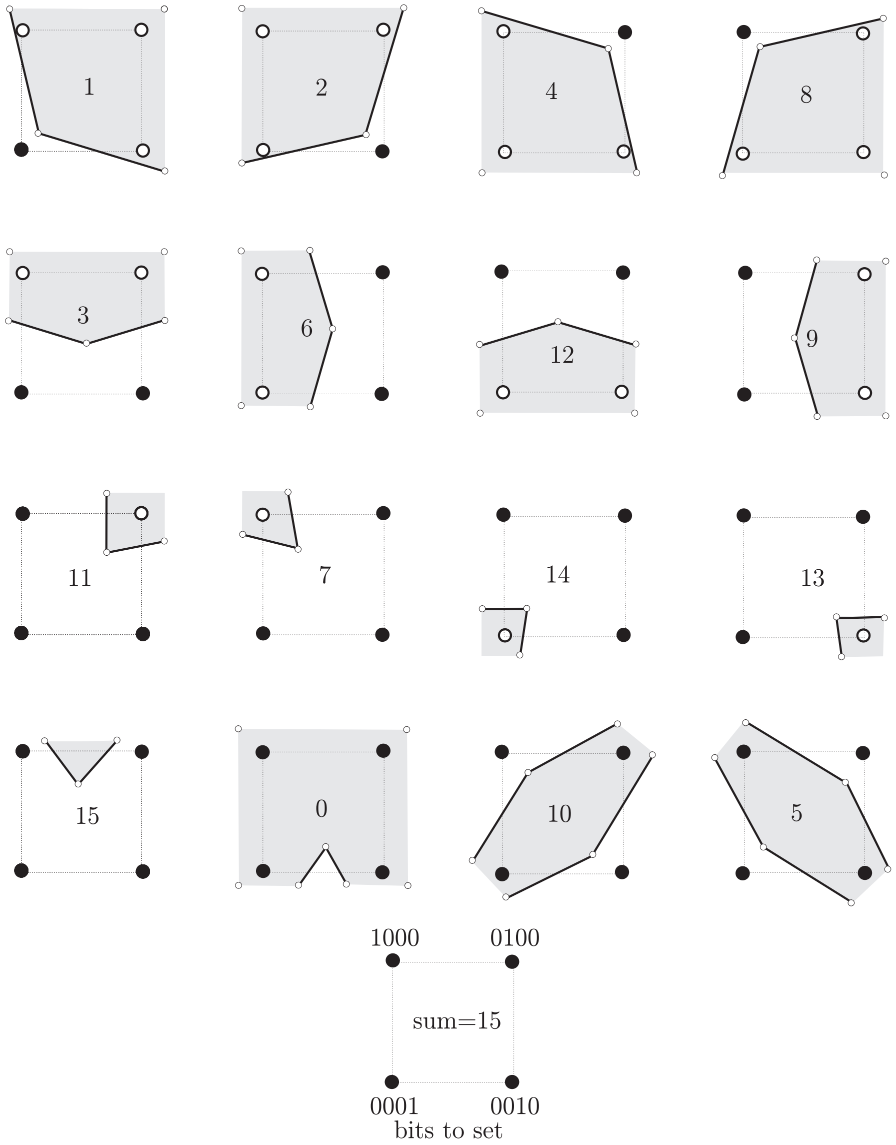
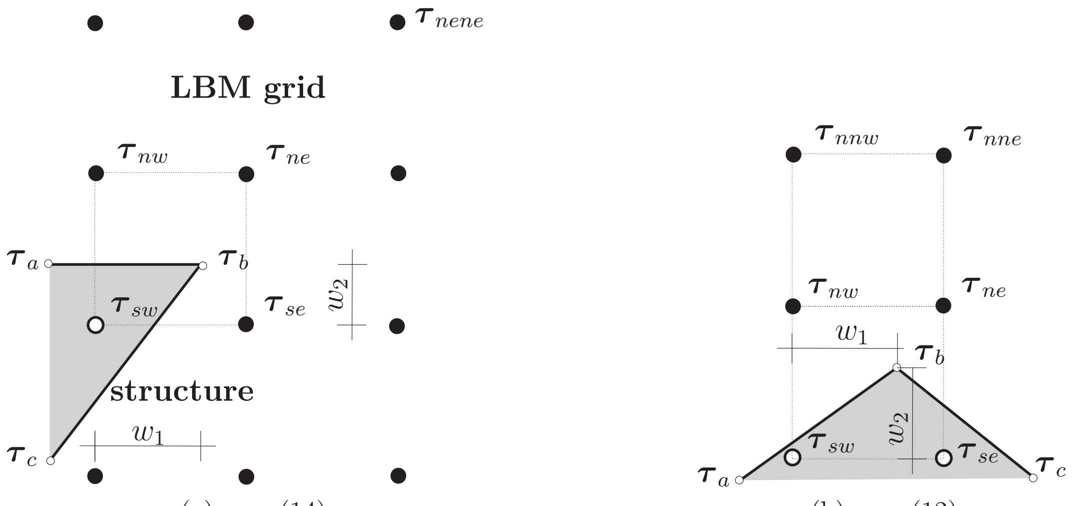


Figure 8.6: Extrapolation scheme for computing forces acting on the structure



(a) case (14)

(b) case (12)

Figure 8.7: Two possible situations at the boundary: (a) corresponds to case 14 of Figure 8.6 and (b) corresponds to case 12 of Figure 8.6

first the fluid stresses $\boldsymbol{\tau}_{sw}$ are computed by extrapolation of the stresses $\boldsymbol{\tau}_{ne}$, $\boldsymbol{\tau}_{nene}$:

$$\boldsymbol{\tau}_{sw} = 2 \cdot \boldsymbol{\tau}_{ne} - \boldsymbol{\tau}_{nene}$$

If one were to consider the situation in Figure 8.7b instead, this extrapolation would read:

$$\begin{aligned} \boldsymbol{\tau}_{sw} &= 2 \cdot \boldsymbol{\tau}_{nw} - \boldsymbol{\tau}_{nnw} \\ \boldsymbol{\tau}_{se} &= 2 \cdot \boldsymbol{\tau}_{ne} - \boldsymbol{\tau}_{nne} \end{aligned}$$

2. Bilinear interpolation of stresses to boundary nodes:
With the stresses known at the points *sw*, *nw*, *ne*, and *se* it is now possible to use bilinear interpolation to evaluate the stresses at the structural surface. The two aforementioned steps are repeated for all structural boundary nodes. As the extension of the structure is usually larger than the lattice distance of two neighboring LB-nodes, cases (10) and (5) are excluded. For case (15), the stress interpolation is bilinear. For the rare case (0) the stresses of the last time-step are imposed.

3. Computation of fluid loads on the boundary:
To obtain the traction vector $\mathbf{t}_s$ for the structure, the stress tensor $\boldsymbol{\tau}$ has to be multiplied by the normal of the boundary $\mathbf{n}_{boundary}$.

$$\mathbf{t}_s = \boldsymbol{\tau}\mathbf{n}_{boundary} \tag{8.3}$$

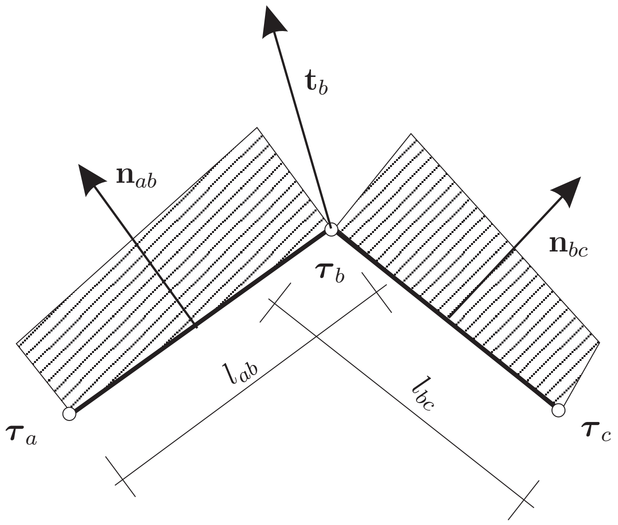


Figure 8.8: Evaluation of tractions from stress integration

Assuming that the stress components are linear between two boundary nodes (see Figure 8.8), the traction vector $\mathbf{t}_b$ at point b e.g. is computed by:

$$\mathbf{t}_b \quad = \quad \frac{(\frac{1}{4}\boldsymbol{\tau}_a + \frac{3}{4}\boldsymbol{\tau}_b)\mathbf{n}_{ab}\frac{1}{2}l_{ab} + (\frac{3}{4}\boldsymbol{\tau}_b + \frac{1}{4}\boldsymbol{\tau}_c)\mathbf{n}_{bc}\frac{1}{2}l_{bc}}{\frac{1}{2}\,(l_{ab} + l_{bc})} \tag{8.4}$$

where l_{ab} and l_{bc} are the distances between the points a,b and b,c.

8.4 Solid Boundary Conditions

8.4.1 Force transfer

8.4.1.1 Fluid loads for large structural displacements

The basic equation solved by the structural high order code FEM *AdhoC*^{4} [31] is the weak form of the equation of motion in its *material* configuration as given by Equation (2.19). The most important part in Equation (2.19) is term (d) as it describes the work done by the external tractions $\mathbf{t}_0$ on the boundary of the solid under consideration. In this *material* configuration, all forces and tractions are applied in the *initial* configuration. This is further emphasized by the notation of length of the boundary $d\mathbf{A}$ in capital letters.

However, the fluid solver updates its boundary conditions at each time-step, which implies that the tractions are computed on the *deformed* boundary. To further clarify this point, it is stressed that both the direction of the tractions and the area on which they act are a result of the (previously unknown) deformation state. The final deformation state is determined by the equilibrium of forces between the fluid and the structure, which is approximated either

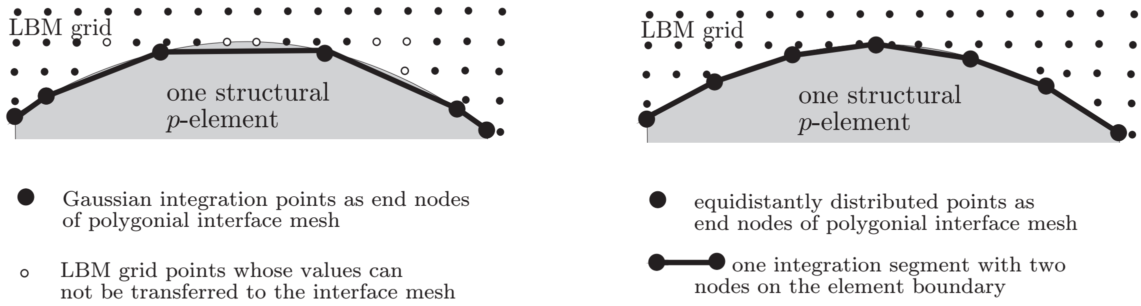
by explicit determination of the state or, more accurately, by an implicit but labor-intensive iteration over both fields.[3]
Therefore, the virtual work resulting from the fluid traction is defined as:

$$\delta W_s = \int\limits_{\partial a} \mathbf{t} \cdot \delta \mathbf{v} \, dl = \int\limits_{\partial a_{\mathrm{ref}}} \delta \mathbf{v} \cdot \mathbf{t} \parallel \mathbf{e}_3 \times \mathbf{q}_{,r} \parallel \, dr \tag{8.5}$$

in two dimensions. In the second part of Equation (8.5) the unit vector $\mathbf{e}_3$ is orthogonal to the x-y plane and $\mathbf{q}_{,r}$ is the derivative of the mapping function $\mathbf{q}$ describing the *deformed* interface. This last part describes the change of the length compared to the initial configuration and is computed in the structural solver while the traction vector $\mathbf{t}$ is computed by the fluid solver using the current length l and the normals $\mathbf{n}_{ab}$ in Equation (8.4).

8.4.1.2 Composed integration

Two types of polygonal interface meshes have been considered. They are depicted in Figure 8.9.


LBM grid
one structural
p-element
Gaussian integration points as end nodes
of polygonial interface mesh
LBM grid points whose values can
not be transferred to the interface mesh
LBM grid
one structural
p-element
equidistantly distributed points as
end nodes of polygonial interface mesh
one integration segment with two
nodes on the element boundary

(a) Gaussian interface mesh

(b) Equidistant interface mesh

Figure 8.9: Two choices for interface meshes: (a) an interface mesh whose nodes coincide with the Gaussian points on the structure (b) an equidistant mesh, ensuring that all LBM-nodes close to the boundary are considered in the load transfer. The tractions then have to be integrated into the structural force vector by means of composed integration.

The first type is a mesh whose nodes are the Gaussian integration points on the boundary of the high order structural element as depicted in Figure 8.9a. This type has been investigated in detail in [116, 115]. With the tractions directly available at the integration points, no further effort is required and the structural force vector may be integrated in the standard way. However, a closer look reveals two disadvantages. Firstly, the Gaussian mesh is much denser towards the vertices of the boundary of the *p*-elements while it is relatively coarse in the middle of the elements. With this type of mesh it is more probable that some fluid points in the middle are omitted because the local inter -/extrapolation scheme described in Chapter 8.3.2 can only bridge about the distance of two fluid nodes without disregarding intermediate ones. Secondly, this method may lead to insufficient accuracy in approximation of Equation (8.5).

[3] The same situation arises with follower loads in solid mechanics (see footnote 4 in Chapter 2.1) where the direction of the load is a result of the a priori unknown equilibrium position of the structure. Rakishly spoken, the deformation dependent term in partitioned FSI, which arises in follower loads and leads to an extra term in the stiffness matrix is treated via the "right hand side" of the coupled system.

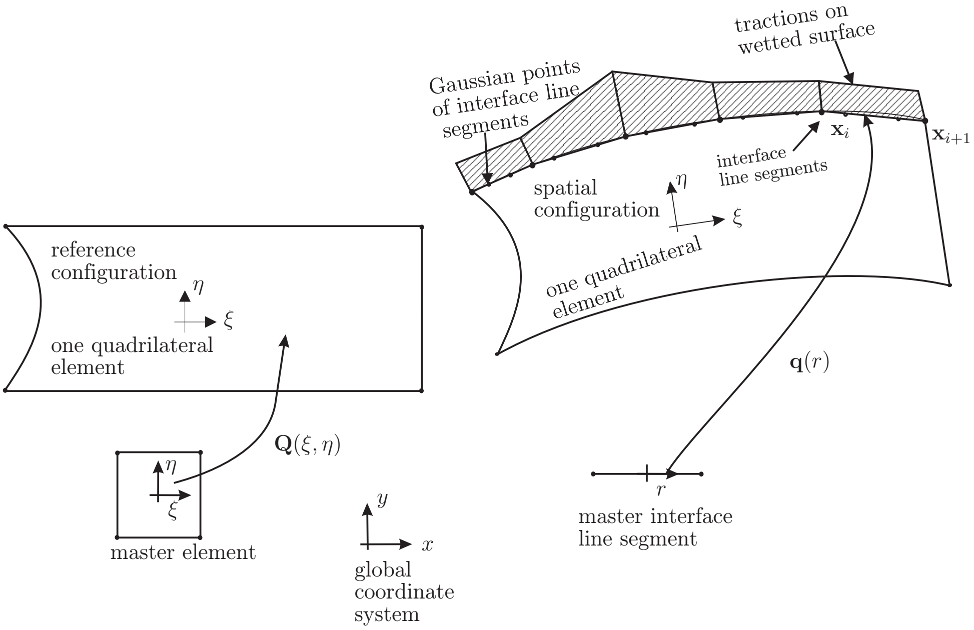


Figure 8.10: Composed integration on the boundary of a quadrilateral in contact with the fluid in a deformed configuration. Note the polygonal approximation of the curved boundary of the structural p-element. The integration of the tractions is carried out on the interface line segments. For clarity the non-deformed but blended configuration is depicted on the left-hand side.

The main difficulty in Equation (8.5) is that the load function is at most C^0 continuous while Gaussian integration assumes a C^p continuous integrand and thus cannot take the kinks of the force function on the boundaries of the fluid domain into account. Examples of the convergence of such an integration have been given in [118], for instance.

In view of these drawbacks, a second type of mesh was used on which an accurate integration may be performed, which is equidistant and may be arbitrarily fine. The principle layout is depicted in Figure 8.9b while the computational procedure is depicted in more detail in Figure 8.10. The basic idea is simple. Instead of integrating the traction function over the entire solid element with one Gaussian integration, the computation of the elemental force vector in Equation (8.5) is split up into sections (e.g. integration segments) and the integration is then performed as a composed integration of these segments (see also Chapter 4.2.3). The integration is carried out in the spatial configuration according to Equation (8.5) where a linear function $\mathbf{q}$ is chosen for the mapping of the interface lines. In this case, the derivative of $\mathbf{q}$ simply evaluates to:

$$\mathbf{q}_{,r}^{i} \;=\; \frac{1}{2}\left(\mathbf{x}^{i+1}-\mathbf{x}^{i}\right) \tag{8.6}$$

where the mapping of the i^{th} interface line segment is denoted by $\mathbf{q}_{,r}^{i}$. In Equation (8.6) $\mathbf{x}^i$ represents the position vector of the i^{th} point of the integration segments. A Gaussian integration is performed within each interface line segment.

The discrete form of the elemental structural surface load vector is then *composed* by summing up the values from all interface line segments as:

$$\mathbf{F}_s^e = \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{N}_s^{eT}(\xi_j^i, \eta_j^i)\mathbf{t}_s^i(r_j) \parallel \mathbf{e}_3 \times \mathbf{q}_{,r}^{\,i}(r_j) \parallel w_j^i \tag{8.7}$$

where m is the number of interface line segments forming the interface mesh of one structural element and n is the number of Gaussian points r_j used for the integration of each individual line segment. $\mathbf{N}_s^{eT}(\xi, \eta)$ are the shape functions of the loaded quadrilateral which are evaluated at the j^{th} Gaussian point of the i^{th} interface segments and w_j^i are the corresponding Gaussian weights. Since the values of $\mathbf{t}_s^i(r)$ are given only at the nodes of the interface line segments and a linear representation is assumed in coincidence with Chapter 8.2.1 and Chapter 8.3.2, the tractions are interpolated linearly to the individual Gaussian points of the interface line segments.

8.4.2 Displacement transfer

The displacements are evaluated by the structural solver directly at the nodes of the interface mesh and transferred as a new boundary position to the fluid. It should be borne in mind that, unlike the fluid, the structure represents its boundary by a piecewise polynomial of *higher* order. This disparity of the boundary description might lead to introducing or rarefying energy at the interface. A global L_2 or a local $H^{\frac{1}{2}}$ projection as described in Chapter 4.2.4 might serve to minimize this error. This has not been performed here.

8.5 Verification against numerical Benchmark

The coupling methods were verified against the two-dimensional Benchmarks described in Chapter 5.1.

All proposed configurations have been computed using the methods described in this chapter. On the structural side, the anisotropic *trunk space* was applied with a polynomial degree of 6 in longitudinal direction and 4 in thickness direction.

The fluid was discretized by a refined grid of level 6-8 unless stated otherwise. The following notion of refinement is used: level 0 states that the grid resolution Δx is as large as the shorter boundary of the domain, meaning that the shorter boundary is resolved by only two LB nodes. Each refinement level bisects the grid spacing. A refinement level of 1 resolves the shorter boundary with three LB nodes, level 2 with five LB nodes and so on. The number of nodes in a square domain can be computed by: $LBM_{nodes} = \left(2^{level\,n} + 1\right)^2$ and accordingly, the number of nodes present for the current benchmark geometry can be estimated for one level by $LBM_{nodes} = \left(2^{level\,n} + 1\right)^2 \frac{2.1[m]}{0.42[m]}$. Level 6-8 states that an adaptive *local* refinement has been performed starting from level 6 down to level 8. The refinement criterion used here was proposed in [21]. The fluid calculation was started from a converged solution using a fixed flag. The adapted refinement resulted in a “cloud” like refinement around the structure and into the field. A graphic illustration of the spatial refinement is depicted in Figure 8.11. The total amount of fluid grid points is shown in Table 8.1. Although it is possible in principle

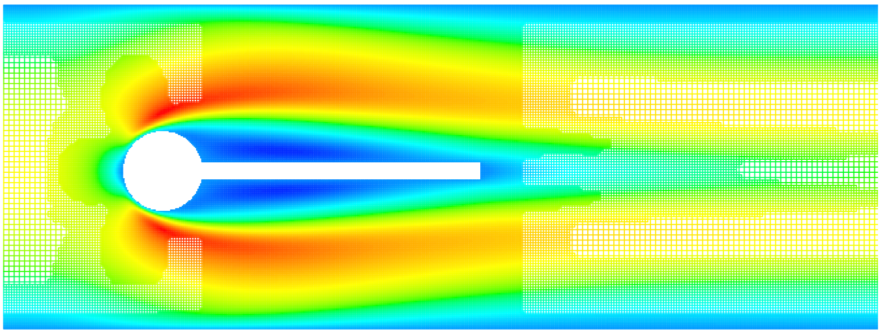


Figure 8.11: FSI2: Zoom of discretization and refinement for level 6-8.

to adapt the refinement during the solution of the adapted transient problems, it is quite an expensive operation. Therefore, the precomputed adapted refinement was used and kept fixed, which leads to satisfactory results also for the instationary benchmarks.
Unless stated otherwise, a refinement of level 8 was present at the structure. This leads to a spacing of the LBM nodes at the structural interface of $h_b = 1.68 \times 10^{-3}[m]$. The size of interface line segments of the interface mesh was chosen to be $h_b \approx 2.0 \times 10^{-3}[m]$ such that almost every fluid node has a corresponding point on the interface mesh.
The structure is composed of extremely soft material and exhibits strong geometrically nonlinear deflections for cases FSI2 and FSI3. It is difficult to define a stable time integration method for such a configuration. Applying the generalized-α method leads to unstable results in the coupled computation. As stated in Chapter 6.2 and Chapter 7.2.2, a Newmark method with a parameter set of $\beta = 0.49$, $\gamma = 0.9$ to eliminate numerical instabilities in the structural time integration was utilized. For the sake of comparison, these parameters were also used in the stationary case FSI1. All cases were computed with a fluid subcycling of two: for one overall coupling step, the fluid performed two complete fluid solutions including internal substeps, while the structure only performed one. The fluid Mach number was chosen to be $Ma = 0.1$ in each case. All other specifications are listed in Table 8.1.

Case	Δt *structure[s]*	*LBM nodes*
FSI1	2.52591×10^{-3}	[125553]
FSI2	5.05182×10^{-4}	[160170]
FSI3	2.52591×10^{-4}	[275646]

Table 8.1: Specification of parameters used to compute FSI1, FSI2 and FSI3

The deflection of the flag at point A, as depicted in Figure 5.2, is summarized in Table 8.2 and compared with the results obtained by [132]. The deflections are given as "$\langle mean \rangle$ ±

$\langle amplitude \rangle$[Hz]" where $mean = \frac{1}{2}(max + min)$ and $amplitude = \frac{1}{2}(max - min)$ are computed from the last period of the oscillations. The frequency was computed by means of a Fourier analysis of the periodic data. To give some idea of the computational efficiency, the column entitled "time" gives the real time in seconds which was computed over 24 hours on a 1.6GHz AMD OPTERON processor. The computations in Chapter 6.2 are three-dimensional and the computations in Chapter 7.2.2 were obtained with inefficient programming. A direct comparison of the computational time needed is not considered to be reasonable. The stated computational times are only meant to give an estimate of the effort needed to compute these benchmark problems.

Drag and lift are evaluated by means of Equation (5.1). The results are summarized in Table 8.3 and compared with the results obtained by [132].

While FSI1 leads to a stationary displacement, FSI2 and FSI3 lead to a periodic motion of the flag. These instationary results are also depicted in Figure 8.12 and Figure 8.13 in which the results obtained by [132] are shown as well.

Case	$u_x(A)[m] \pm [Hz]$	$u_y(A)[m] \pm [Hz]$	time [s/day]
FSI1	2.29×10^{-5}	8.10×10^{-4}	5.2
FSI1$_{ref}$	2.27×10^{-5}	8.209×10^{-4}	∅
FSI1 [%] deviation	0.9	-1.3	∅
FSI2	$-1.51 \times 10^{-2} \pm 1.28 \times 10^{-2}[3.8]$	$1.20 \times 10^{-3} \pm 8.34 \times 10^{-2}[1.9]$	2.7
FSI2$_{ref}$	$-1.458 \times 10^{-2} \pm 1.244 \times 10^{-2}[3.8]$	$1.23 \times 10^{-3} \pm 8.306 \times 10^{-2}[2.0]$	∅
FSI2 [%] deviation	$3.6 \pm 2.9[0.0]$	$-2.4 \pm 0.4[-5.0]$	∅
FSI3	$-2.88 \times 10^{-3} \pm 2.71 \times 10^{-3}[11.0]$	$1.48 \times 10^{-3} \pm 3.51 \times 10^{-2}[5.5]$	0.8
FSI3$_{ref}$	$-2.69 \times 10^{-3} \pm 2.53 \times 10^{-3}[10.9]$	$1.48 \times 10^{-3} \pm 3.438 \times 10^{-2}[5.3]$	∅
FSI1 3 [%] deviation	$7.1 \pm 7.11[0.9]$	$0.0 \pm 2.1[3.8]$	∅

Table 8.2: Results obtained and comparison with reference [132]. The deflections $u_x(A)$ and $u_y(A)$ are given as follows:"$\langle mean \rangle \pm \langle amplitude \rangle$[Hz]" where $mean = \frac{1}{2}(max + min)$ and $amplitude = \frac{1}{2}(max - min)$ are computed from the last period of the oscillations; ∅ no results available

Figure 8.14 shows the fluid and the deflection of the flag of FSI2.

8.5.1 Sensitivity to selected discretization parameters

In order to show the sensitivity of the chosen parameters to the results the following convergence studies have been carried out as exemplary for test case FSI1. The structural discretization remained fixed in all test cases to the one mentioned above.

The convergence of the deflection of Point A was computed while varying the fluid discretization. As opposed to the computations with a priori local refinement, a uniform fluid discretization with only one level of fluid nodes throughout the domain was used. To balance

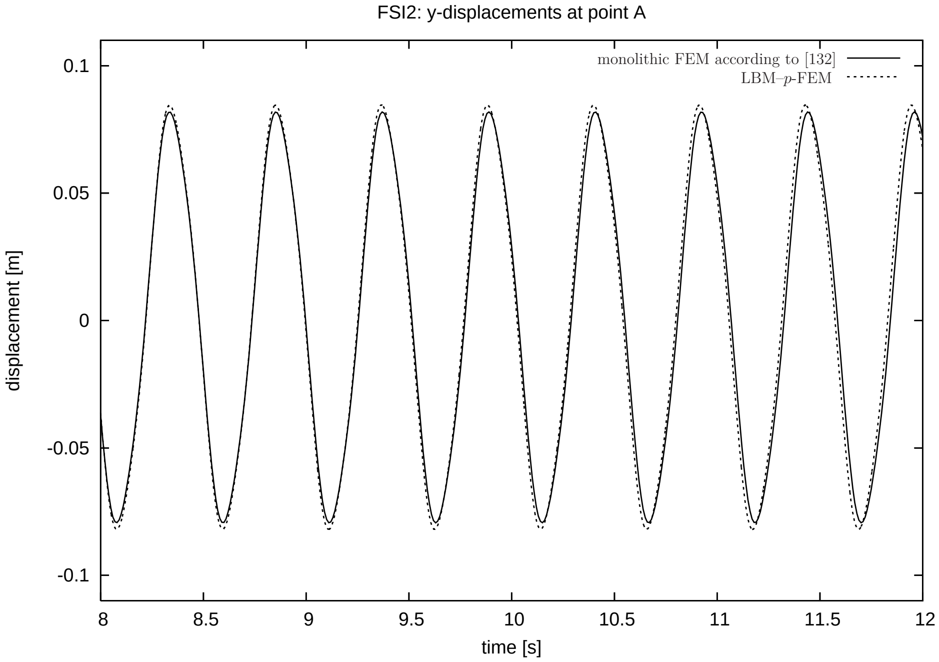


Figure 8.12: FSI2: y- deflection at point A obtained by LBM–p-FEM coupling compared to [132]

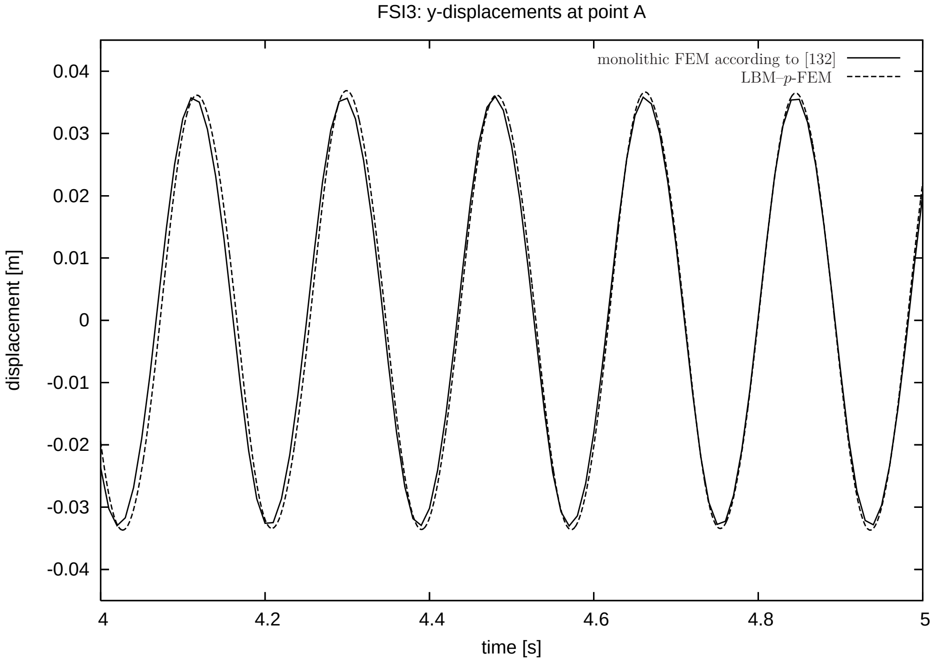


Figure 8.13: FSI3: y-deflection at point A obtained by LBM–p-FEM coupling compared to [132]

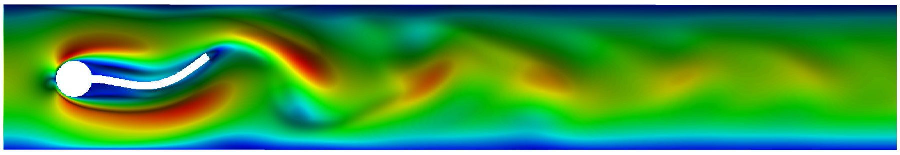


Figure 8.14: FSI2: velocity norm in the fluid at $t = 5.0[s]$

Case	$F_d[N] \pm [Hz]$	$F_l[N]\pm$ *[Hz]*
FSI1	14.31	0.757
FSI1$_{ref}$	14.295	0.7638
[%] deviation	0.1	-0.9
FSI2	$216.60 \pm 89.31[3.8]$	$-1.23 \pm 283.00[1.9]$
FSI2$_{ref}$	$208.83 \pm 73.75[3.8]$	$0.88 \pm 234.20[2.0]$
[%] deviation	$3.7 \pm 21.1[0]$	$239.77 \pm 20.9[5.3]$
FSI3	$462.53 \pm 31.34[11.0]$	$1.81 \pm 154.22[5.5]$
FSI3$_{ref}$	$457.3 \pm 22.66[10.9]$	$2.22 \pm 149.78[5.3]$
[%] deviation	$1.1 \pm 38.3[0.9]$	$-18.4 \pm 3.0[3.8]$

Table 8.3: Results obtained and comparison with reference [132], F_d and F_l are computed according to Equation (5.1) whereby ∂A is the wetted boundary of the flag plus the wetted boundary of the cylinder. Drag F_d lift F_d are given as follows:"$\langle mean \rangle \pm \langle amplitude \rangle$[Hz]" where $mean = \frac{1}{2}(max + min)$ and $amplitude = \frac{1}{2}(max - min)$ are computed from the last period of the oscillations

the error resulting from the spatial discretization and the error introduced by utilizing a Mach number greater than zero, the Mach number was reduced with each level of refinement as well. The computational time in real time seconds per day is given additionally. The results are presented in Table 8.4[4].

level	Ma	$\Delta t \times 10^{-5}[s]$	$u_y(A)[m]$	$u_x(A)[m]$	time [s/day]
7	0.05	31.57	1.018×10^{-3}	2.127×10^{-5}	11.4
8	0.025	7.89	8.337×10^{-4}	2.199×10^{-5}	3.1
9	0.0125	1.97	8.201×10^{-4}	2.216×10^{-5}	0.55

Table 8.4: FSI1: Deflections at point A for different fluid discretizations.

The computation converges to $u_y(A)[m] = 8.200 \times 10^{-4}$, which is in excellent agreement with

[4] As opposed to [76] Table IV, the data in Table 8.4 was recomputed with faster computer, an Intel Xeon CPU E5520 with 2.27GHz and a i386-apple-darwin-9.0 operating system. Additionally, the interface meshsize h_g was chosen to be $0.004[m], 0.002[m], 0.001[m]$, for level 7,8,9, respectively and is thus finer and more appropriate as in [76]

the reference value of $u_y(A)[m] = 8.209 \times 10^{-4}$.

In order to judge the influence of the size of the interface line segments two studies have been performed. Both were carried out with a fluid discretization of level 6-8, a Mach number of 0.1, and a fluid subcycling of two.

First, the size of the interface line segments was varied while the flag was fixed by assigning it a very large Young's modulus. The drag F_d and lift F_l on the structure was computed according to Equation (5.1) by evaluating the tractions through composed integration. The results are given in Table 8.5. The fluid discretization of level 6-8 results in a spatial resolution of the Boltzmann grid points of $h_g = 1.68 \times 10^{-3}[m]$ at the fluid-structure interface. The size of the interface line segments was then successively decreased from $h_b \approx 1.6 \times 10^{-2}[m]$ down to $h_b \approx 2.0 \times 10^{-3}[m]$. This corresponds to decreasing the ratio h_b/h_g from 9.8 to $h_b/h_g \approx 1.2$ whereby at $h_b/h_g \approx 1.2$ almost every fluid node has a corresponding point on the interface mesh.

h_b/h_g	$F_d[N]$	$F_l[N]$
9.8	5.483×10^{-2}	1.221×10^{-3}
4.9	5.020×10^{-2}	1.252×10^{-3}
2.4	5.118×10^{-2}	1.270×10^{-3}
1.2	5.119×10^{-2}	1.277×10^{-2}

Table 8.5: FSI1 with fixed structure: Forces on flag with different interface mesh sizes, F_d and F_l are computed according to Equation (5.1) whereby ∂A is the wetted boundary of the flag only.

As a second study the size of the interface line segments was varied with the flag having the proposed Young's Modulus of $E^s = 1.4 \times 10^6[\frac{N}{m^2}]$. The deflection of the flag and the forces at the flag were measured. The results are given in Table 8.6

h_b/h_g	$F_d[N]$	$F_l[N]$	$u_y(A)[m]$
9.8	5.162×10^{-2}	7.537^{-3}	9.798×10^{-4}
4.9	5.026×10^{-2}	7.888×10^{-3}	8.855×10^{-4}
2.4	5.033×10^{-2}	8.121×10^{-3}	8.214×10^{-4}
1.2	5.102×10^{-2}	8.189×10^{-2}	8.096×10^{-4}

Table 8.6: FSI1: Forces on flexible flag with deflection at point A with different interface mesh sizes. F_d and F_l are computed according to Equation (5.1) whereby ∂A is the wetted boundary of the flag only.

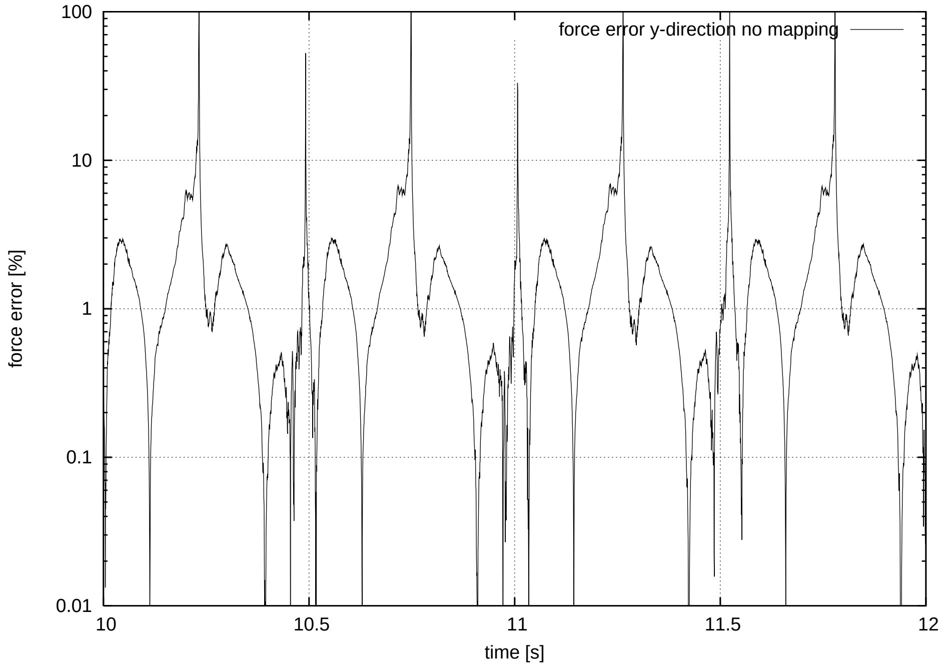


Figure 8.15: FSI2: Force error obtained by integration in material configuration

8.5.2 Discussion of consistent traction evaluation in the structure

As stated in Chapter 8.4.1, it is important to evaluate the fluid forces consistently with the deformed fluid boundary by an appropriate mapping. This is emphasized by Figure 8.16 where this mapping was performed, as compared to Figure 8.15 where the mapping was disregarded for illustrative purposes. Figure 8.15 clearly depicts the periodic behavior of the error in the force transfer, which stems from the periodic motion of the boundary. The relative error measure:

$$e[\%] = \frac{F_{fluid} - F_{structure}}{F_{fluid}} 100[\%] \tag{8.8}$$

was used to evaluate the error in the transfer of the forces. When the fluid forces F_{fluid} become zero, this error measure tends towards infinity. Graph 8.16 was thus cut off at an error of 100[%].

8.5.3 Discussion of energy conservation at the interface

As stated in Chapter 8.2 the chosen coupling method is equivalent to the *CSS* procedure analyzed e.g. by [105] with the modification that the fluid-solver performs fluid-internal steps on its subgrids.

Therefore, the analysis presented in [105] can be applied here as well. The time integration of the structure is performed by the Newmark scheme, while the fluid time integration is best characterized by an explicit forward Euler scheme. Thus, the discrete approximation to Equation (4.4) is computed on the interface mesh as both systems are advanced from time

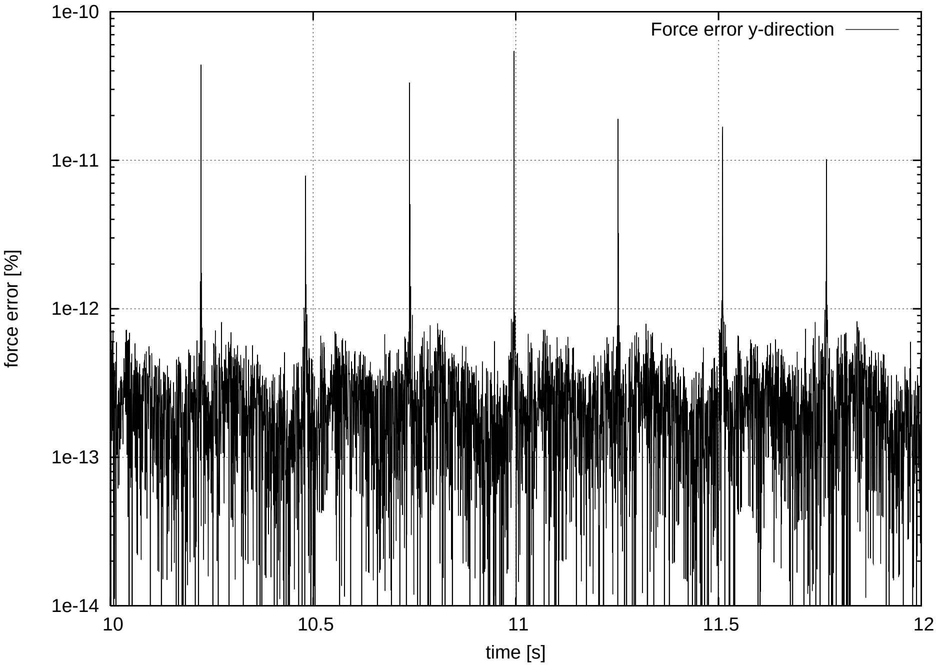


Figure 8.16: FSI2: Force error obtained by integration in spatial configuration

step n to $n+1$ as follows:

$$\begin{aligned} \Delta\mathbf{E}_\Gamma^{n\rightarrow n+1} &= \Delta\mathbf{E}_\Gamma^{S,n\rightarrow n+1} + \Delta\mathbf{E}_\Gamma^{F,n\rightarrow n+1} \\ &= \mathbf{t}_s^{(n+1),c}\cdot\left(\mathbf{d}_s^{n+1}-\mathbf{d}_s^{n}\right) - \mathbf{t}_f^{n}\cdot\left(\mathbf{d}_f^{(n+1),p}-\mathbf{d}_f^{n,p}\right) \end{aligned} \tag{8.9}$$

Here, $\mathbf{t}_s^{(n+1),c}$ are the tractions at the interface as experienced by the structure at time step $n+1$ and $\mathbf{d}_f^{(n+1),p}$ are the displacements at the interface as experienced by the fluid at time step $n+1$.

The algorithm described in Chapter 8.2 chooses a zero order predictor $\cdot^{,p}$ for the displacements and a (corrected $\cdot^{,c}$) force transfer according to:

$$\begin{aligned} \mathbf{d}_f^{(n+1),p} &= \mathbf{d}_s^{n+1} \\ \mathbf{t}_s^{(n+1),c} &= \frac{1}{\Delta t}\int\limits_n^{n+1}\mathbf{t}_f(t)\,dt \end{aligned} \tag{8.10}$$

where $\frac{1}{\Delta t}\int_n^{n+1}\mathbf{t}_f(t)\,dt$ denotes that the tractions are averaged over the considered time step. An improved version of *CSS* suggests the choice of a first order structural predictor and a force correction according to:

$$\begin{aligned} \mathbf{d}_f^{(n+1),p} &= 2\mathbf{d}_s^{n} - \mathbf{d}_s^{n-1} \\ \mathbf{t}_s^{(n+1),c} &= \frac{2}{\Delta t}\int\limits_n^{n+1}\mathbf{t}_f(t)\,dt - \mathbf{t}_f^{n} \end{aligned} \tag{8.11}$$

Exemplarily, the energy artificially created (or dissipated) at the interface is computed for case FSI2 because this configuration exhibits the largest deflections.

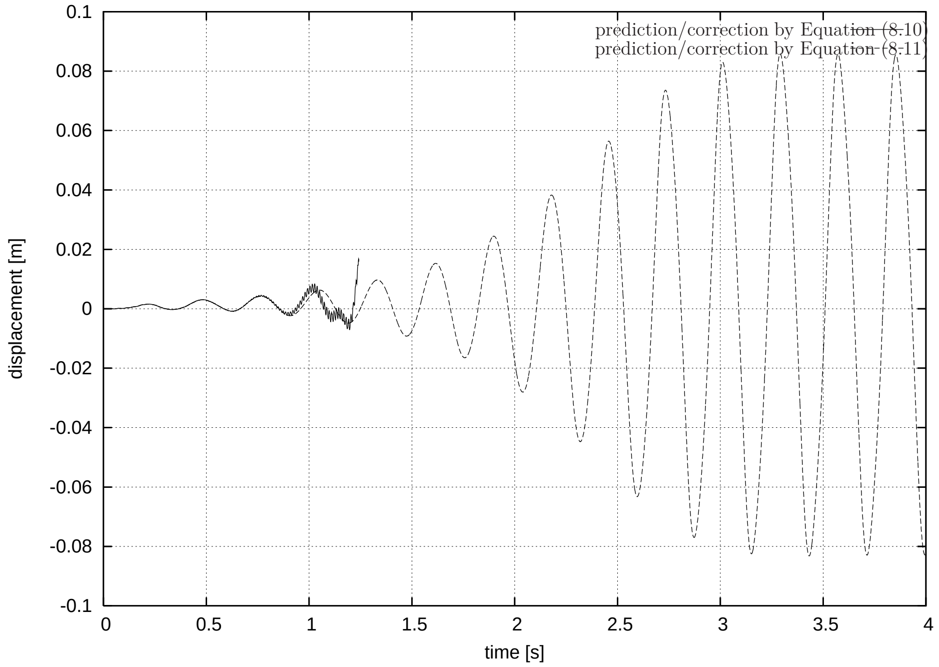


Figure 8.17: FSI2: y-deflection at point A obtained by utilizing the predictor/corrector pairs of Equation (8.10) and Equation (8.11)) at a fluid subcycling of four

While the transfer of displacements and tractions according to Equation (8.10) lead to a stable computation of FSI2 up to a fluid subcycling of two, FSI2 exhibits catastrophic instabilities with a fluid subcycling of four. Stability may be recovered by using Equation (8.11) for the exchange of variables at the interface. The corresponding displacement curves for point A are depicted in Figure 8.17 for the first four seconds of the coupled computation. Figure 8.18 displays the created artificial energy at the interface by summing up Equation (8.9) for all time steps to give $\mathbf{E}_{tot} = \sum_{n=1}^{n=n_{current}} \mathbf{E}_{\Gamma}^{n\rightarrow n+1}$. The instability of the inferior method is clearly depicted by the overproportionally large growth of $\mathbf{E}_{tot}$. By using Equation (8.11) instead of Equation (8.10) $\mathbf{E}_{tot}$ remains within acceptable bounds.

More accurate versions of the *CSS* procedure or other methods such as the *I*mproved *S*erial *S*taggered procedure *ISS* may be applied. However, as stated e.g. in [105], the optimal choice for $\mathbf{d}_f^{(n+1),p}$ and $\mathbf{t}_s^{(n+1),c}$ depends on the properties of the time integration of the individual subdomains. To be more precise, the chosen version of the Lattice Boltzmann Method is a second order time accurate method which achieves its second order accuracy by directionally integrating the unknowns $f_i(t, \mathbf{x})$ along the discrete positions at its lattice by a forward Euler procedure. In view of this background, it is not apparent at first sight which pair for $\mathbf{d}_f^{(n+1),p}$ and $\mathbf{t}_s^{(n+1),c}$ is optimal for the presented setting. A detailed investigation is beyond the scope of this thesis.

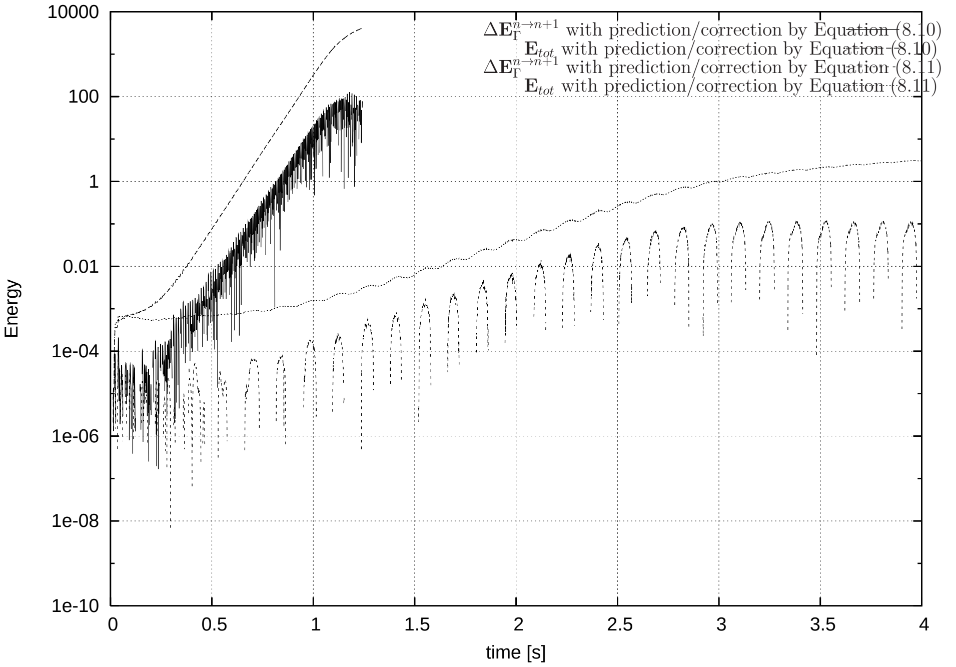


Figure 8.18: FSI2: artificial interface energy for predictor/corrector pairs of Equation (8.10) and Equation (8.11) at a fluid subcycling of four

8.6 Validation against experimental Benchmark

In this section, the computational setup is validated against the experimental Benchmarks defined in Chapter 5.2. For both Reynolds numbers this benchmark turned out to be quite challenging. Stable computations could only be obtained for a fluid grid with 1360 × 960 nodes. For lower resolutions, the mass of the flag had to be increased in order to avoid instabilities. The results of the computation are visually compared to the experiment in Figure 8.19 and Figure 8.20 for $Re = 140$ and $Re = 190$, respectively. The computed pattern of the deformation is found to be in good agreement with the experiment. The frequencies of the vibration of the flag were measured to be $6.38[Hz]$ and $13.58[Hz]$ while the computation leads to $6.71[Hz]$ and $16.7[Hz]$. The low frequency can be captured quite well. However, the deviation of the higher frequencies is not negligible and subject to further investigations. Possible remedies are to use an even finer discretization of the fluid domain, a lower Mach number or the usage of more sophisticated coupling schemata.

8.7 A three-dimensional Benchmark

The presented methods were transferred to three dimensions. However, the extrapolation scheme as described in Chapter 8.3.2 requires a quick localization not only of the discretization points next to the structure, but also of the discretization points in the second row next to the structure including all its discrete distribution function values f_i. A quick access to these is not implemented into the data structure. The provision of all the necessary f_i's is work in progress.

To avoid this difficulty, the stress computation at the fluid boundary in the fluid solver is

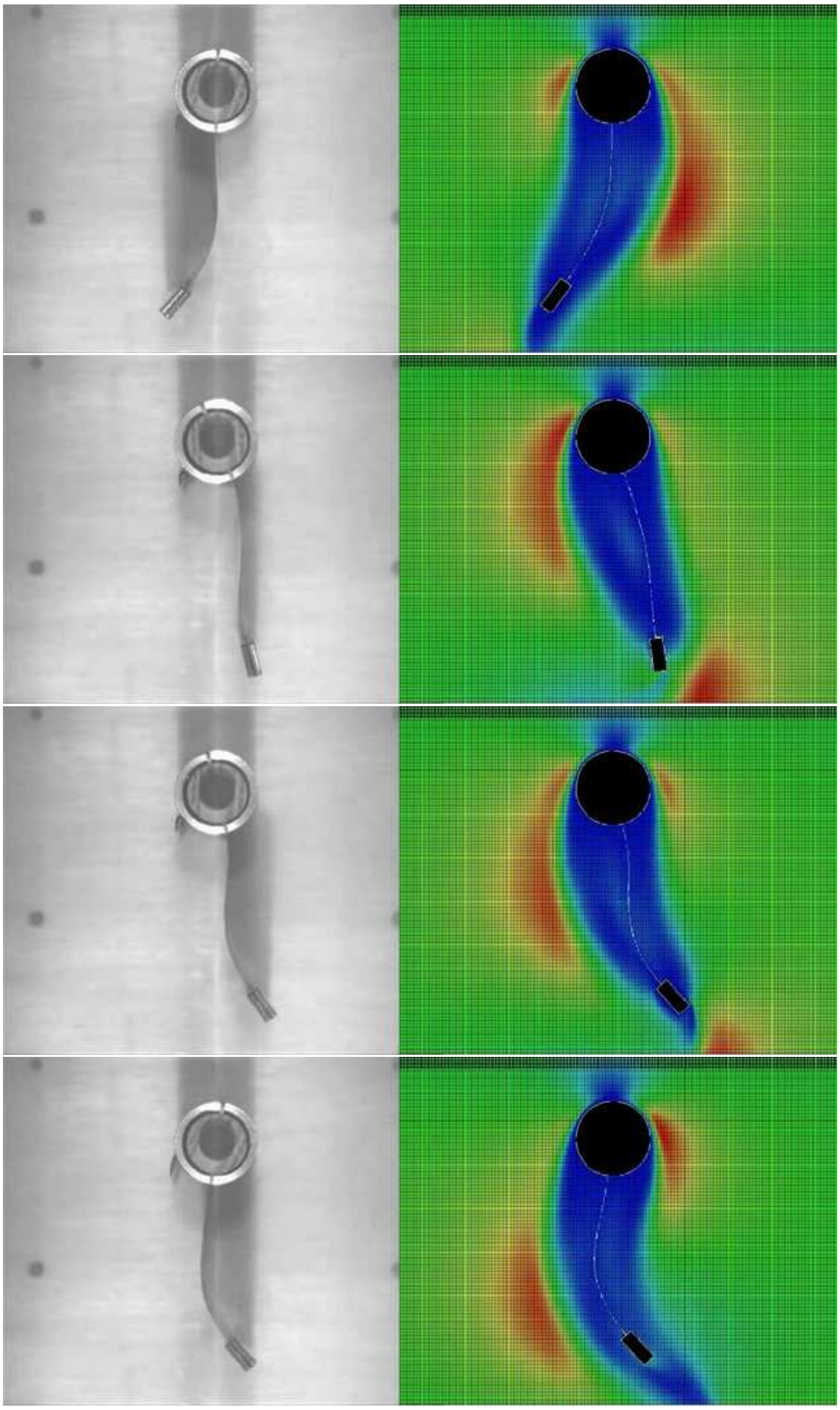


Figure 8.19: experimental Benchmark 1 $Re = 140$

implemented in a simplified form. For now, only the closest fluid node along the normal direction of the surface is detected. Its stress tensor is then computed and directly imposed on the boundary without extrapolation. Such a procedure is also called next neighbour interpolation (see i. e. [58]). This is only a first step and it is the final goal to fully implement the method presented in Chapter 8.3.2. However, for simple cases, the simplified method leads to

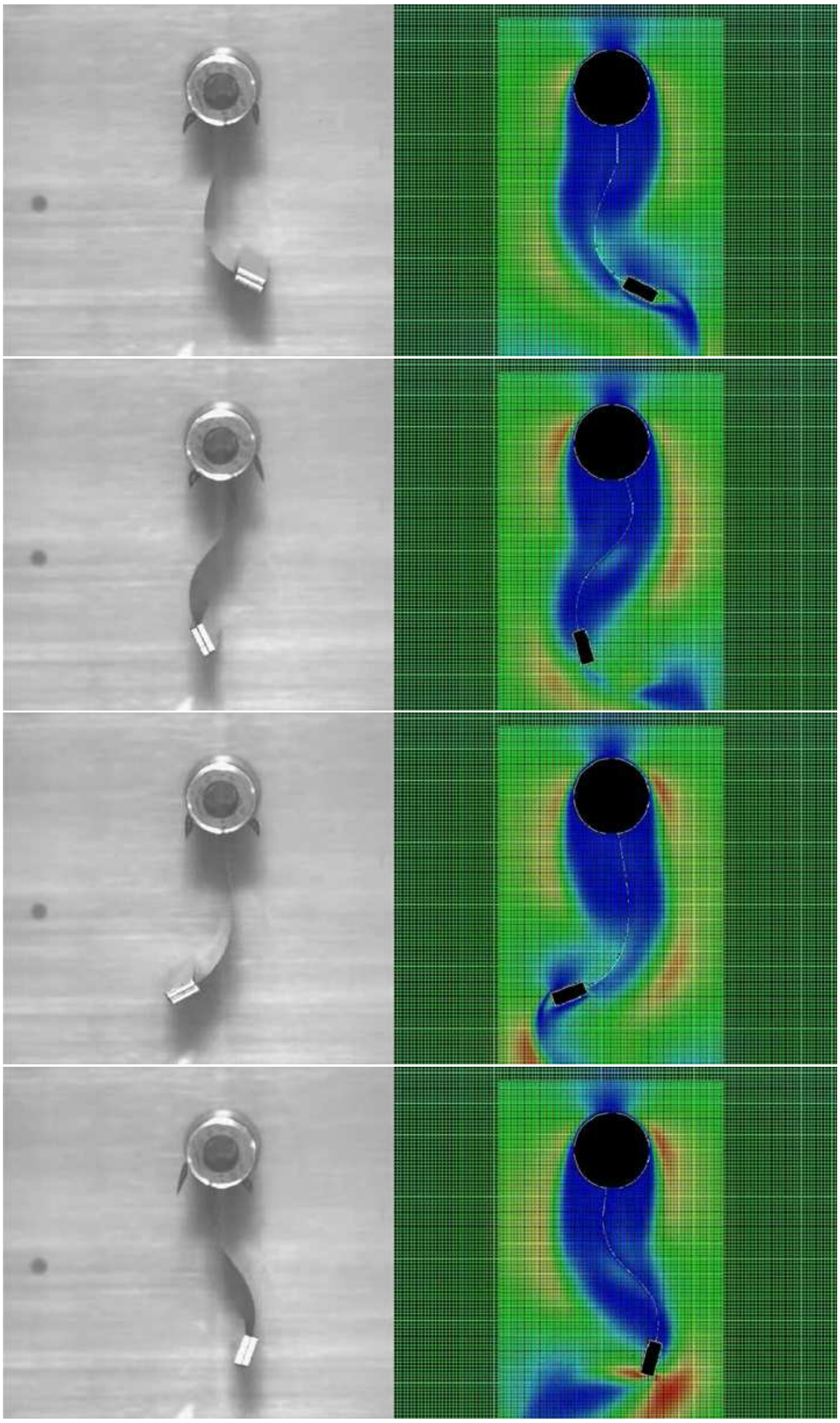


Figure 8.20: experimental Benchmark 2 $Re = 190$

relatively accurate results as shown next.

In order to evaluate the three-dimensional setup, the Benchmark presented by Bathe and Ledezma in [6] is used. It consists of a plate in cross flow as depicted in Figure 8.21. The Reynolds number is defined according to Equation (3.22) where the characteristic length L is chosen to be the height of the channel $H = 5[m]$ and the kinematic viscosity ν is computed according to Equation (3.20). In order to obtain a stationary fluid solution, the low Reynolds number of $Re = 10$ was chosen to be simulated first. Generally, the force acting on the structure can be estimated as i.e. given in [14]:

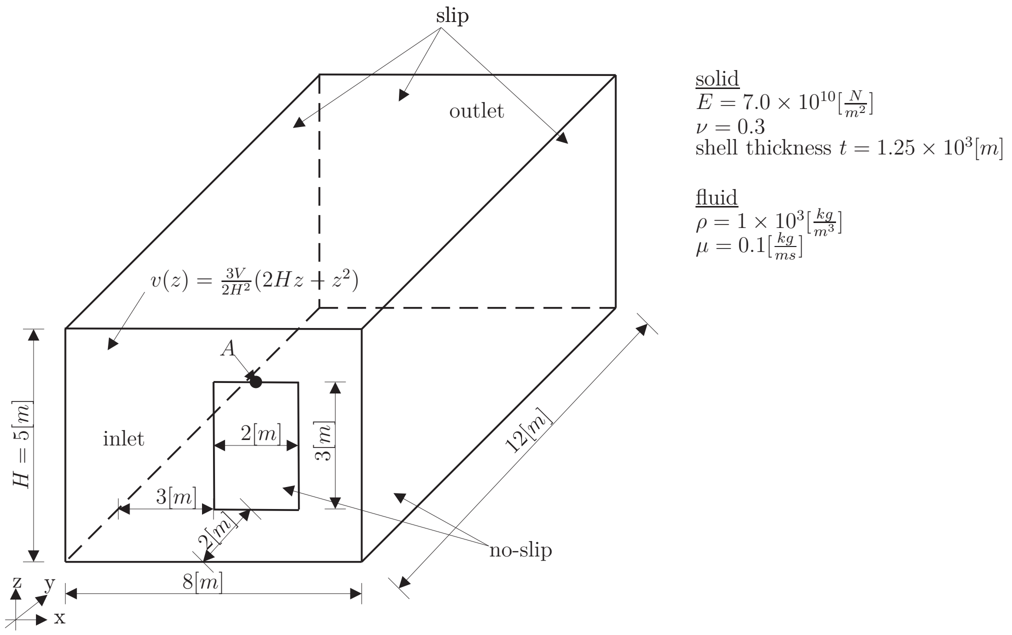

slip
outlet
solid
$E = 7.0 \times 10^{10}[\frac{N}{m^2}]$
$\nu = 0.3$
shell thickness $t = 1.25 \times 10^3[m]$
fluid
$\rho = 1 \times 10^3[\frac{kg}{m^3}]$
$\mu = 0.1[\frac{kg}{ms}]$
$v(z) = \frac{3V}{2H^2}(2Hz + z^2)$
A
$H = 5[m]$
inlet
$2[m]$
$3[m]$
$3[m]$
$2[m]$
$12[m]$
no-slip
$8[m]$
z
y
x

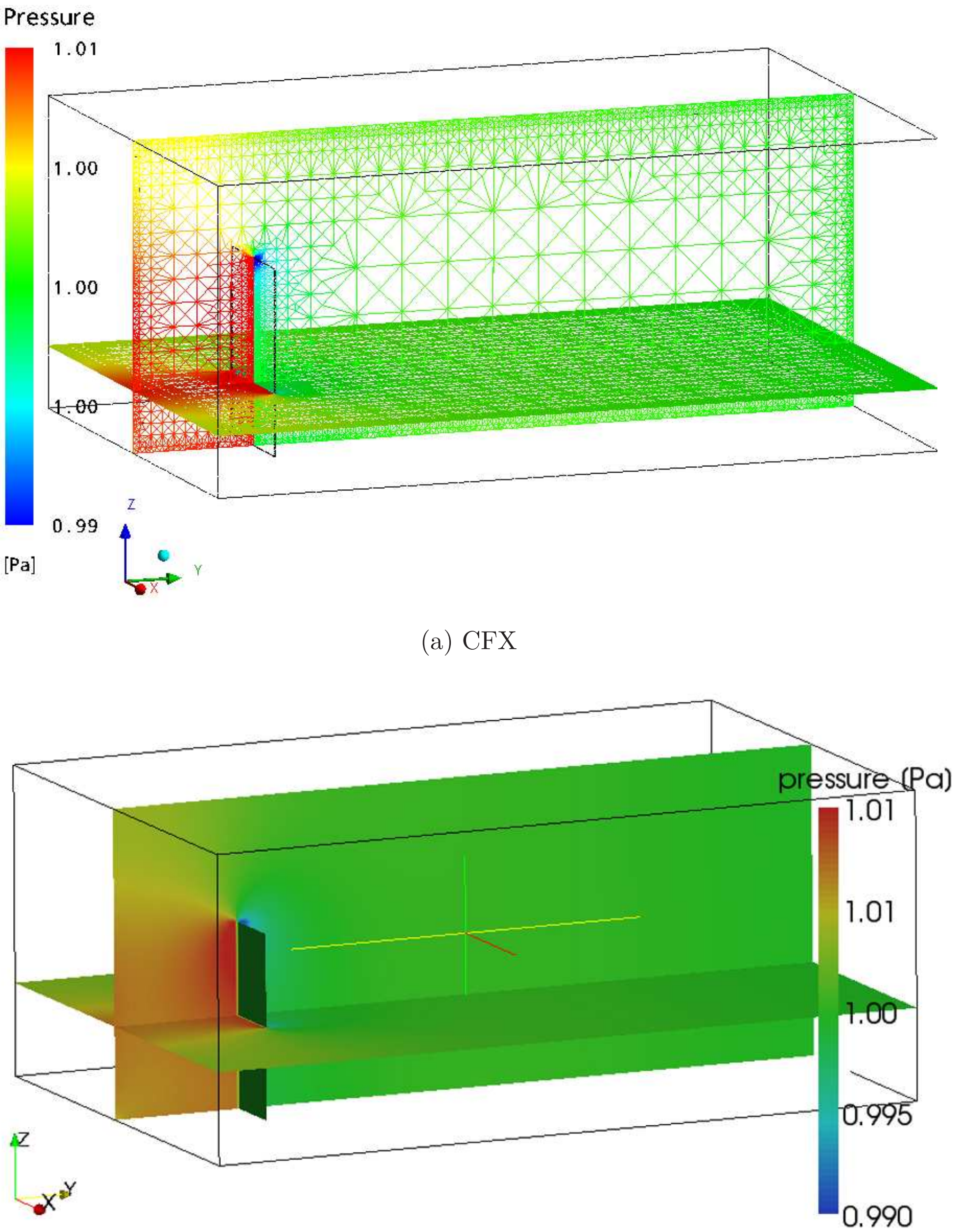
Figure 8.21: three-dimensional Benchmark according to [6]

$$F_d = \frac{1}{2} C_d A \rho v^2 \tag{8.12}$$

C_d is the drag coefficient, A is the area perpendicular to the flow, ρ is the density of the fluid and v^2 is the free field velocity of the fluid. The crucial point here is that C_d is not only dependent on the shape of the object but also on the Reynolds number. It is tabulated for standard cases e.g. in [97]. For a sphere at a $Re = 1$ the drag coefficient is $C_d \approx 140$ while at $Re \geq 1000$ it decreases to around 0.47. For a plate in cross flow at $Re \geq 1000$, $C_d \approx 1$. To the authors knowledge, the case of a clamped plate subject to a cross flow at $Re = 10$ is not tabulated. It has therefore been determined by a fully three-dimensional computation via CFX where the discretization was adapted to the pressure field in three steps. This resulted in a discretization of about 2 016 708 tetrahedral elements with 399 399 nodes. A view of the pressure field and its discretization as computed by CFX is depicted in Figure 8.22a. The pressure field as computed by LBM is depicted in Figure 8.22b.

From the steady state computation with CFX, the drag coefficient was determined to be $C_d = 9.46$. A rough estimate of the expected deflection at the top of the plate can be computed via application of the force given by Equation (8.12) to a clamped beam. Under the assumption of small displacements and a constant load, the analytical solution simply is:

$$u_y(A) = \frac{qL^4}{8EI}; \tag{8.13}$$



(a) CFX

(b) LBM

Figure 8.22: Pressure field of plate in cross flow computed by two different methods: (a) CFX: Visualization with CFX-postprocessor (b) LBM

where the load is $q = \dfrac{F_d}{A}$, the height of the beam is $L = 3[m]$. The stiffness EI is the plate stiffness and can be computed by:

$$EI = \frac{Et^3}{12(1-\nu^2)} \tag{8.14}$$

The first study is to evaluate the simplified stress computation and the correct detection of

$v[\frac{m}{s}]$	$u_y(A)[m]$ according to Equation (8.13)	$u_y(A)[m]$ computed via AdhoC-LBM
2.0×10^{-4}	1.53×10^{-4}	1.69×10^{-4}
2.0×10^{-3}	1.53×10^{-2}	1.65×10^{-2}
2.0×10^{-2}	1.53	1.24

Table 8.7: Benchmark Bathe at $Re = 10$, $Ma = 0.05$ and different inflow velocities

the neighbouring nodes next to the structure according to its normal. In order to achieve a range of different deflections at $Re = 10$, the Benchmark was computed with three different inflow velocities $v_1 = 2.0 \times 10^{-4}[\frac{m}{s}]$, $v_2 = 2.0 \times 10^{-3}[\frac{m}{s}]$ and $v_3 = 2.0 \times 10^{-2}[\frac{m}{s}]$ where the fluid viscosity was adjusted accordingly. The deflection should be in the range of the prediction via Equations (8.12) and (8.13) and scale quadratically with the inflow velocity as long as the deflection remains within the limits of geometrical linearity.

The structure was discretized by 180 elements with a polynomial degree of three in all directions and computed geometrically linear. To reach a steady state quickly, the mass of the structure was chosen to be $\rho_s = 1.0 \times 10^6[\frac{kg}{m^3}]$ and the first Eigenfrequency of the structure was damped out with a stiffness proportional damping. The fluid was discretized with a uniform mesh of $96 \times 144 \times 60$ nodes. The Mach number was chosen to be $Ma = 0.05$. The size of the interface mesh elements was chosen to be approximately equal to the fluid discretization. The results lie within the expected range of accuracy and are given in Table 8.7. The transient behaviour over time is depicted in Figure 8.23.

In a second step, the Reynolds number was raised to $Re = 500$ by means of increasing the inflow velocity to $v = 1.0 \times 10^{-2}[\frac{m}{s}]$, while all other parameters were kept constant. Unlike at a Reynolds number of $Re = 10$, a Reynolds number of $Re = 500$ leads to transient behaviour in the fluid. Vortexes detach at the tip of the plate and disperse into the domain. A snapshot of the streamlines is depicted in Figure 8.24a clearly demonstrating these vortexes. However, they only have a minor effect on the structure which settles at a quasi stationary deflection of $u_y(A) = 8.97 \times 10^{-2}[m] \pm 5.64 \times 10^{-4}[m]$. In contrast, Bathe and Ledezma [6] compute the deflection at Point A to be $u_y(A) = 6.6 \times 10^{-2}[m]$. In search for the true value, the Benchmark was recomputed with Ansys-CFX Multiphysics Version 12. In this setting, explicit coupling algorithms lead to instabilities and a two-way fluid-structure interaction analysis needed to be set up. This implicit coupling required an underrelaxation in the displacements according to Equation (4.27) with $\omega = 0.15$. The time step step was chosen to be $\Delta t = 2.5[s]$ for fluid and structure alike. The structure was discretized with 204 hexahedral elements. The fluid was discretized with 569 402 tetrahedral Finite Volume elements with 102 384 nodes as opposed to Bathe and Ledezma who used a mesh of 37*54*22 hexahedral elements in their finest resolution. The complete time history of the deflection of point A as computed by *Virtual Fluids* - AdhoC (LBM-*p*-FEM) is directly compared to the time history of Point A obtained by the computation with Ansys-CFX Multiphysics in Figure 8.24b.

The direct comparison as shown in Figure 8.24b is very strict. Different starting values, for

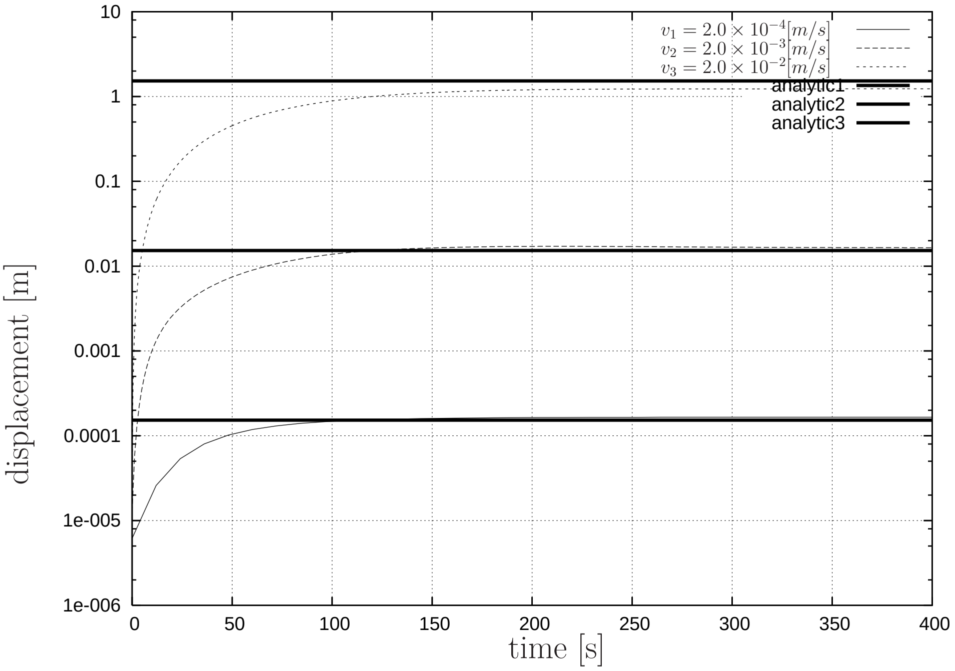


Figure 8.23: Plate in cross flow at $Re = 10$: y- deflection at point A obtained by LBM–p-FEM for different inflow velocities

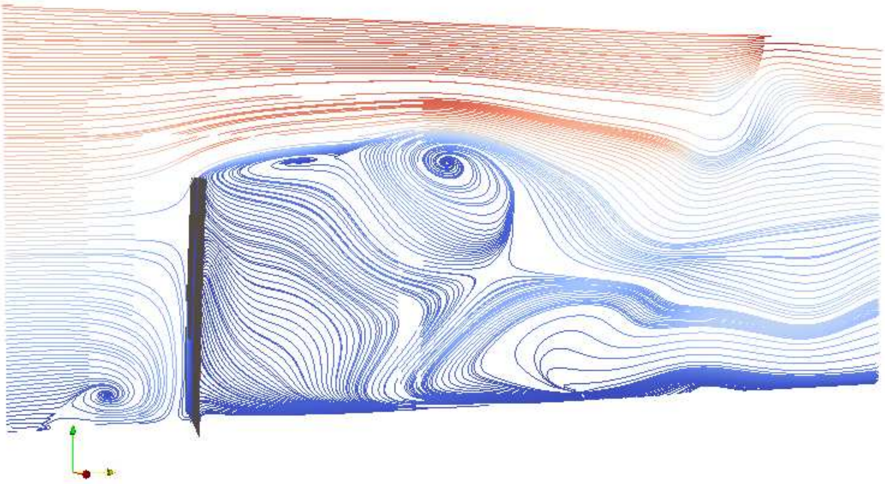
example, will almost always lead to different time-histories before the system has locked in[5]. This difference diminishes as time procedes. The deflection at point A is thus compared in its last oscillation only and amounts to $u_y(A) = 9.43 \times 10^{-2}[m] \pm 1.0 \times 10^{-3}[m]$ for the Ansys-CFX Multiphysics computation. It is thus 4.9% larger in the mean and a little less than double as large in its amplitude as compared to the solution computed by *Virtual Fluids* - AdhoC (LBM-p-FEM) .

These differences might be due to the fact that LBM discretization is still too compressible considering the intermediate Mach number of $Ma = 0.05$ as compared to the incompressible nature of the fluid discretization utilized in CFX. These lower deflections might also partly be attributed to the simpler and less exact next neighbor force transfer method chosen in the LBM-solver. However, the results agree quite well considering the complicated physics involved and the *completely different approaches* used for this comparison.

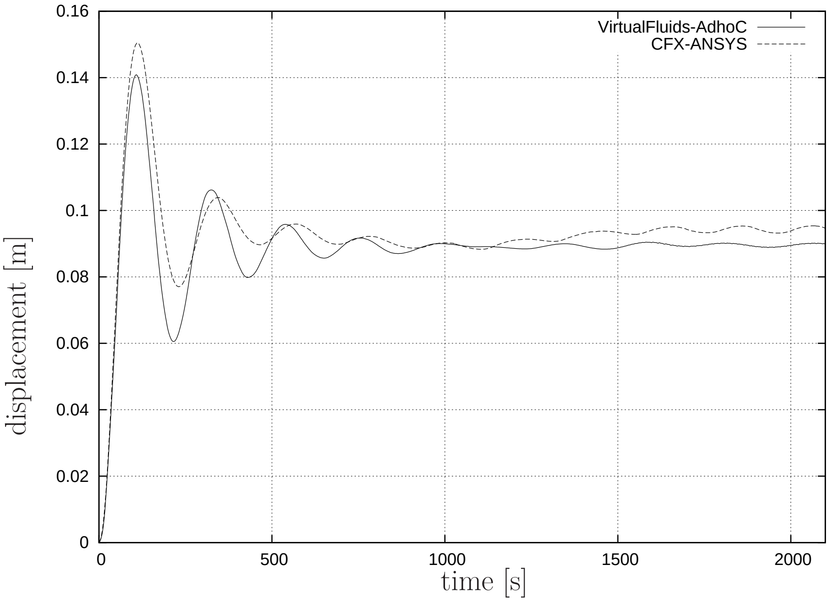
In a third test the Reynolds number was raised to $Re = 2500$ by increasing the fluid velocity at the inflow to $v = 5.0 \times 10^{-2}[\frac{m}{s}]$ while all other fluid and structural discretization parameters were kept constant w.r.t. the test case $Re = 500$. The resulting deflection at point A is depicted in Figure 8.25a and the streamlines are depicted in Figure 8.25b. The mean deflection for this case amounts to $u_y(A) = 1.36[m]$. As in the previous case where $Re = 500$, Bathe and Ledezma [6] again arrive at a smaller deflection and give $u_y(A) \approx 1.0[m]$. Unfortunately, no comparison with Ansys-CFX Multiphysics is available at the time of submission of this thesis[6].

[5] This is especially difficult in a turbulent, transient computation because it is virtually impossible to start with the same fluid state for both solvers.

[6] Part of the problem here is that such comparative studies are painfully time consuming, even on the most modern machines. The last run with Ansys-CFX Multiphysics for the case $Re = 500$, for example, took 4 days even though it was computed on an up to date machine and in parallel using 4 Intel Xeon E5405 processors



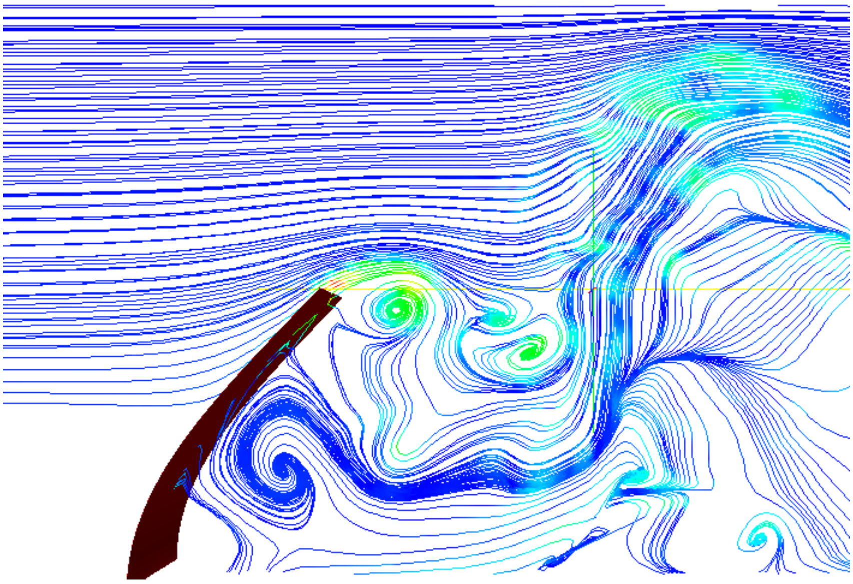
(a) Streamlines as computed by *Virtual Fluids* - AdhoC (LBM-*p*-FEM)



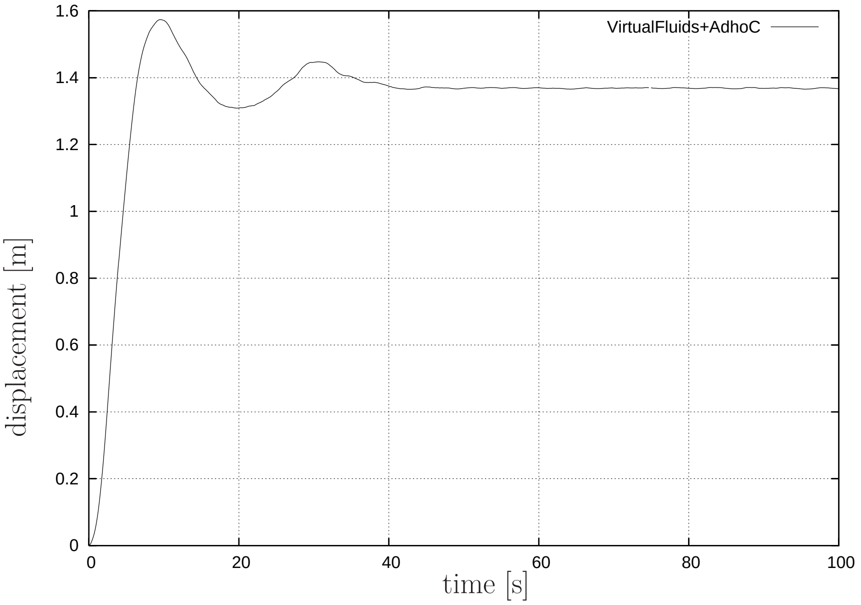
(b) Deflection at point A: Comparison of computations with *Virtual Fluids* - AdhoC (LBM-*p*-FEM) and Ansys-CFX Multiphysics

Figure 8.24: Plate in cross flow at $Re = 500$

each with 2GHz. This after many previous runs of trying to guess the correct, problem dependent settings such as a good underrelaxation factor, the proper time step, tight but not too tight convergence values, and all the other nitty gritty details one has to find out and pre set for such multyphysics computations.



(a) Streamlines



(b) Deflection at point A

Figure 8.25: Plate in cross flow at $Re = 2500$

Chapter 9

Beyond Benchmarks

9.1 Very large structural deflection

With the coupling setup being an Immersed Boundary Method where the LBM solver provides the fluid solution on its fixed background mesh, it comes as no surprise that well known advantages of coupling to fixed grid methods can be enjoyed, namely the ability to compute very large displacements without the necessity of remeshing. As shown in Figure 9.1 it is possible, for example, to let the structural component of the experimental benchmark described in Chapter 5.1 fall freely through the fluid domain. This can be realized via releasing the Dirichlet boundary conditions at the center of the cylinder. It is stressed that, once the boundary conditions are correctly imposed and the coupling setup is validated and verified, large deflections impose virtually no additional difficulty to this type of coupling.

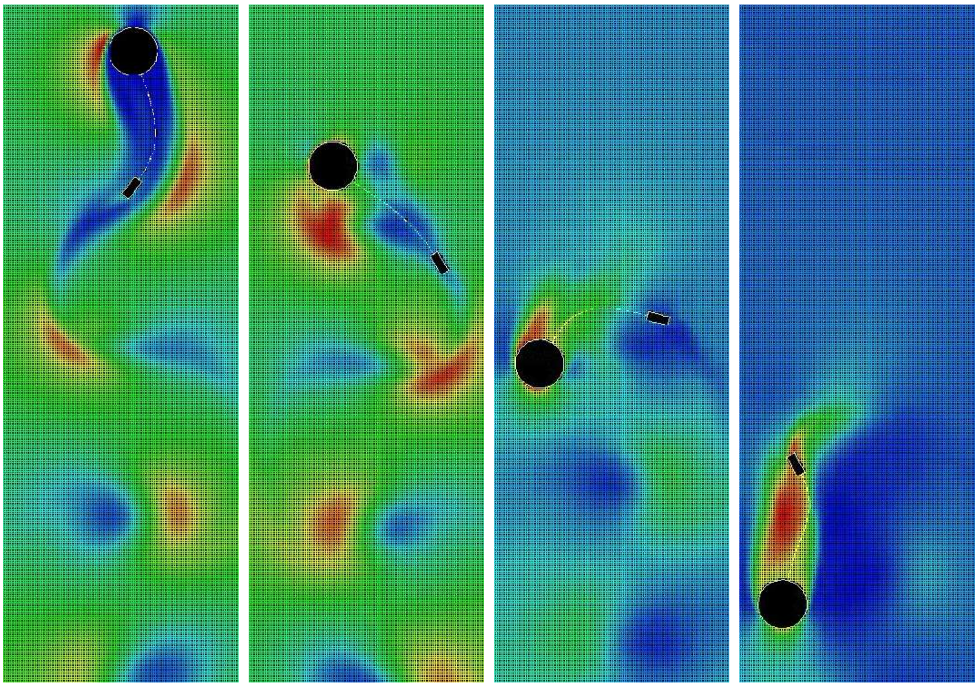


Figure 9.1: Very large displacement FSI: free falling experimental benchmark[5] ($Re = 140$)

9.2 A free surface example

This example is to show a typical benefit of a partitioned as opposed to a monolithic solution approach. The computation of other types of surface coupled problems are possible with only minor changes. Figure 9.2 shows the coupling between a high order solid tower hit by a free surface wave. The tower was computed by *AdhoC*^{4}and the fluid by *Virtual Fluids*.

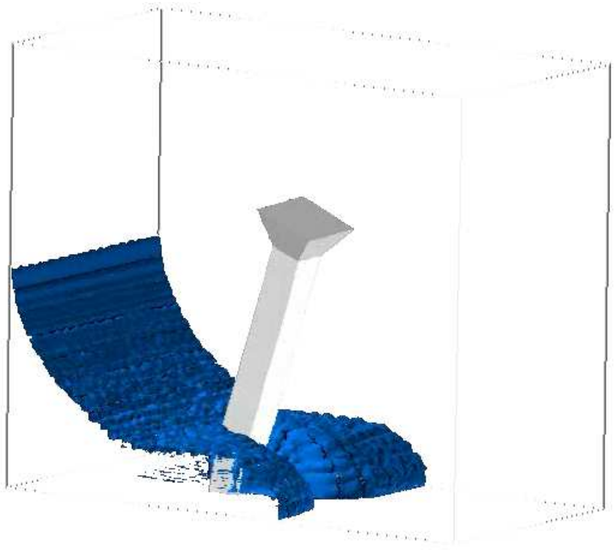


Figure 9.2: Tower hit by a wave

9.3 An example of practical interest

Large parts of this thesis deal with verification and validation of examples which are interesting for research purposes in order to gain confidence in the involved codes and the newly developed methods. However, having achieved this it is of interest as well to attempt to compute a very large example. The Millau bridge fits into this class of problems. This example is to serve as a motivation for future development and to discuss the principle applicability of the setup to problems of this size.

The Millau bridge is the world largest cable-stayed road-bridge spanning the valley of the river Tan in southern France [141]. A picture of the bridge in construction is depicted in Figure 9.3. It was built using a timed shifting method. It is prone to wind induced vibrations especially in the construction phase when the cantilever arm of the bridge is not yet supported by the subsequent bridge pier.

The computational model of this most critical situation is depicted in Figure 9.4. The structure was discretized by 239 Hexahedral elements of order 4 in all directions while 2.85×10^6 nodes were used for the LBM computation in *Virtual Fluids*. Figure 9.3 depicts this computational setup, the geometry of the bridge and the valley. The slices show the magnitude of the velocity vector. The Reynolds number was set to $Re = 1.0 \times 10^6$ and a Smagorinsky turbulence model [122] was applied. For examples of such size, three main problems arise: (a) The spatial resolution necessary to resolve the eddies leads to a large number of degrees of freedoms.

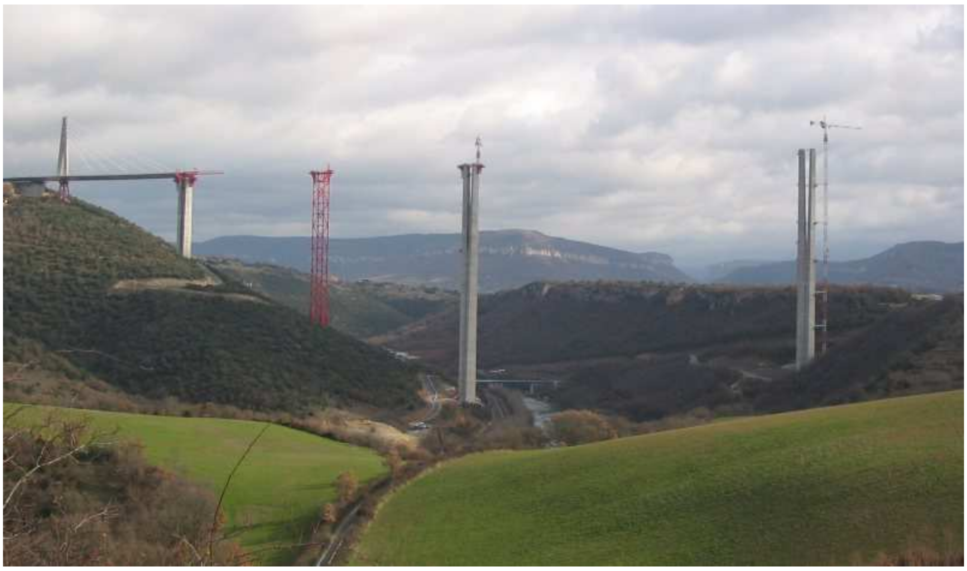


Figure 9.3: Millau Bridge in construction [141]

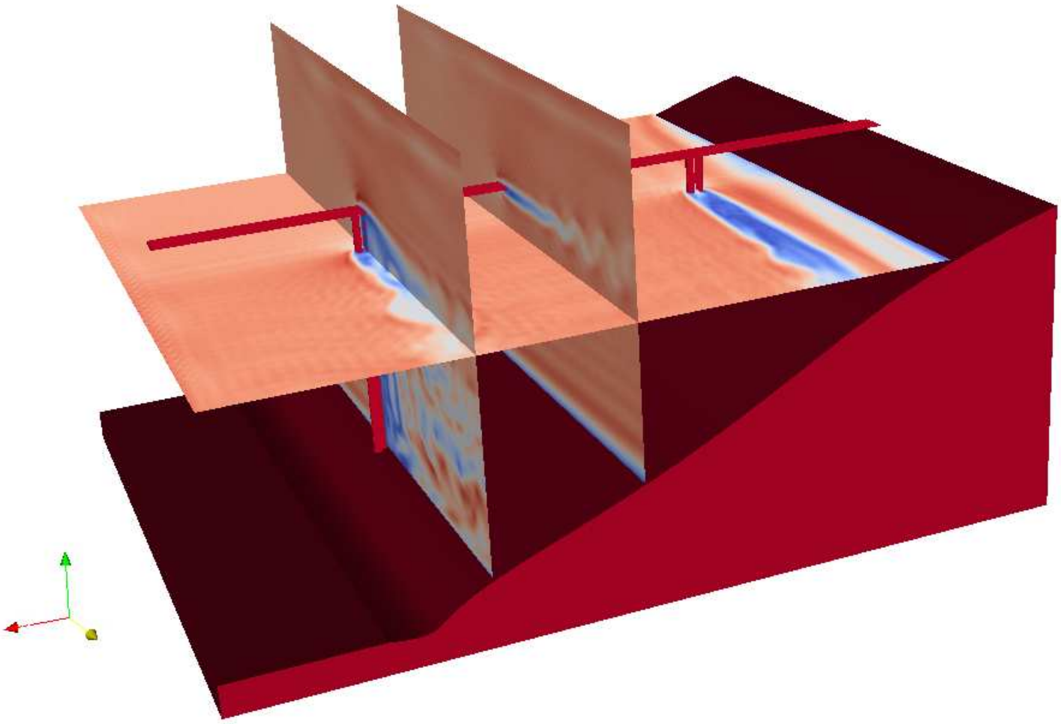


Figure 9.4: Millau Bridge modeled by AdhoC coupled to *Virtual Fluids*

Although 2.85×10^6 nodes for the model depicted in Figure 9.4 seems a lot, this is surely insufficient to resolve the boundary layers. With the computational domain having a hight of $400[m]$ the resolution along this height is only about 200 nodes resulting in approximately one node for every two meters or about 2.5 over the height of the cross section of the bridge. One way to reach an adequate resolution around the bridge is to utilize local refinement. This is already implemented in *Virtual Fluids* but was not applied here. Another possibility to model eddies which stem from obstacles with the same size as the grid resolution is to apply a cascaded LBM model; see for example [48] in which a vortex shedding behind a cylinder with the radius equal to the grid spacing is accurately modeled. (b) The critical wind speeds to be modeled might be beyond the limitations for the Mach number in the LBM model. The current LBM model is applicable for Mach numbers smaller than 0.1. In air this limits the maximum flow velocity to around $130[\frac{km}{h}]$. The remedy is to utilize LBM models which are also capable of higher Mach numbers. (c) The structural model might yet be to simple. For a realistic modeling, structural cable elements would need to be introduced. Generally, depending on the level of detail required, the structural solver, too, reaches its limits in terms of computational complexity. In an effort to reduce the structural complexity of the model, the bridge deck and pillars are currently modeled as one solid body. This can lead to an overestimated rotational stiffness due to an overestimated shear modulus. The computational models are currently refined step by step in order to be able to tackle these type of engineering problems.

Chapter 10

Summary and conclusions

This thesis treats selected aspects of partitioned fluid structure interaction for interface coupled problems where the structure was always discretized by Lagrange type elements of high order. There are two main approaches of this type of fluid-structure interaction: (a) either, the fluid is discretized via an *A*rbitrary *L*agrangean-*E*ulerian (ALE) formulation which allows a smooth transition between the moving structure and the fixed fluid boundaries. This class of fluid-structure interaction was treated in Chapter 6 where the commercial Finite Volume Solver CFX was utilized and in Chapter 7 where the research code $N\epsilon\kappa\tau\alpha\rho$ deploying a Spectral Element Method was applied. The second class of fluid-structure interaction (b) discretizes the fluid entirely on a fixed Cartesian grid. Here, the fluid was chosen to be discretized via the Lattice Boltzmann Method implemented in Virtual Fluids. All coupling setups were verified and / or validated against the benchmarks defined in Chapter 5 for two dimensions. Additionally, a three-dimensional verifictation was carried out against the Benchmark proposed by Bathe and Ledezma in [6]. Although deviations to the results published in [6] were observed, the obtained results could be verified against the commercial solver Ansys-CFX Multiphysics using a large number of degrees of freedom.

In this thesis, incompressible fluid solvers were coupled via setup (a). This called for an overall implicit formulation in order to lead to convergence. Explicit formulations diverged even for very small time steps. For the coupling to the commercial fluid solver CFX, the recently developed Interface-GMRES method by Michler [92] was selected among the manyfold of implicit coupling algorithms. It was evaluated and compared to a standard Gauss-Seidel type procedure. While promising results have been published for Interface-GMRES in connection with compressible fluids, a closer look disclosed weak points of the method. It was found that these, unfortunately, do not make it applicable in its present form in connection with incompressible flow solvers.

The coupling to spectral element methods, therefore, utilized a Gauss-Seidel type iteration for an implicit overall coupling algorithm. For this purpose, a new, implicit, stiffly stable time integration method needed to be introduced which was jointly developed in the framework of the Master Thesis of Papaioannou [99]. Applied to the benchmarks, the results agreed well with the ones obtained by other codes. Yet, differences came forth in regions with sharp gradients in the solution.

The coupling setup of type (b) as presented in Chapter 8 utilizes a Lattice Boltzmann Method. Based on the work of Krafczyk [78, 79], it was jointly developed in the course of this thesis in close corporation with Geller [77, 76, 51, 53, 118] and is the first of its kind. The Lattice-

Boltzmann Method turned out to be well suited for this type of fluid-structure interaction. Within this setup it was possible to:

- use an efficient, explicit overall coupling approach in a straight forward manner
- verify *and* validate the results against the purely computational and the experimental benchmarks, respectively
- compute examples in which the structure undergoes *very* large deflections, only limited by the size of the computational domain.

None of this could be achieved with the two other fluid discretizations utilized within this thesis. This is not to disqualify the other, well established methods. Some members of the research group DFG 493 surely reached all or at least some of these goals as well, also with newly developed approaches. Admittedly, part of the successful coupling of high order finite elements with the Lattice Boltzmann Method can be attributed to the disproportional amount of work put into this approach as opposed the other ones presented within this thesis. But this does not taint the main result of this work.

The main result of this work is that the following can be stated without doubt:

> The Lattice Boltzmann Method in connection with a structural discretization of high order is a promising combination for the solution of surface coupled problems in partitioned fluid-structure interaction with large displacements.

Stakes used to be against this statement at the beginning of the work for this thesis.

Bibliography

[1] http://www.cfm.brown.edu/crunch/nektar.html.

[2] Ansys CFX 10.0. *Solver: Theory*. http://www.ansys.com/Products/cfx.asp, 2005.

[3] V.V. Aristov. *Direct Methods for Solving the Boltzmann Equation and Study of Nonequilibrium Flows*. Springer, 2001.

[4] I. Babuška and M. Suri. Locking effects in the finite element approximation of elasticity problems. *Numerische Mathematik*, 62:439–463, 1992.

[5] I. Babuška and M. Suri. On locking and robustness in the finite element method. *SIAM Journal on Numerical Analysis*, 29:1261–1293, 1992.

[6] K.J. Bathe and G.A. Ledezma. Benchmark problems for incompressible fluid flows with structural interactions. *Computers and Structures*, 85:628–644, 2007.

[7] A. Beskok and T. C. Warburton. An Unstructured *hp* Finite-Element Scheme for Fluid Flow and Heat Transfer in Moving Domains. *Journal of Computational Physics*, 174:492–509, 2001.

[8] A. Beskok and T. C. Warburton. Arbitrary lagrangian eulerian analysis of a bidirectional micro pump using spectral elements. *International Journal of Computational Engineering Science*, 2:43–57, 2001.

[9] P.L. Bhatnagar, E.P. Gross, and M. Krook. A model for collision processes in gases. *Physical Review*, 94:511–525, 1954.

[10] N. Bodard and M. O. Deville. Fluid-structure interaction by the spectral element method. *Journal of Scientific Computing*, 27:123–136, 2006.

[11] J. Bonet and R.D. Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, New York, 2008.

[12] M. Bouzidi, M. Firdaouss, and P. Lallemand. Momentum transfer of a Boltzmann-Lattice fluid with boundaries. *Physics of Fluids*, 13(11):3452–3459, 2001.

[13] M. Brenk, H.-J. Bungartz, M. Mehl, and T. Neckel. Fluid-Structure Interaction on Cartesian Grids: Flow Simulation and Coupling Interface. In *Fluid-Structure Interaction, Modelling, Simulation and Optimisation*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 294–335. Springer, 2006.

[14] Encyclopedia Britannica. fluid mechanics. *Encyclopedia Britannica* `http://www.britannica.com/EBchecked/topic/211272/fluid-mechanics`, 2009.

[15] P.N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems of equations. *SIAM Journal of Scientific and Statistical Computing*, 11:450–481, 1990.

[16] C. Canuto, M. Y. Hussaini, A. Quateroni, and T.A. Zank. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, New York, 1993.

[17] P. Causin, J. F. 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, 2005.

[18] S. Chen and G. D. Doolen. Lattice Boltzmann method for fluid flows. *Annual Review Fluid Mechanics*, 30:329–364, 1998.

[19] J. Chung and G.M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of Applied Mechanics, Transactions ASME*, 60:371–375, 1993.

[20] M.A. Crisfield. *Non-linear finite element analysis of solids and structures, Volume 2*. John Wiley & Sons, 1997.

[21] B. Crouse, E. Rank, M. Krafczyk, and J. Tölke. A LB-based approach for adaptive flow simulations. *International Journal of Modern Physics B*, 17:109–112, 2003.

[22] A. de Boer, A. H. van Zuijlen, and H. Bijl. Review of coupling methods for non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 196:1515–1525, 2007.

[23] A. de Boer, A. H. van Zuijlen, and H. Bijl. Comparison of conservative and consistent approaches for the coupling of non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, page doi:10.1016/j.cma.2008.05.001, 2008.

[24] I. Demirdžić and M. Perić. Space conservation law in fnite volume calculations of fluid flow. *International Journal for Numerical Methods in Engineering*, 8:1037–1050, 1988.

[25] I. Demirdžić and M. Perić. Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries. *International Journal for Numerical Methods in Engineering*, 10:771–790, 1990.

[26] L. Demkowicz. *Computing with hp-adaptive finite elements: one and two dimensional elliptic and Maxwell problems*, volume 1. Chapman & Hall / CRC Applied Mathematics & Nonlinear Science, 2006.

[27] S. Deparis, M. Discacciati, and Quarteroni A. A Domain Decomposition Framework for Fluid-Structure Interaction Problems. In *Computational Fluid Dynamics 2004, Proceedings of the Third International Conference on Computational Fluid Dynamics, ICCFD3, Toronto, 12–16 july 2004*, volume 1, pages 41–58. Springer, Berlin-Heidelberg, 2004.

[28] D. d'Humières. Generalized lattice-Boltzmann equations. In B. D. Shizgal and D. P. Weave, editors, *Rarefied Gas Dynamics: Theory and Simulations*, volume 159 of *Prog. Astronaut. Aeronaut.*, pages 450–458, Washington DC, 1992. AIAA.

[29] D. d'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L. Luo. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 360(1792):437–451, 2002.

[30] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems.* John Wiley & Sons, 2003.

[31] A. Düster, H. Bröker, H. Heidkamp, U. Heißerer, S. Kollmannsberger, R. Krause, A. Muthler, A. Niggl, V. Nübel, M. Rücker, and D. Scholz. *AdhoC^4 – User's Guide.* Lehrstuhl für Bauinformatik, Technische Universität München, 2004.

[32] A. Düster, H. Bröker, and E. Rank. The p-version of the finite element method for three-dimensional curved thin walled structures. *International Journal for Numerical Methods in Engineering*, 52:673–703, 2001.

[33] A. Düster, L. Demkowicz, and E. Rank. High order finite elements applied to the discrete Boltzmann equation. *International Journal for Numerical Methods in Engineering*, 67:1094–1121, 2006.

[34] A. Düster, S. Hartmann, and E. Rank. p-fem applied to finite isotropic hyperelastic bodies. *Computer Methods in Applied Mechanics and Engineering*, 192:5147–5166, 2003.

[35] A. Düster, A. Niggl, V. Nübel, and E. Rank. A numerical investigation of high-order finite elements for problems of elasto-plasticity. *Journal of Scientific Computing*, 17:429–437, 2002.

[36] A. Düster, A. Niggl, and E. Rank. Applying the *hp-d* version of the FEM to locally enhance dimensionally reduced models. *Computer Methods in Applied Mechanics and Engineering*, 196:3524–3533, 2007.

[37] A. Düster and E. Rank. A p-version finite element approach for two- and three-dimensional problems of the J_2 flow theory with non-linear isotropic hardening. *International Journal for Numerical Methods in Engineering*, 53:49–63, 2002.

[38] A. Düster, D. Scholz, and E. Rank. *pq*-Adaptive solid finite elements for three-dimensional plates and shells. *Computer Methods in Applied Mechanics and Engineering*, 197:243–254, 2007.

[39] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational Differential Equations.* Cambridge University Press, 1996.

[40] 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*, 157:95–114, 1998.

[41] C. Farhat, M. Lesoinne, and P. Le Tallec. Geometric consevation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering*, 134:71–90, 2000.

[42] C. Farhat, K. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorythms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195:1973–2001, 2006.

[43] C. Felippa, K. Park, and J. DeRuntz. Stabilization of staggered solution procedures for fluid-structure interaction analysis. In T. Blytschko and T. Geers, editors, *Computational Methods for Fluid-Structure Interaction Problems*, volume 26, pages 95–124. American Society of Mechanical Engineers, New York, 1977.

[44] C. Felippa, K. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190:3247–3270, 2001.

[45] Ch. Förster, W.A. Wall, and E. Ramm. Artificial Added Mass Instabilities in Sequential Staggered Coupling of Nonlinear Structures and Incompressible Flows. *Computer Methods in Applied Mechanics and Engineering*, 196:1278–1293, 2007.

[46] Message Passing Interface Forum. MPI: A Message-Passing Interface Standard. *International Journal of Supercomputer Applications*, 8(3/4), 1994.

[47] P. Gamnitzer and W. Wall. An ALE-chimera method for large deformation fluid structure interaction. In P. Wesseling, E. Onate, and J. Periaux, editors, *Proceedings of the European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2006*, TU Delft, The Netherlands, 2006.

[48] M. Geier. De-aliasing and stabilization formalism of the cascaded lattice Boltzmann automaton for under-resolved high reynolds number flow. *International Journal for Numerical Methods in Engineering*, 56:1249–1254, 2008.

[49] S. Geller. *Ein explizites Modell für die Fluid-Struktur-Interaktion basierend auf LBM und p-FEM.* PhD thesis, Fakultät Architektur, Bauingenieurwesen und Umweltwissenschaften der Technischen Universität Carolo-Wilhelmina zu Braunschweig, 2010.

[50] S. Geller, M. Krafczyk, J. Tölke, S. Turek, and J. Hron. Benchmark computations based on Lattice-Boltzmann, Finite Element and Finite Volume Methods for laminar Flows. *Computers & Fluids*, 35:888–897, 2006.

[51] S. Geller, J. Tölke, and M. Krafczyk. Lattice Boltzmann Methods on Quadree-Type Grids for Fluid-Structure Interaction. In H.J. Bungartz and M. Schäfer, editors, *Fluid-Structure Interaction, Modelling, Simulation and Optimisation*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 270–293. Springer, 2006.

[52] S. Geller, J. Tölke, M. Krafczyk, S. Kollmannsberger, A. Düster, and E. Rank. A coupling algorithm for high order solids and lattice Boltzmann fluid solvers. In P. Wesseling, E. Onate, and J. Periaux, editors, *Proceedings of the European Conference on*

Computational Fluid Dynamics, ECCOMAS CFD 2006, TU Delft, The Netherlands, 2006.

[53] S. Geller, J. Tölke, M. Krafczyk, D. Scholz, A. Düster, and E. Rank. Simulation of bidirectional fluid-structure interaction based on explicit coupling approaches of Lattice Boltzmann and p-FEM solvers. In *Proceedings of the Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering*, Santorini, Greece, 2005.

[54] A. Gerstenberger and W. Wall. An eXtended Finite Element Method / Lagrange Multiplier based approach for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 197:1699–1714, 2008.

[55] A. Gerstenberger and W. Wall. Efficient treatment of moving interfaces on fixed grids for surface coupled problems. *International Journal for Numerical Methods in Engineering*, 57:1227–1248, 2008.

[56] I. Ginzburg, F. Verhaeghe, and D. d'Humières. Two-Relaxation-Time Lattice Boltzmann Scheme: About Parametrization,Velocity, Pressure and Mixed Boundary Conditions. *Communications In Computational Physics*, 3:427–478, 2008.

[57] J. P. Gomes and H. Lienhart. Experimental Study on a Fluid-Structure Interaction Reference Test Case. In H.J. Bungartz and M. Schäfer, editors, *Fluid-Structure Interaction, Modelling, Simulation and Optimisation*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 356–370. Springer, 2006.

[58] A. Halfmann. *Ein geometrisches Modell zur numerischen Simulation der Fluid-Struktur-Interaktion windbelasteter, leichter Flächentragwerke.* PhD thesis, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2002.

[59] D. Hänel. *Molekulare Gasdynamik.* Springer, Heidelberg, 2004.

[60] G. A. Hansen, R. W. Douglass, and A. Zardecki. *Mesh Enhancement: Selected Elliptic Methods, Foundations and Applications.* Imperial College Press, 2005.

[61] X. He and L.-S. Luo. Lattice Boltzmann model for the incompressible Navier-Stokes equation. *Journal of Statistical Physics*, 88:927–944, 1997.

[62] U. Heißerer. Solution of the semidiscrete equations of structural dynamics by the generalized-alpha method and its implementation in a p-fem code. Diploma thesis, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2001.

[63] U. Heißerer, S. Hartmann, A. Düster, and Z. Yosibash. On volumetric locking-free behavior of p-version finite elements under finite deformations. *Communications in Numerical Methods in Engineering (in press) DOI: 10.1002/cnm.1008*, 2007.

[64] H.M. Hilber, T. J. R. Hughes, and R.L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering & Structural Dynamics*, 5:283–292, 1977.

[65] C Hirsch. *Numerical Computation of Iinternal & External Flows.* Butterworth-Heinemann, Great Britan, 2007.

[66] L.-W. Ho. *A Legendre Spectral Element Method for Simulation of Incompressible Unsteady Viscous Free-Surface Flows.* PhD thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 1989.

[67] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Dover Publications, 2000.

[68] G.M. Hulbert. Computational structural dynamics. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 2, chapter 5, pages 169–193. John Wiley & Sons, 2004.

[69] S.R. Idelsohn, E. Oñate, F. Del Pin, and N. Calvo. Fluid-structure interaction using the particle finite element method. *Computer Methods in Applied Mechanics and Engineering*, 195(17-18):2100–2123, 2006.

[70] B. Irons and R. C. Tuck. A version of the Aitken Accelerator for Computer implementation. *International Journal for Numerical Methods in Engineering*, 1:275–277, 1969.

[71] E. Jansen, C. Whiting, and G. Hulbert. A generalized-alpha method for integrating the filtered navier-stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190(3-4):305 – 319, 2000.

[72] G.E. Karniadakis, M. Israeli, and S.A. Orszag. High order splitting methods for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 97:414–443, 1991.

[73] G.E. Karniadakis and S.J. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics.* Oxford University Press, New York, 2005.

[74] R.M. Kirby, Z. Yosibash, and G.E. Karniadakis. Towards stable coupling methods for high-order discretization of fluid-structure interaction: Algorithms and observations. *Journal of Computational Physics*, 223:489–518, 2007.

[75] S. Kollmannsberger, A. Düster, and E. Rank. Exact force transfer for high order finite element methods using intersected meshes. In *Proceedings of PVP 2007*, 12th International Symposium on Emerging Technologies in Fluids, Structures, and Fluid/Structure Interactions, within the ASME Pressure Vessel and Piping Conference, San Antonio, Texas, USA, 2007.

[76] S. Kollmannsberger, S. Geller, A. Düster, J. Tölke, M. Krafczyk, and E. Rank. Fluid-Structure Interaction based on Lattice Boltzmann and p-FEM: Verification and Validation. In *Proceedings of the International Conference on Computational Methods for Coupled Problems in Science and Engineering*, Ischia Island, Italy, 2009.

[77] S. Kollmannsberger, S. Geller, A. Düster, J. Tölke, C. Sorger, M. Krafczyk, and E. Rank. Fixed-grid fluid-structure interaction in two dimensions based on a partitioned Lattice Boltzmann and p-FEM approach. *International Journal for Numerical Methods in Engineering*, 79(7):817–845, 2009.

[78] M. Krafczyk. *Gitter-Boltzmann-Methoden: Von der Theorie zur Anwendung.* Postdoctoral thesis, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2001.

[79] M. Krafczyk, J. Toelke, E. Rank, and M. Schulz. Two-dimensional simulation of fluid-structure interaction using lattice-boltzmann methods. *Computers and Structures*, 79:2031–2037, 2001.

[80] R. Krause and E. Rank. Multiscale computations with a combination of the h- and p-versions of the finite-element method. *Computer Methods in Applied Mechanics and Engineering*, 192:3959–3983, 2003.

[81] D. Kuhl and M. A. Crisfield. Energy-Conserving and Decaying Algorithms in Nonlinear Structural Dynamics. *International Journal for Numerical Methods in Engineering*, 45:569–599, 1999.

[82] U. Küttler and W. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 1(43):61–72, 2008.

[83] U. Küttler and W. Wall. Vector extrapolation for strong coupling fluid-structure interaction solvers. *Journal of Applied Mechanics*, 2(76), 2009.

[84] P. Lallemand and Luo L.-S. Lattice boltzmann method for moving boundaries. *Journal of Computational Physics*, 184:406–421, 2003.

[85] T. Lee and C.L. Lin. A Characteristic Galerkin method for Discrete Boltzmann Equation. *Journal of Computational Physics*, 171:336–356, 2001.

[86] A. Legay, J. Chessa, and T. Belytschko. An Eulerian-Lagrangian method for fluid-structure interaction based on level sets. *Computer Methods in Applied Mechanics and Engineering*, 195:2070–2087, 2006.

[87] R. Löhner, J. D. Baum, E. Mestreau, D. Sharov, C. Charman, and D Pelessone. Adaptive embedded unstructured grid methods. *International Journal for Numerical Methods in Engineering*, 60:641–660, 2004.

[88] R. Löhner, Joseph D. Baum, and Orlando A. Soto. On Some Open Problems in Fluid-Structure Interaction. In *International Workshop on Fluid-Structure Interaction*, pages 343–354, Kassel University Press, 2009.

[89] W.E. Lorensen and H.E. Cline. A Discontinuous Galerkin Method for the Navier-Stokes Equations. *ACM SIGGRAPH Computer Graphics archive*, 21:163–169, 1987.

[90] X.J. Luo. *An automatic adaptive directional variable p-version method in 3D curved domains.* PhD thesis, Rensselaer Polytechnic Institute, Troy, New York, 2005.

[91] R. Mei, D. Yu, W. Shyy, and L. Lou. Force evaluation in the lattice Boltzmann method involving curved geometry. *Physical Review E*, 65:041203, 2002.

[92] 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*, 47:1189–1195, 2005.

[93] C. Michler, H. van Brummelen, R. Hulshoff, and R. de Borst. The relevance of conservation for stability and accuracy of numerical methods for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 192:4195–4215, 2003.

[94] R. Mittal and G. Iaccarino. Immersed Boundary Method. *Annual Review Fluid Mechanics*, 37:239–260, 2005.

[95] D. P. Mock. *Partitionierte Lösungsansätze in der Strukturdynamik und der Fluid-Struktur-Interaktion.* PhD thesis, Institut für Baustatik, Universität Stuttgart, 2001.

[96] N.M. Newmark. A numerical method for structural dynamics. *Journal of Engineering Mechanics (ASCE)*, 85:67–94, 1959.

[97] H. Oertel, Böhle M., and U. Dohrmann. *Strömnungsmechanik.* Vierweg+Teubner, 5. edition, 2009.

[98] N. Oreskes, K. Shrader-Frechette, and K. Belitz. Verification, validation, and confirmation of numerical models in the earth sciences. *Science*, 263:641–648, 1994.

[99] I. Papaioannou. *Fluid-structure interaction with high order finite element methods.* Master thesis, Chair for Computation in Engineering, 2007.

[100] K. Park and C. Felippa. Partitioned analysis of coupled systems. In T. Blytschko and T. Hughes, editors, *Computational Methods for Transient Analysis*, chapter 3, pages 157–219. North-Holland,Amsterdam- New York, 1984.

[101] G. Peng, H. Xi, Duncan C., and S.-H. Chou. Lattice Boltzmann method on irregular meshes. *Physical Review E*, 58(4):4124–4127, 1998.

[102] G. Peng, H. Xi, Duncan C., and S.-H. Chou. Finite volume scheme for the lattice Boltzmann method on unstructured meshes. *Physical Review E*, 59(4):4675–4682, 1999.

[103] C. Peskin. The immersed boundary method. *Acta Numerica*, 11:1–39, 2002.

[104] C. Petersen. *Dynamik der Baukonstruktionen.* Vieweg & Sohn, 2000.

[105] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aroelastic problems – part II: Energy transfer analysis and three dimensional applications. *Computer Methods in Applied Mechanics and Engineering*, 190:3147–3170, 2001.

[106] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aroelastic problems – part I: Model problem, theory and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering*, 124:79–112, 1995.

[107] Y.H. Qian, D. d'Humières, and P. Lallemand. Lattice BGK models for Navier-Stokes equations. *Europhysics Letters*, 17(6):479–484, 1992.

[108] E. Rank, A. Düster, V. Nübel, K. Preusch, and O.T. Bruhns. High order finite elements for shells. *Computer Methods in Applied Mechanics and Engineering*, 194:2494–2512, 2005.

[109] E. Rank and R. Krause. A multiscale finite-element-method. *Computers & Structures*, 64:139–144, 1995.

[110] B.D. Reddy. *Introductory Functional Analysis with Applications to Boundary Value Problems and Finite Elements*. Springer-Verlag, 1998.

[111] M. Renwei, L.S. Luo, P. Lallemand, and D d'Humières. Consistent Initial Conditions for LBE Simulations. *Computers & Fluids*, 35:855–862, 2006.

[112] Kirby R.M., Z. Yosibash, and G.E. Karniadakis. High order methods for fluid-structure interaction simulations. In *Proceedings of the Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering*, CIMNE, Barcelona, 2005.

[113] M. Schäfer, M. Heck, and S. Yigit. An Implicit Partitioned Method for the Numerical Simulation of Fluid-Structure Interaction. In H.J. Bungartz and M. Schäfer, editors, *Fluid-Structure Interaction, Modelling, Simulation and Optimisation*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 171–194. Springer, 2006.

[114] D. Scholz. *An anisotropic p-adaptive method for linear elastostatic and elastodynamic analysis of thin-walled and massive structures.* Dissertation, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2006.

[115] D. Scholz, A. Düster, and E. Rank. Fully three-dimensional modelling of fluid-structure interaction problems by using high order finite elements for the structural simulation. In *Proceedings of PVP 2004*, 8th International Symposium on Emerging Technologies in Fluids, Structures, and Fluid/Structure Interactions, within the ASME Pressure Vessel and Piping Conference, San Diego, California, USA, 2004.

[116] D. Scholz, A. Düster, and E. Rank. Model-adaptive fluid-structure interaction using high order structural elements. In *Proceedings of the Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering*, Santorini, Greece, 2005.

[117] D. Scholz, A. Düster, and E. Rank. Model-adaptive structural FEM computations for fluid-structure interaction. In *Proceedings of the Third M.I.T. Conference on Computational Fluid and Solid Mechanics*, Cambridge, USA, 2005.

[118] D. Scholz, S. Kollmannsberger, A. Düster, and E. Rank. Thin Solids for Fluid-Structure Interaction. In H.J. Bungartz and M. Schäfer, editors, *Fluid-Structure Interaction, Modelling, Simulation and Optimisation*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 294–335. Springer, 2006.

[119] X. Shan and H. Chen. Lattice Boltzmann model for simulating flows with multiple phases and components. *Physical Review E*, 47:1815—-1819, 1993.

[120] Xing Shi and Siak Piang Lim. A LBM-DLM/FD method for 3D fluid-structure interactions. *Journal of Computational Physics*, 226:2028–2043, 2007.

[121] G. Sieber. *Numerical Simulation of Fluid-Structure Interaction using Loose Coupling Methods.* PhD thesis, TU Darmstadt, `http://tuprints.ulb.tu-darmstadt.de/epda/000254/`, 2002.

[122] J. Smagorinsky. General circulation experiments with the primitive equations – i the basic experiment. *Monthly Weather Review*, 91:99–164, 1963.

[123] Ch. Sorger. Abschnittsweise Lastintegration bei gekoppelten Problemen mit nicht zusammenfallenden Finite-Elemente-Netzen. Diploma thesis, Lehrstuhl für Bauinformatik, Fakultät fürBauingenieurwesen, TU-München, 2005.

[124] S.P. Spekreijse. Elleptic generation systems. In J. F. Thompson, B. K. Soni, and N. P. Weatherill, editors, *Handbook of Grid Generation.* CRC Press, 1999.

[125] B.A. Szabó and I. Babuška. *Finite element analysis.* John Wiley & Sons, 1991.

[126] B.A. Szabó, A. Düster, and E. Rank. The p-version of the Finite Element Method. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 1, chapter 5, pages 119–139. John Wiley & Sons, 2004.

[127] P. D. Thomas and C. K. Lombard. Geometric conservation law and its application to flow computations on moving grids. *Journal of the American Institute of Aeronautics and Astronautics*, 17:1030–1037, 1979.

[128] J. F. Thompson, B. K. Soni, and N. P. Weatherill, editors. *Handbook of Grid Generation.* CRC Press, 1999.

[129] J. Tölke. *Gitter-Boltzmann-Verfahren zur Simulation von Zweiphasenströmungen.* PhD thesis, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2001.

[130] J. Tölke, S. Freudiger, and M. Krafczyk. An adaptive scheme using hierarchical grids for lattice boltzmann multi-phase flow simulations. *Computers & Fluids*, 35(8–9):820–830, 2006. Proceedings of the First International Conference for Mesoscopic Methods in Engineering and Science.

[131] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory.* Springer-Verlag, Berlin, Heidelberg, 2005.

[132] S. Turek and J. Hron. Proposal for Numerical Benchmarks for Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow. In H.J. Bungartz and M. Schäfer, editors, *Fluid-Structure Interaction, Modelling, Simulation and Optimisation*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 371–385. Springer, 2006.

[133] S. Turek and M. Schäfer. Benchmark computations of laminar flow around cylinder. In E. H. Hirschel, editor, *Notes on Numerical Fluid Mechanics*, volume 52. Vieweg-Verlag, 1996.

[134] S. Ubertini, G. Bella, and S. Succi. Lattice Boltzmann method on unstructured grids: Further developments. *Physical Review E*, 68:016701–1–016701–10, 2003.

[135] S. Ubertini, S. Succi, and G. Bella. Lattice Boltzmann schemes without coordinates. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 362:1763–1771, 2004.

[136] H. van Brummelen, R. Hulshoff, and R. de Borst. Energy conservation under incompatibility for fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 192:2727–2748, 2003.

[137] K. Vemaganti and J.T. Oden. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering*, 134:71–90, 1996.

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

[139] W. Wall, P. Gammnitzer, and A. Gerstenberger. Fluid-structure interaction approaches on fixed grids based on two different domain decomposition ideas. *International Journal of Compuational Fluid Dynamics*, 22:411–427, 2008.

[140] W.A. Wall. *Fluid-Struktur-Interaktion mit stabilisierten Finiten Elementen.* PhD thesis, Universität Stuttgart, Germany http://elib.uni-stuttgart.de/opus/volltexte/2000/623/, 1999.

[141] Wikipedia. Viaduc de millau. http://de.wikipedia.org/wiki/Viaduc_de_Millau.

[142] W.L. Wood, M. Bossak, and O.C. Zienkiewicz. An alpha modification of newmark's method. *International Journal for Numerical Methods in Engineering*, 5:1562–1566, 1981.

[143] P. Wriggers. *Nonlinear Finite-Element-Methods.* Springer-Verlag, 2009.

[144] H. Xi, G. Peng, and S.-H. Chou. Finite-volume lattice Boltzmann method. *Physical Review E*, 59(5):6202–6205, 1999.

[145] H. Xi, G. Peng, and S.-H. Chou. Finite-volume lattice Boltzmann schemes in two and three dimensions. *Physical Review E*, 60(3):3380–3388, 1999.

[146] Michael T. Heath Xiangmin Jiao. Common-refinement-based data transfer between non-matching meshes in multiphysics simulations. *International Journal for Numerical Methods in Engineering*, 61:2402–2427, 2004.

[147] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method – Its Basis and Fundamentals*, volume 1. Butterworth-Heinemann, 6th edition, 2005.