Published in **International Journal for numerical Methods in Engineering**, Vol 104, Issue 5, pp. 382-390, 2015

A Monolithic Geometric Multigrid Solver for Fluid-Structure Interactions in ALE formulation

Thomas Richter*

We present a monolithic geometric multigrid solver for fluid-structure interaction problems in Arbitrary Lagrangian Eulerian coordinates. The coupled dynamics of an incompressible fluid with nonlinear hyperelastic solids gives rise to very large and ill conditioned systems of algebraic equations. Direct solvers usually are out of question due to memory limitations, standard coupled iterative solvers are seriously affected by the bad condition number of the system matrices. The use of partitioned preconditioners in Krylov subspace iterations is an option, but the convergence will be limited by the outer partitioning. Our proposed solver is based on a Newton linearization of the fully monolithic system of equations, discretized by a Galerkin finite element method. Approximation of the linearized systems is based on a monolithic GMRES iteration, preconditioned by a geometric multigrid solver. The special character of fluid-structure interactions is accounted for by a partitioned scheme within the multigrid smoother only. Here, fluid and solid field are segregated as Dirichlet-Neumann coupling. We demonstrate the efficiency of the multigrid iteration by analyzing 2d and 3d benchmark problems. While 2d problems are well manageable with available direct solvers, challenging 3d problems highly benefit from the resulting multigrid solver.

1 Introduction

The interaction of an incompressible fluid with an elastic solid gives rise to a complex coupled system of partial differential equations. Coupling in fluid-structure interaction (fsi) problems usually takes place at a common interface between solid and fluid, that both live in separate domains, $\mathcal{S} \subset \mathbb{R}^d$ and $\mathcal{F} \subset \mathbb{R}^d$. The coupling is realized by means of continuity of stresses (the dynamic condition) and velocities (the kinematic condition) on the interface $\mathcal{I} = \partial\mathcal{F} \cap \partial\mathcal{S} \subset \mathbb{R}^{d-1}$. One technique to tackle fsi problems is the *partitioned approach*, where both subproblems are considered as separate entities and where the coupling is enforced by means of boundary conditions on the interface. In strongly coupled schemes, an outer iteration assures convergence to the coupled solution. Partitioned approaches are well described in literature, see [LM01] or the review article [HWL12]. They have the benefit, that existing solvers can be applied for the efficient approximation of the two sub-systems. Partitioned approaches are state of the art for real-world applications.

*Institute of Applied Mathematics, University of Heidelberg, Im Neuenheimer Feld 294, 69120 Heidelberg, Germany, thomas.richter@iwr.uni-heidelberg.de

For incompressible flows, partitioned approaches lack stability due to the *added-mass effect* [CGN05, FWR07]. Many sub-iterations can be required in every time-step to satisfy the interface coupling conditions [HHB08, van11]. While acceleration techniques like Aitken extrapolation [Hei98, MW01, KGF+10] improves the convergence, the numerical effort is still substantial.

In contrast, *monolithic approaches* consider the coupled problem as one entity. Hence, the coupling conditions as well as the two sub-fields are part of a mathematical problem description. This approach allows the use of implicit discretization techniques and strong coupled solvers (like Newton and multigrid) for the whole system. The drawback of course is the lack of standard tools available to tackling the coupled problem. Coupled fsi systems are an enormous challenge to numerical techniques, as discretization results in linear systems with very bad conditioning. Various factors play a role: different scales of material parameters (e.g. viscosities at about 10^{-3} and Young modulus of 10^6), the already mentioned *added-mass effect*, stiffness due to temporal effects, caused by the coupling of the Navier-Stokes problem (of parabolic type) with the elastic structure equation (of hyperbolic type) and of course the bad conditioning that arises from the discretization of second order spatial differential operators. Several contributing elements however stem from the coupling of the two fields.

Numerical simulation of realistic application problems is a very challenging task due to different factors:

1. The complex coupling calls for either very small time-steps or strongly coupled implicit discretization and solution schemes with enormous demands in terms of memory consumption and cpu times.

2. In the three dimensional case, the full system of fluid-structure interactions consists of a system of partial differential with a total of seven coupled components. On the interface, three fluid- and solid-velocities, three fluid- and solid-deformations as well as the fluid pressure component couple.

3. Problems with large deformation bring along technical difficulties regarding the handling of moving domains and meshes and give rise to additional nonlinearities.

4. The fluid problem alone can be a major challenge, if large Reynolds numbers, strong anisotropies or complex material laws are considered.

1.1 Analysis of a benchmark-problem

To start the discussion, we first present two different test problems to be used throughout this article. First, we consider the non-stationary *fsi-3 benchmark* problem of Hron and Turek [HT06c], a 2d test case featuring large deformation and stability problems caused by the *added-mass effect*. Second, we choose a three dimensional test case with smaller deformation. In the following, we describe the full configuration including all problem parameters for both test cases. Figure 1 shows a sketch of the geometry describing both problems. Both configurations are driven by an inflow condition $\mathbf{v}_f = \mathbf{v}_{\text{in}}$ for the velocity on Γ_{in}:

$$\mathbf{v}_{\text{in}}^{\text{2d}}(y) = \frac{y(H-y)}{(H/2)^2} \cdot \frac{3}{2}\bar{\mathbf{v}}_{\text{mean}}^{\text{2d}}, \quad \mathbf{v}_{\text{in}}^{\text{3d}}(y,z) = \frac{9}{8}\frac{y(H-y)(H^2-z^2)}{(H/2)^2 H^2} \cdot \frac{9}{8}\bar{\mathbf{v}}_{\text{mean}}^{\text{3d}},$$

with average inflow velocities of $\bar{\mathbf{v}}_{\text{mean}}^{\text{2d}} = 2m/s$ and $\bar{\mathbf{v}}_{\text{mean}}^{\text{3d}} = 1m/s$. Both profiles are temporally smoothed to give a smooth transition from $\mathbf{v} = 0$ at $t = 0$ to the maximum velocity at time $t = 2s$,

Mesh Level (2d)	2	3	4	5	6	7
Unknowns	5 260	20 640	80 960	320 640	1 276 160	5 091 840
Avg. conv. rate	0.14	0.29	> 0.99	> 0.99	> 0.99	> 0.99
Memory usage	19 MB	48 MB	—	—	—	—
Avg. Time	0.12 sec	0.72 sec	—	—	—	—

Mesh Level (3d)	2	3	4	5
Unknowns	18 711	131 495	983 367	7 600 775
Avg. conv. rate	0.086	0.067	0.094	0.33
Memory usage	156 MB	1.0 GB	7.8 GB	64 GB
Avg. Time	1.23 sec	10.17 sec	120.15 sec	2399 sec

Table 1: Number of unknowns, average convergence rates (see Remark 2), memory usage and average computational time for linear solution with monolithic geometric multigrid solver using a fully coupled block-wise incomplete decomposition of the smoother. In the 2D case, there was no convergence starting from mesh-level 4.

Problem configuration	2D	3D
Fluid density ρ_f	$10^3\ kg/m^3$	$10^3\ kg/m^3$
Kinematic viscosity ν_f	$10^{-3}\, m^2/s$	$10^{-3}\, m^2/s$
Average inflow velocity $\bar{v}_{\text{mean}}$	$2\, m/s$	$1\, m/s$
Solid density ρ_s	$10^3\, kg/m^3$	$10^3\, kg/m^3$
Shear modulus μ_s	$2 \cdot 10^6\, kg/ms^2$	$5 \cdot 10^5\, kg/ms^2$
Poisson ratio ν_s	0.4	0.4

Table 2: Description of the two benchmark problems

by multiplying with $\alpha(t) \in \mathbb{R}$ given by

$$\alpha(t) = \begin{cases} \frac{1}{2}\left(1 - \cos(\pi t/2)\right) & t < 2 \\ 1 & t \geq 2 \end{cases}.$$

On the outflow boundary Γ_{out} we prescribe the do-nothing condition

$$\rho_f \nu_f \vec{n} \cdot \nabla \mathbf{v}_f - p\vec{n} = 0, \tag{1}$$

see [HRT92], on the walls Γ_{wall} a no-slip condition $\mathbf{v}_f = 0$. In the case of the 3d-configuration, we consider a symmetry condition at Γ_{sym}. In both cases, the solid is firmly attached to the boundary at Γ_{base} by prescribing Dirichlet conditions for velocity and deformation $\mathbf{v}_s = \mathbf{u}_s = 0$. In Table 2, we collect all parameters, that completely describe the settings.

The 2d-case is challenging due large deformations, that makes up about 50% of the fluid domain. The 3d-case is less demanding in this respect, as deformations are small. This reduces the effects of nonlinearities as well as the role of the ALE-mapping of the fluid problem. We start by collecting convergence rates of a fully monolithic multigrid solver in Table 1 for both problems.

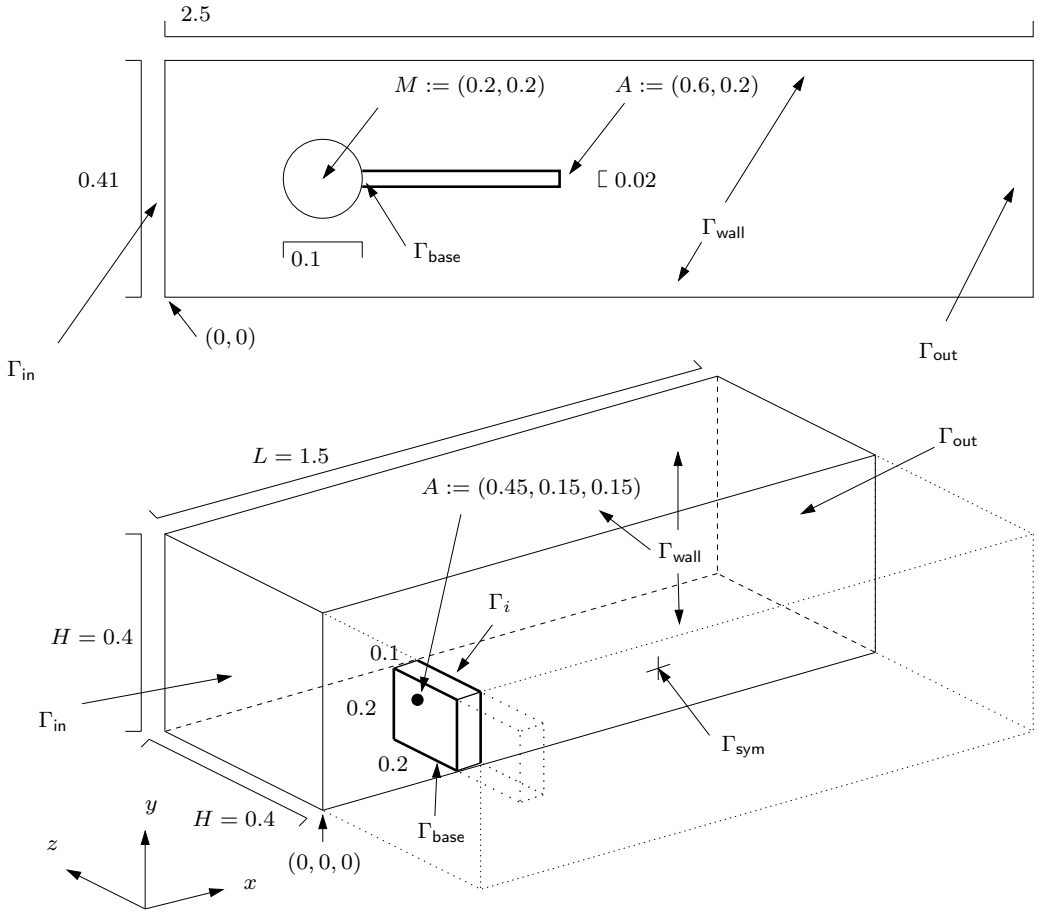


Figure 1: Configuration of the two test cases.

Remark 2 (Estimation of convergence rate and time, computational setup) *As the coupled fsi problem is highly nonlinear with time-dependent dynamics, we always estimate convergence rates and computational time as averages over a sequence of time-steps. Furthermore, as the number of Newton steps may vary from time-step to time-step, we fix the averaging by the following algorithm: in the case of the 2d-problem, we compute averages over* 100 *time-steps and we include 5 Newton steps per time-step into the averaging. Furthermore, we use an approximate Newton scheme by reusing the Jacobian: only every tenth time-step, a new Jacobian is assembled. In the case of the 3d-problem we choose the same procedure, but averaging is limited to* 20 *time-steps. To sum up: all results belonging to the 2d test case contain averages over* 500 *steps of the linear solver, including* 10 *assemblies of the system matrix. In the 3d test cases, we average over* 100 *runs of the linear solver including* 2 *assemblies of the Jacobian (and the necessary preparation of the smoother or preconditioner).*

This fixation allows for a fair comparison scaling over the mesh levels. All computations have been carried out on a Xeon E5-2690 cpu at 2.90 GHz with 256 GB of memory. Single core performance only is used for all computations.

The results in Table 1 present the performance of a standard geometric multigrid solver, used as preconditioner in an outer GMRES iteration. Coarse mesh problems are solved with help of a direct solver, and smoothing is done by a blockwise incomplete decomposition of the coupled system

matrix. This solver is the standard technique in the software library GASCOIGNE 3D[BBM+] and is highly efficient for problem in fluid-dynamics (compressible and incompressible), solid mechanics and various coupled multiphysics problems, see [KR]. Here however, we see that the convergence rates quickly deteriorate on fine meshes. Already starting with about 81 000 unknowns, this solvers ceases to work in the (more difficult) 2d-case. In terms of material parameters and deformation, the three dimensional test case is easier. This explains the better convergence rates. Besides the computational times, we see that memory consumption is a severe issue, in particular for the three dimensional benchmark configuration. Although multigrid convergence rates are worsening for large problem sizes, the robustness of this standard solver must be highly appreciated.

A straight-ahead alternative to coupled multigrid solvers is the solution via direct solvers. In Table 3 we give indications of the convergence rates, memory usage and computational time for the two benchmark problems using the direct solver UMFPACK [Dav14]. Memory consumption quickly goes beyond feasible limits. Computations on the finest meshes have not been possible. Furthermore, solution times increase in a similar fashion, such that direct solution - in particular for 3d problems - is no alternative. It is surprising, that the direct solver (using double precision arithmetic) shows very bad error reduction, giving only one or two digits in every step. We will see, that this behavior is due to the very bad conditioning of the coupled matrix.

Mesh Level (2d)	2	3	4	5	6	7
Avg. conv. rate	0.015	0.011	0.019	0.043	0.069	—
Memory usage	36 MB	135 MB	527 MB	2.9 GB	18.1 GB	$>$ 256 GB
Avg. Time	0.042 sec	0.21 sec	1.18 sec	8.75 sec	47.74 sec	—

Mesh Level (3d)	2	3	4	5
Avg. conv. rate	0.084	0.048	0.14	—
Memory usage	307 MB	7.3 GB	132 GB	$>$ 256 GB
Avg. Time	0.92 sec	36.25 sec	2472 sec	—

Table 3: Convergence rates, memory usage and average computation time for linear solution with a monolithic direct solver.

2.1 Literature overview

Here, we briefly give an overview over some of the existing literature on solving the linear systems of equations arising from the discretization of fluid-structure interactions. We limit this overview to the case of strongly monolithic schemes. For further reading and a wider overview, that also includes state of the art versions of segregated methods, we refer to the literature [MS02, MS03, BS06, FGG07, QQ07, HHB08, KW08, KGF+10, GKW10, BS10, van11, HWL12].

Monolithic approaches [Hei04, RDH+12, BTT13] are usually believed to be more robust than their partitioned counterparts. However, due to a lack of efficient solvers for the resulting algebraic systems, they have been rarely used for realistic and complex problems [HHB08]. As the underlying advantage of segregated approaches, some authors [FWR07] identify the inherent use of smaller matrices with better condition numbers. This idea will also guide the design of our monolithic multigrid solver: where matrices are to be inverted, we will partition by splitting into the sub-systems. Another obvious

drawback of monolithic schemes is the need for specially adapted solution methods and their efficient implementation [FGG07].

Efficient monolithic solvers for fsi problems are mostly based on block preconditioned Krylov subspace methods with an outer iteration of partitioned Gauss-Seidel type, see [Hei04, WHWT05, HHB08]. This keeps the monolithic residual of the coupled problem, but allows to use well-suited solver techniques for preconditioning in the two subfields, e.g. multigrid iterations. Roughly speaking, these solvers are of domain decomposition type, with the special character, that different systems are solved on different domains. Further splitting of the domains leads to a natural parallelism. Even without taking care of the fluid-solid interface, domain decomposition helps to reduce matrix sizes and enhances the conditioning [BC10]. A drawback of such approaches is the limited rate of convergence, that is restricted to the performance of the Krylov, the Gauss-Seidel coupling or the domain decomposition iteration, regardless of the inner solution procedure.

Brummelen, et al. [vvd08] analyzed a simplified problem with potential flow and a lower dimensional solid. Here, the authors showed, that a monolithically coupled multigrid iteration – with a decoupling within the multigrid smoother only – would serve as an optimal solver (convergence rates going to zero for increasing mesh-levels), if the two subproblems in the smoother are solved exactly. In the following we propose to follow this approach.

The basic idea of multigrid solvers is easily applied to monolithic fluid-structure interactions. For designing robust smoothers, similar difficulties as for preconditioners must be considered: due to the bad condition numbers, matrix sizes must be kept small and a partitioning into fluid and solid is advisable, see again [vvd08]. Hron and Turek et al. [HT06b, HT06a, RDH$^+$12] describe a geometric multigrid solver with different smoothers of Vanka type. Locally, fluid- and solid-problem are monolithically coupled in small blocks, that are inverted with the help of direct solvers. They report good convergence rates for two dimensional problems. Another recent approach – also based on standard block-smoothers of Vanka type – uses multigrid domain decomposition methods [Bna14].

Gee, Küttler and Wall [GKW10] presented a monolithic algebraic multigrid solver as preconditioner for Krylov subspace iterations, which is very similar to the solver presented here. Partitioning is applied in the smoother, all other parts are treated in a monolithic manner. For complex 2d and 3d problems, the authors report nearly optimal convergence rates for the coupled solution procedure. The fundamental difference is the choice of an algebraic multigrid method (AMG). AMG methods are more flexible, as coarse problems are constructed without a geometrical hierarchy of the problem. In particular for complex 3d problems, coarse meshes are often not available. Geometric multigrid solvers are very efficient in adaptive finite element schemes, that are based on a hierarchy of triangulations and finite element spaces [Ric12]. This construction gives rise to a natural mesh hierarchy usable for the geometric multigrid solver [BB00].

2.2 Outline

The aim of this work is to design a robust smoother for geometric multigrid iterations applied to coupled fluid-structure interactions. We will follow the idea of Brummelen and coworkers [vvd08], that a partitioned iteration can work as optimal smoother within the multigrid iteration. In the following section, we will shortly introduce the governing equations of fluid-structure interactions as well as a discretization in time (with the backward Euler scheme) and space (with finite elements). Further, we derive the algebraic equations to be solved in every step of a globally coupled Newton's iteration. Then, in Section 5, a thorough investigation of the conditioning of the algebraic systems is presented. It will be seen, that the devastating conditioning of the global system largely stems from the coupling of the two very different subproblems. The standalone matrices of fluid and solid problem carry condition numbers, that are expected and that can be taken care of by established

techniques. Section 6 introduces the geometric multigrid solver and in particular the partitioned smoother. Finally, in Section 7, the multigrid solver is applied to the two benchmark problems presented in this introduction. A short summary concludes.

3 Problem setting

As literature on fluid-structure interactions is rich, we will quickly focus on deriving the relevant system of equations. We present the coupled problem in the framework of the *Arbitrary Lagrangian Eulerian* coordinates (ALE) by transformation of the fluid problem onto a fixed reference domain [HLZ81, Don82].

Let $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ be a domain, that is split into a fluid-domain $\mathcal{F} \subset \mathbb{R}^d$ and solid-domain $\mathcal{S} \subset \mathbb{R}^d$, separated by a common interface $\mathcal{I} \subset \mathbb{R}^{d-1}$

$$\Omega = \mathcal{F} \cup \mathcal{I} \cup \mathcal{S}, \quad \mathcal{F} \cap \mathcal{S} = \emptyset, \quad \mathcal{I} = \partial\mathcal{F} \cap \partial\mathcal{S}.$$

We call this partitioning the reference configuration. Under evolving dynamics, solid and fluid domains will deform $\Omega(t) = \mathcal{F}(t) \cup \mathcal{I}(t) \cup \mathcal{S}(t)$. By $T_s(t) := \mathrm{id} + \mathbf{u}_s(t) : \mathcal{S} \to \mathcal{S}(t)$ we denote the Lagrangian-Eulerian map between the structural Lagrangian reference system and its moving Eulerian counterpart, where by $\mathbf{u}_s$ we denote the solid's deformation. By $T_f(t) : \mathcal{F} \to \mathcal{F}(t)$ we denote an ALE-map between the reference framework in $\mathcal{F}$ and the Eulerian fluid domain $\mathcal{F}(t)$, see [RW]. To describe this mapping, we introduce with $T_f(t) := \mathrm{id} + \mathbf{u}_f(t)$ a *artificial deformation* of the fluid domain. We assume, that $T(t) : \Omega \to \Omega(t)$ with $T|_{\mathcal{S}} = T_s$ and $T|_{\mathcal{F}} = T_f$ is continuous on $I \times \Omega$ and a piece-wise C^2 diffeomorphism on $I \times \mathcal{F}$ and $I \times \mathcal{S}$. Then, let $\mathbf{F} := \nabla T$ and $J := \det \mathbf{F}$. Globally on Ω, we denote by $\mathbf{u}$ the combined deformation of fluid and solid system, and by $\mathbf{v}$ the global velocity field.

The monolithic ALE-formulation of the coupled fluid-structure interaction problem is given by finding velocity $\mathbf{v}$, deformation $\mathbf{u}$ and pressure p in

$$\mathbf{v}(t), \mathbf{u}(t) \in \mathcal{V} := H_0^1(\Omega)^d, \quad p(t) \in \mathcal{L}_f := L^2(\mathcal{F}) \setminus \mathbb{R},$$

such that almost everywhere in time it holds

$$\begin{aligned}
&\Big(\rho_f J_f(\partial_t \mathbf{v} + \mathbf{F}_f^{-1}(\mathbf{v} - \partial_t \mathbf{u}) \cdot \nabla \mathbf{v}), \boldsymbol{\phi}\Big)_{\mathcal{F}} + \Big(J_f \sigma_f \mathbf{F}_f^{-T}, \nabla \boldsymbol{\phi}\Big)_{\mathcal{F}} \quad - \Big\langle \rho_f \nu_f J_f \vec{n} \cdot \nabla \mathbf{v}^T, \boldsymbol{\phi} \Big\rangle_{\Gamma_{\text{out}}} \\
&+ \Big(\rho_s^0 \partial_t \mathbf{v}, \boldsymbol{\phi}\Big)_{\mathcal{S}} + \Big(\mathbf{F}_s \boldsymbol{\Sigma}_s, \nabla \boldsymbol{\phi}\Big)_{\mathcal{S}} - \Big(J_f \rho_f \mathbf{f}_f, \boldsymbol{\phi}\Big)_{\mathcal{F}} - \Big(\rho_s^0 \mathbf{f}_s, \boldsymbol{\phi}\Big)_{\mathcal{S}} = 0 \qquad \forall \boldsymbol{\phi} \in \mathcal{V} \qquad (2a) \\
&\Big(\operatorname{div}(J_f \mathbf{F}_f^{-1} \mathbf{v}), \xi_f\Big)_{\mathcal{F}} + s(p, \xi_f)_{\mathcal{F}} = 0 \qquad \forall \xi_f \in \mathcal{L}_f \qquad (2b) \\
&a^{ALE}(\mathbf{u}, \boldsymbol{\psi}_f) = 0 \qquad \forall \boldsymbol{\psi}_f \in \mathcal{V}_f \qquad (2c) \\
&\Big(\partial_t \mathbf{u} - \mathbf{v}, \boldsymbol{\psi}_s\Big)_{\mathcal{S}} = 0 \qquad \forall \boldsymbol{\psi}_s \in \mathcal{V}_s, \qquad (2d)
\end{aligned}$$

where $\mathcal{V}_f := H_0^1(\mathcal{F})^d$ and $\mathcal{V}_s := L^2(\mathcal{S})^d$. By ρ_f and ρ_s^0 we denote fluid's and solid's density, by $\mathbf{f}_f$ and $\mathbf{f}_s$ acting volume forces. The integral over the outflow-boundary Γ_{out} is required to realize the *do-nothing* outflow condition (1).

By $s(\cdot, \cdot)$ we denote a pressure stabilization term, e.g. by Local Projections, see e.g. [BB01]. By $a^{ALE}(u, \boldsymbol{\psi}_f)$ in (2c) we denote a bilinear form, that describes a differential operator defining the ALE mapping as an extension of $\mathbf{u}_s$ to the fluid domain, see [Wic11] for different choices. Here, we assume, that $a^{ALE}(\cdot, \cdot)$ corresponds to a simple elliptic operator.

Remark 4 *We formulate the coupled system, equation (2), on the common fixed reference domain $\Omega = \mathcal{F} \cup \mathcal{I} \cup \mathcal{S}$. Mesh motion is hidden in the fluid deformation field $\mathbf{u}_f$. By transformation of the fluid problem, nonlinearities arise. In an equivalent formulation, the fluid problem could be written in the Eulerian coordinate system on a moving Eulerian domain $\mathcal{F}(t)$. This procedure is well-established in literature [BTT13] and has the benefit, that the Navier-Stokes system is given in standard formulation, with the exception of an additional convection term coming from the motion of the domain. As both formulation – transformed in ALE coordinates on a fixed mesh, vs. Eulerian on a deformed mesh – are equivalent in the discretized form, they result in the same system matrices with the same properties.*

4.1 Temporal discretization

A seminal work on time discretizations of fluid-structure interactions in ALE formulation is given by Le Tallec and Mouro [LM01]. Exact details on the application to the coupled formulation (2), where no mesh-update is applied is given in [RW15]. In the present context, time-discretization is of lesser importance and we consider the backward Euler method only. Let

$$0 = t_0 < t_1 < \cdots < t_M = T, \quad I_m := [t_{m-1}, t_m], \quad k_m := t_m - t_{m-1},$$

be a subdivision of the time-interval and let $\mathbf{v}^m, \mathbf{u}^m$ and p^m be the approximation at time t_m. The backward Euler scheme is derived by approximating $\mathbf{v}'(t_m) \approx k_m^{-1}(\mathbf{v}^m - \mathbf{v}^{m-1})$ and in (2a) and $\mathbf{u}'(t_m) \approx k_m^{-1}(\mathbf{u}^m - \mathbf{u}^{m-1})$ in (2a) and (2d). The different time-steps can be considered as a quasi-stationary problems. All terms related to the old solution are known and therefore shifted to the right hand side of the equation. What remains, is the time-discretization parameter k_m as scaling factor. As example, equation (2d) gets

$$(k_m^{-1}\mathbf{u}^m - \mathbf{v}, \boldsymbol{\psi}_s)_{\mathcal{S}} = (k_m^{-1}\mathbf{u}^{m-1}, \boldsymbol{\psi}_s)_{\mathcal{S}}.$$

For the following, we will skip all indices "m" referring to the time-discretization. To shorten notation, we write every time-step as

$$U := \{\mathbf{v}, p, \mathbf{u}\} \in \mathcal{V} \times \mathcal{L}_f \times \mathcal{V} : \; A(U)(\Phi) = F(\Phi) \; \forall \Phi := \{\boldsymbol{\phi}, \xi_f, \boldsymbol{\psi}_f, \boldsymbol{\psi}_s\} \in \mathcal{V} \times \mathcal{L}_f \times \mathcal{V}_f \times \mathcal{V}_s, \qquad (3)$$

with a semi linear form $A(\cdot)(\cdot)$ defined as sum over (2a-2d), and a linear form $F(\cdot)$ depending on the problem data and the old solution.

4.2 Linearization and spatial discretization

The system of equations introduced in the previous section is nonlinear due to convective terms, and material laws. A special nonlinearity however comes from the motion of the fluid-domain expressed by the ALE mapping $\mathrm{id} + \mathbf{u}_f$. Given an initial solution $U^{(0)}$, e.g. $U^{(0)} = \{\mathbf{v}^{\mathrm{old}}, \mathbf{u}^{\mathrm{old}}, p^{\mathrm{old}}\}$ taken from the last time-step, we iterate $U^{(k+1)} = U^{(k)} + W^{(k)}$:

$$W^{(k)} = \{\mathbf{v}^{(k)}, \mathbf{u}^{(k)}, p^{(k)}\} : \quad A'(U^{(k)})(W^{(k)}, \Phi) = F(\Phi) - A(U^{(k)})(\Phi), \qquad (4)$$

where by $A'(\cdot)(\cdot,\cdot)$, we denote the directional derivative of $A(\cdot)(\cdot)$ around the current approximation $U^{(k)}$ in direction of the update $W^{(k)}$, defined by

$$A'(U)(W, \Phi) := \frac{d}{ds} A(U + sW)(\Phi)\Big|_{s=0}.$$

While the Jacobian A' could be evaluated by finite differences [Tez01, GB95, HT06a] or with help of automatic differentiation [Dun07], we choose an analytic evaluation, as it gives better insight to the

structure of the linearized system. See [FM05] for a detailed derivation of the Jacobian and [Ric12] for the exact application to the coupled system in reference formulation (2). For the following, we introduce the notation $W := \{\mathbf{w}, \mathbf{z}, q\}$ for the update and write the Jacobian of (3) as

$$\begin{aligned} A'(U)(W,\Phi) = & a_{\mathbf{v}}^{NS}(\mathbf{v},\mathbf{u},p)(\mathbf{w},\boldsymbol{\phi}) + a_{\mathbf{u}}^{NS}(\mathbf{v},\mathbf{u},p)(\mathbf{z},\boldsymbol{\phi}) + a_{p}^{NS}(\mathbf{v},\mathbf{u},p)(q,\boldsymbol{\phi}) \\ & + a_{\mathbf{v}}^{ES}(\mathbf{v},\mathbf{u})(\mathbf{w},\boldsymbol{\phi}) + a_{\mathbf{u}}^{ES}(\mathbf{v},\mathbf{u})(\mathbf{z},\boldsymbol{\phi}) \\ & + a_{\mathbf{v}}^{div}(\mathbf{v},\mathbf{u},p)(\mathbf{w},\xi_f) + a_{\mathbf{u}}^{div}(\mathbf{v},\mathbf{u},p)(\mathbf{z},\xi_f) + a_{p}^{div}(\mathbf{v},\mathbf{u},p)(q_f,\xi_f) + s(q_f,\xi_f) \\ & + a_{\mathbf{u}}^{ALE}(\mathbf{u})(\mathbf{z},\boldsymbol{\psi}_f) + a_{\mathbf{v}}^{uv}(\mathbf{v},\mathbf{u})(\mathbf{w},\boldsymbol{\psi}_s) + a_{\mathbf{u}}^{uv}(\mathbf{v},\mathbf{u})(\mathbf{z},\boldsymbol{\psi}_s), \end{aligned} \tag{5}$$

where the upper index refers to the equation, i.e. "NS" for the fluid part in the momentum equation (2a), "ES" the elastic structure part of (2a), "div" the divergence equation (2b), "ALE" the ALE extension (2c) and "uv" the velocity deformation coupling in (2d). The lower index refers to the derivative. Once again, see [Ric12] for a detailed derivation of all these derivatives.

Spatial discretization is straightforward. We choose equal-order Lagrangian finite elements for all variables, velocity, deformation and pressure. By $V_h := \{\boldsymbol{\phi}_h^i,\ i = 1,\dots,N\}$ we denote the space of continuous finite elements of local degree $r \geq 1$ on a triangulation Ω_h of Ω. We assume, that the interface $\mathcal{I}$ is resolved in such a way, that by $\Omega_h := \Omega_{h,f} \cup \Omega_{h,s}$ natural triangulations of solid and fluid domains exist. By $V_{h,f}, V_{h,s}$ and $V_{h,i}$ we denote the restrictions of the finite element space V_h to the subdomains (fluid, solid and interface). Then, let $N := N_f + N_i + N_s$ with $N_f := \dim(V_{h,f})$, $N_s := \dim(V_{h,s})$ and $N_i := \dim(V_{h,i})$. Finally, we consider the discrete spaces:

$$\begin{aligned} \mathbf{v}_h, \mathbf{u}_h, \boldsymbol{\phi}_h \in V_h^d \subset \mathcal{V}, \qquad & p_h, \xi_{f,h} \in V_{h,f} + V_{i,f} \subset \mathcal{L}_f, \\ \boldsymbol{\psi}_{f,h} \in V_{h,f}^d + V_{h,i}^d \subset \mathcal{V}_f, \qquad & \boldsymbol{\psi}_{s,h} \in V_{h,s}^d + V_{h,i}^d \subset \mathcal{V}_s. \end{aligned}$$

To cope with the lacking inf-sup stability, we add stabilization terms to the divergence equation in form of the Local Projection Stabilization technique (LPS) in $s(\cdot,\cdot)$, see [BB01]. Details on all aspects of this discretization for fluid-structure interactions in ALE coordinates are given in the literature [RW, Ric12].

4.3 Matrix formulation

In a Galerkin formulation of the linearized problem (4), every linear form in (5) relates to a matrix acting on the coefficient vectors of the finite element representation. To fully comprehend the structure of the system, a twofold block-structure must be considered: First, parts of the equation act on solid domain, parts on fluid domain. Second, each part by itself can be formulated as a block-system, i.e. the usual structure of incompressible flows with velocity-velocity, velocity-pressure and in the case of stabilized finite elements with pressure-pressure couplings. Here, we denote all matrices belonging to the fluid problem, i.e. they stem from integration over $\mathcal{F}$ by F and all solid parts by S. The upper index refers to the part in the equation, the lower index refers to the derivative. To show two examples, by $F_{\mathbf{v}}^{NS}$ we denote the main part of the Navier-Stokes equations

$$(F_{\mathbf{v}}^{NS})_{ij} = a_{\mathbf{v}}^{NS}(\mathbf{v},\mathbf{u},p)(\boldsymbol{\phi}_h^{(j)}, \boldsymbol{\phi}_h^{(i)}), \quad \boldsymbol{\phi}_h^{(i)}, \boldsymbol{\phi}_h^{(j)} \in V_{h,f}^d + V_{h,i}^d,$$

and by $F_{\mathbf{u}}^{div}$ the matrix relating to the divergence equation derived with respect to the ALE deformation $a_{\mathbf{u}}^{div}(\cdot,\cdot)$:

$$(F_{\mathbf{u}}^{div})_{ij} = a_{\mathbf{u}}^{div}(\mathbf{v},\mathbf{u},p)(\boldsymbol{\phi}_h^{(j)}, \boldsymbol{\phi}_h^{(i)}), \quad \boldsymbol{\phi}_h^{(i)} \in V_{h,f} + V_{h,i}, \quad \boldsymbol{\phi}_h^{(j)} \in V_{h,f}^d + V_{h,i}^d.$$

Hence

$$F_{\mathbf{v}}^{NS} \in \mathbb{R}^{d(N_f+N_i)\times d(N_f+N_i)}, \quad F_{\mathbf{u}}^{div} \in \mathbb{R}^{(N_f+N_i)\times d(N_f+N_i)}.$$

The complete system matrix is given in form of two matrix-blocks:

$$\mathbf{F} = \begin{pmatrix} F_p^{div} & F_{\mathbf{v}}^{div} & F_{\mathbf{u}}^{div} \\ F_p^{NS} & F_{\mathbf{v}}^{NS} & F_{\mathbf{u}}^{NS} \\ 0 & 0 & F_{\mathbf{u}}^{ALE} \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} S_{\mathbf{v}}^{ES} & S_{\mathbf{u}}^{ES} \\ S_{\mathbf{v}}^{uv} & S_{\mathbf{u}}^{uv} \end{pmatrix}. \tag{6}$$

By introducing restriction operators from the global set of unknowns to the fluid and solid part

$$\mathcal{R}_f : \mathbb{R}^{2dN+(N_f+N_i)} \to \mathbb{R}^{(2d+1)(N_f+N_i)}, \quad \mathcal{R}_s : \mathbb{R}^{2dN+(N_f+N_i)} \to \mathbb{R}^{2d(N_i+N_s)},$$

the global system matrix is given as

$$\mathbf{A} := \mathcal{R}_f^T \mathbf{F} \mathcal{R}_f + \mathcal{R}_s^T \mathbf{S} \mathcal{R}_s. \tag{7}$$

The matrices $\mathbf{F}$ and $\mathbf{S}$ overlap on interface degrees of freedom. If we distinguish between fluid-unknowns $\mathbf{v}_f, \mathbf{u}_f, \mathbf{p}_f$, interface unknowns $\mathbf{v}_i, \mathbf{u}_i, \mathbf{p}_i$ and solid-unknowns $\mathbf{v}_s, \mathbf{u}_s$, the coupled system matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = \left(\begin{array}{ccc|ccc|cc} F_p^{div} & F_{\mathbf{v}}^{div} & F_{\mathbf{u}}^{div} & F_p^{div} & F_{\mathbf{v}}^{div} & F_{\mathbf{u}}^{div} & 0 & 0 \\ F_p^{NS} & F_{\mathbf{v}}^{NS} & F_{\mathbf{u}}^{NS} & F_p^{NS} & F_{\mathbf{v}}^{NS} & F_{\mathbf{u}}^{NS} & 0 & 0 \\ 0 & 0 & F_{\mathbf{u}}^{ALE} & 0 & 0 & F_{\mathbf{u}}^{ALE} & 0 & 0 \\ \hline F_p^{div} & F_{\mathbf{v}}^{div} & F_{\mathbf{u}}^{div} & F_p^{div} & F_{\mathbf{v}}^{div} & F_{\mathbf{u}}^{div} & 0 & 0 \\ F_p^{NS} & F_{\mathbf{v}}^{NS} & F_{\mathbf{u}}^{NS} & F_p^{NS} & F_{\mathbf{v}}^{NS} + S_{\mathbf{v}}^{ES} & F_{\mathbf{u}}^{NS} + S_{\mathbf{u}}^{ES} & S_{\mathbf{v}}^{ES} & S_{\mathbf{u}}^{ES} \\ 0 & 0 & 0 & 0 & S_{\mathbf{v}}^{uv} & S_{\mathbf{u}}^{uv} & S_{\mathbf{v}}^{uv} & S_{\mathbf{u}}^{uv} \\ \hline 0 & 0 & 0 & 0 & S_{\mathbf{v}}^{ES} & S_{\mathbf{u}}^{ES} & S_{\mathbf{v}}^{ES} & S_{\mathbf{u}}^{ES} \\ 0 & 0 & 0 & 0 & S_{\mathbf{v}}^{uv} & S_{\mathbf{u}}^{uv} & S_{\mathbf{v}}^{uv} & S_{\mathbf{u}}^{uv} \end{array}\right) \left[\begin{array}{c} \mathbf{p}_f \\ \mathbf{v}_f \\ \mathbf{u}_f \\ \hline \mathbf{p}_i \\ \mathbf{v}_i \\ \mathbf{u}_i \\ \hline \mathbf{v}_s \\ \mathbf{u}_s \end{array}\right]. \tag{8}$$

Coupling takes place at the interface only. Further, we note, that the ALE extension equation does not act on interface degrees of freedom as $\boldsymbol{\psi}_f = 0$ on $\mathcal{I}$. Hence, $\mathbf{F}$ itself, see (6) cannot be regular.

5 Condition number analysis of the system matrices

In this section, we will analyze the condition numbers of the Jacobian $\mathbf{A}$ (see (6) and (7)) and its different sub-parts. For this analysis, we consider the two benchmark problems introduced in Section 1.1. The first lines of Table 4 show the condition number of the coupled matrix

$$\text{cond}(\mathbf{A}) = \|\mathbf{A}\|_1 \|\mathbf{A}^{-1}\|_1,$$

in the 1-norm (maximum column sum). Furthermore, we indicate the condition numbers for the solid matrix $\mathbf{S}$ in (6), the main part of the Navier-Stokes system

$$\mathbf{F}^{NS} = \begin{pmatrix} F_p^{div} & F_{\mathbf{v}}^{div} \\ F_p^{NS} & F_{\mathbf{v}}^{NS} \end{pmatrix},$$

and the ALE extension matrix $F_{\mathbf{u}}^{ALE}$. The latter two matrices implement homogenous Dirichlet values on the fluid-structure interface $\mathcal{I}$. A separate discussion of $\mathbf{F}^{NS}$ and $F_{\mathbf{u}}^{ALE}$ is reasonable, as the system naturally decouples. All condition numbers are approximated with Matlab [TM]. To avoid scaling effects (from the size of mesh elements or from problem parameters), we apply diagonal scaling before computing the condition numbers. For both benchmark problems we show the resulting condition numbers on a sequence of uniform meshes.

Mesh Level (2d)	1	2	3	4	5
cond($\mathbf{A}$)	$\mathbf{3.30 \cdot 10^{12}}$	$\mathbf{9.85 \cdot 10^{12}}$	$\mathbf{4.36 \cdot 10^{13}}$	$\mathbf{1.76 \cdot 10^{14}}$	$\mathbf{7.40 \cdot 10^{14}}$
cond($\mathbf{S}$)	$4.19 \cdot 10^{7}$	$1.49 \cdot 10^{8}$	$5.54 \cdot 10^{8}$	$1.45 \cdot 10^{9}$	$5.26 \cdot 10^{9}$
cond($\mathbf{F}^{NS}$)	$3.12 \cdot 10^{8}$	$5.78 \cdot 10^{8}$	$1.15 \cdot 10^{9}$	$2.27 \cdot 10^{9}$	$4.34 \cdot 10^{9}$
cond($F_{\mathbf{u}}^{ALE}$)	$1.85 \cdot 10^{3}$	$7.67 \cdot 10^{3}$	$2.95 \cdot 10^{4}$	$1.16 \cdot 10^{5}$	$4.59 \cdot 10^{5}$

Mesh Level (3d)	1	2	3
cond($\mathbf{A}$)	$\mathbf{5.25 \cdot 10^{12}}$	$\mathbf{3.52 \cdot 10^{13}}$	$\mathbf{1.80 \cdot 10^{14}}$
cond($\mathbf{S}$)	$3.48 \cdot 10^{5}$	$2.22 \cdot 10^{6}$	$8.36 \cdot 10^{6}$
cond($\mathbf{F}^{NS}$)	$1.43 \cdot 10^{7}$	$3.25 \cdot 10^{7}$	$5.39 \cdot 10^{7}$
cond($F_{\mathbf{u}}^{ALE}$)	$7.74 \cdot 10^{1}$	$2.86 \cdot 10^{2}$	$1.62 \cdot 10^{3}$

Table 4: Condition numbers of the 2d (top) and 3d (bottom) benchmark problems for the full system matrix $\mathbf{A}$, the solid part $\mathbf{S}$, the Navier-Stokes part $\mathbf{F}^{NS}$ and the matrix of the ALE extension $F_{\mathbf{u}}^{ALE}$ (considering harmonic extension).

A first glance at the numbers in Table 4 reveals the expected results with a proper scaling in terms of the mesh-size. This analysis however puts forward the dramatic effect of the monolithic coupling on the conditioning of the coupled system matrix $\mathbf{A}$, that finally causes standard coupled multigrid solvers (with coupled multigrid smoothers) to cease work, see the introduction and Table 1. By a decoupling, all condition numbers are within reasonable limits. This observation will guide the design of the partitioned multigrid smoother in the following section.

6 Solution of the linear systems

This is the main section, that describes the multigrid solver for the coupled system

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

where $\mathbf{x} = (\mathbf{v}, \mathbf{u}, \mathbf{p})$ with $\mathbf{v}, \mathbf{u} \in \mathbb{R}^{2dN}$ and $\mathbf{p} \in \mathbb{R}^{N_f+N_i}$ is the vector of solution coefficients. By $\mathbf{v}_f, \mathbf{u}_f \in \mathbb{R}^{2d(N_f+N_i)}$ and $\mathbf{v}_s, \mathbf{u}_s \in \mathbb{R}^{2d(N_i+N_s)}$ we denote the overlapping (on the interface) restrictions of these vectors to the fluid- and solid degrees of freedom. $\mathbf{A}$ is the coupled system matrix. The philosophy of the linear solver is to treat the coupled system in a monolithic manner as long as possible. The analysis in Section 5 shows, that the condition number $\text{cond}_1(\mathbf{A})$ is very large. This large condition number not only stems from the second order character of the partial differential equations, but also from the different numerical scales acting in fluid- and solid-problem. Just to highlight one example: the viscosities of water is about $10^{-3} Pa \cdot s$, the Young's modulus of steel is $2 \cdot 10^{11} Pa$. At the interface degrees of freedom, both equations are coupled in $\mathbf{A}$, see (6) and (7). Diagonal preconditioning does not significantly help to improve the condition number, see Table 4. Hence, whenever it is necessary to compute the inverse of $\mathbf{A}$, we will apply a splitting into fluid- and solid-part. The general outline of the solver is as follows:

1. As outer iteration to solve $\mathbf{A}\mathbf{x} = \mathbf{b}$ we employ a monolithic GMRES iteration.

2. The GMRES solver is preconditioned by a geometric monolithic multigrid solver.

3. The multigrid smoother is constructed as a domain decomposition iteration with a Dirichlet-Neumann coupling on the interface into solid-problem governed by **S** and fluid-problem governed by **F** (formulated as Dirichlet problem).

4. Each of these subproblems is smoothed with some steps of a simple iteration, e.g. Richardson or BiCGStab preconditioned with solvers of Vanka type or of block-ILU type.

5. The coarse-mesh problem will be treated by a direct solution of the monolithic coupled system.

The reason for applying the partitioning in the multigrid smoother and not as outer preconditioner is motivated by two arguments: first, it has been shown by Brummelen and coworkers [vvd08], that a partitioned smoother with exact solution of the two subproblems is a perfect smoother for a certain class of fluid-structure interactions. Perfect here implies, that the convergence rate will go to zero for increasing number of mesh-levels. Second, it is the simple observation, that the role of the multigrid smoother is not that of finding a global solution, but it's only intend is to locally smooth high frequent error contributions. Here, global coupling conditions must not be resolved.

6.1 Outer iteration

As outer solver we apply the usual GMRES solver, see [Saa96]. Every iteration of the GMRES requires the solution of preconditioning problems $\mathbf{x} = \mathcal{P}(\mathbf{b})$. As preconditioner $\mathcal{P} \approx \mathbf{A}^{-1}$, we consider one iteration of a geometric multigrid solver.

For this multigrid iteration, a hierarchical layout of the finite element spaces must be present. Hence, let

$$\Omega_h = \Omega_L, \ldots, \Omega_l, \ldots, \Omega_0 = \Omega_H, \quad V_h = V_L \supset \cdots \supset V_l \supset \cdots \supset V_0 = V_H,$$

be a hierarchy of triangulations and corresponding nested hierarchy of finite element spaces. These meshes can be locally refined in such a way, that an element $K \in \Omega_l$ is either also an element $K \in \Omega_{l+1}$ of the next fine mesh, the element K is split into a set of elements $K = K_1 \cup \cdots \cup K_r$ with $K_i \in \Omega_{l+1}$. By splitting, we usually refer to cutting a quadrilateral into four and a hexahedra into eight smaller hexahedra. However, anisotropic splitting is also possible, see [Ric10]. By V_l we denote the corresponding finite element space on Ω_l. We apply global smoothing, see [BB00], such that every space covers the complete domain Ω and such that it holds $V_l \subset V_{l+1}$.

All components of this multigrid-solver apart from the smoother are standard and found in the literature. The prolongation operator is simply the embedding in the nested finite element spaces $V_{l-1} \subset V_l$. The restriction is the L^2-projection $V_l \to V_{l-1}$. By exploiting the hierarchical structure of the meshes, this projection can be easily performed by node-wise local operations on the residual vector $\mathbf{r}_l \mapsto \mathbf{r}_{l-1}$ without solving further global problems.

6.2 Smoothing

Every smoothing step of the multigrid algorithm requires the approximation of the system

$$\mathbf{A}_l \mathbf{x}_l = \mathbf{b}_l.$$

For the following, we can skip the level index, as all levels cover the complete domain and can be treated in the same way. The smoothing operator

$$\mathbf{x}^{(i)} = \mathscr{S}(\mathbf{x}^{(i-1)}, \mathbf{b}), \quad i = 1, \ldots, \nu,$$

is realized as a preconditioned iteration with Gauss-Seidel coupling:

1. Calculate residual of the solid problem

$$\mathbf{r}_s^{(i)} = \mathcal{R}_s(\mathbf{b} - \mathbf{A}\mathbf{x}^{(i-1)})$$

2. Solve solid problem

$$\mathbf{S}\mathbf{w}_s^{(i)} = \mathbf{r}_s^{(i)}$$

3. Update

$$\mathbf{x}^{(i-\frac{1}{2})} = \mathbf{x}^{(i-1)} + \mathcal{R}_s^T \mathbf{w}_s^{(i)}$$

4. Calculate residual of the fluid problem

$$\mathbf{r}_f^{(i)} = \mathcal{R}_f(\mathbf{b} - \mathbf{A}\mathbf{x}^{(i-\frac{1}{2})})$$

5. Solve fluid problem

$$\bar{\mathbf{F}}\mathbf{w}_f^{(i)} = \bar{\mathbf{r}}_f^{(i)}$$

6. Update

$$\mathbf{x}^{(i)} = \mathbf{x}^{(i-\frac{1}{2})} + \mathcal{R}_f^T \mathbf{w}_f^{(i)}$$

The two subproblems for solid and fluid are treated by a Dirichlet-Neumann coupling with homogenous Dirichlet values realized for velocity and deformation on the interface $\mathcal{I}$ in the fluid matrix $\bar{\mathbf{F}}$ and the right hand side $\bar{\mathbf{r}}_f^{(i)}$ (indicated by the bar). This corresponds to assigning the kinematic coupling condition to the fluid problem and the dynamic condition to the solid problem, see Section 3. An alternative would be to treat the interface in a balanced way with a Robin condition for both subproblems. This however could deteriorate the condition numbers of the sub matrices by canceling the strict partitioning into fluid and solid. Matrix entries belonging to the interface variables would be mixed, see (8).

We start by describing the single fields. In a first step, we assume, that the local subproblems are solved exactly with help of a direct solver. By this intermediate construction, we will validate the smoothing property of the partitioning.

6.2.1 The solid problem.

The solid part in the smoothing operation asks for an approximation to the system

$$\mathbf{S}\mathbf{w} = \mathbf{r} \quad \Leftrightarrow \quad \begin{pmatrix} S_{\mathbf{v}}^{ES} & S_{\mathbf{u}}^{ES} \\ S_{\mathbf{v}}^{uv} & S_{\mathbf{u}}^{uv} \end{pmatrix} \begin{pmatrix} \mathbf{v}_{s+i} \\ \mathbf{u}_{s+i} \end{pmatrix} = \mathbf{r}_s := \mathcal{R}_s \left(\mathbf{b} - \mathbf{A}\mathbf{x}^{\text{old}} \right), \tag{9}$$

where $\mathbf{x}^{\text{old}}$ is the old approximation. We start by analyzing the effect of this solid problem on the interface condition. Acting on the interface unknowns $\mathbf{v}_i$ and $\mathbf{u}_i$ only, the solid problem relates to (omitting the right hand side $\mathbf{b}$):

$$S_{\mathbf{v}}^{ES}(\underbrace{\mathbf{v}_i + \mathbf{v}_i^{\text{old}}}_{\mathbf{v}_i^{\text{new}}}) + S_{\mathbf{u}}^{ES}(\underbrace{\mathbf{u}_i + \mathbf{u}_i^{\text{old}}}_{\mathbf{u}_i^{\text{new}}}) = F_p^{NS}\mathbf{p}_i^{\text{old}} - F_{\mathbf{v}}^{NS}\mathbf{v}_i^{\text{old}} - F_{\mathbf{u}}^{NS}\mathbf{u}_i^{\text{old}}.$$

The dynamic condition constitutes itself as boundary terms in $S_{\mathbf{u}}^{ES}$, $F_{\mathbf{v}}^{NS}$ and F_p^{NS}. Hence, this iteration corresponds to the dynamic coupling condition

$$\sigma_s(\mathbf{u}^{\text{new}})\vec{n}_s + \sigma_f(\mathbf{v}^{\text{old}}, p^{\text{old}})\vec{n}_f = 0.$$

Alternative approaches are possible. By adding fluid-interface parts to the solid-matrix $S_{\mathbf{v}}^{ES}$, the dynamic condition would include an intermediate fluid-velocity. Gee, Küttler and Wall [GKW10] shift the complete interface treatment to the fluid-subproblem. Here, we strictly decouple both problems in a Dirichlet-Neumann sense in order to separate different parameter scales.

Problem (9) can be decoupled, as both $S_{\mathbf{v}}^{ES}$ and $S_{\mathbf{v}}^{uv}$ correspond to the mass matrix, see (2) and (5). It holds

$$a_v^{ES}(\mathbf{w}, \boldsymbol{\phi}) = k^{-1}(\rho_s^0 \mathbf{w}, \boldsymbol{\phi}), \quad a_v^{uv}(\mathbf{w}, \boldsymbol{\psi}_s) = -(\mathbf{w}, \boldsymbol{\psi}_s) \quad \Rightarrow \quad S_{\mathbf{v}}^{ES} = -\rho_s^0 k^{-1} S_{\mathbf{v}}^{uv}.$$

Hence, instead of solving (9) as one coupled system, we can approximate the solution in two sub-steps:

$$\left[\rho_s^0 k^{-1} S_{\mathbf{u}}^{uv} + S_{\mathbf{u}}^{ES}\right] \mathbf{u}_{s+i} = \mathbf{r}_{s,\mathbf{v}} + \rho_s^0 k^{-1} \mathbf{r}_{s,\mathbf{u}}, \quad S_{\mathbf{v}}^{ES} \mathbf{v}_{s+i} = \mathbf{r}_{s,\mathbf{v}} - S_{\mathbf{u}}^{ES} \mathbf{u}_{s+i}. \tag{10}$$

In Section 6.3 we describe, how these problems can be approximated by an iterative scheme.

6.2.2 The fluid problem.

The fluid step

$$\bar{\mathbf{F}}\mathbf{w} = \mathbf{r} \quad \Leftrightarrow \quad \begin{pmatrix} F_p^{div} & \bar{F}_{\mathbf{v}}^{div} & \bar{F}_{\mathbf{u}}^{div} \\ \bar{F}_p^{NS} & \bar{F}_{\mathbf{v}}^{NS} & \bar{F}_{\mathbf{u}}^{NS} \\ 0 & 0 & \bar{F}_{\mathbf{u}}^{ALE} \end{pmatrix} \begin{pmatrix} \mathbf{p}_{f+i} \\ \mathbf{v}_{f+i} \\ \mathbf{u}_{f+i} \end{pmatrix} = \bar{\mathbf{r}}_f := \bar{\mathcal{R}}_f(\mathbf{b} - \mathbf{A}\mathbf{x}^{\text{old}}),$$

is modified to carry homogenous Dirichlet values for velocity and deformation on all interface nodes. This problem decouples into the ALE extension part

$$\bar{F}_{\mathbf{u}}^{ALE} \mathbf{u}_{f+i} = \bar{\mathbf{r}}_{f,\mathbf{u}}, \tag{11}$$

followed by the Navier-Stokes part

$$\begin{pmatrix} F_p^{div} & \bar{F}_{\mathbf{v}}^{div} \\ \bar{F}_p^{NS} & \bar{F}_{\mathbf{v}}^{NS} \end{pmatrix} \begin{pmatrix} \mathbf{p}_{f+i} \\ \mathbf{v}_{f+i} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{f,p} - \bar{F}_{\mathbf{u}}^{div} \mathbf{u}_{f+i} \\ \mathbf{r}_{f,\mathbf{v}} - \bar{F}_{\mathbf{u}}^{NS} \mathbf{u}_{f+i} \end{pmatrix}. \tag{12}$$

Again, we first assume, that these two problems (11) and (12) are inverted with help of a direct solver. An approximative approach is described in Section 6.3.

6.2.3 Numerical analysis of the partitioned smoother with exact subproblems.

Before presenting the final multigrid solver that avoids all direct matrix inversions, we show in Table 5 convergence rates, memory usage and computational times for the multigrid iteration with a partitioned smoother. On every mesh level, we use one single smoothing iteration. The different subproblems (10), (11) and (12), are solved with help of the direct solver UMFPACK [Dav14]. Comparing to the results given in Table 3, that correspond to a monolithic direct solver for the coupled problem, we first observe, that only about half the memory is used. Further on, the average convergence rates of the linear solver are even better than those obtained with a monolithic direct solver. This is only due to the very bad conditioning, that causes significant loss of digits in applying direct inversion. Actually, we observe better convergence rates for finer meshes. This result is in accordance to the theoretical observations of Brummelen and coworkers [vvd08].

6.3 Iterative smoothing procedure

The different sub-steps described in the previous section ask for the approximation (smoothing) of subproblems for the structure (10), the ALE extension (11) and the Navier-Stokes equations in ALE formulation (12). For all of these problems of type $\mathbf{Ax} \approx \mathbf{b}$, we choose a simple preconditioned Richardson iteration

$$\mathbf{x}^k = \mathbf{x}^{k-1} + \mathbf{P}(\mathbf{A}^{-1})(\mathbf{b} - \mathbf{A}\mathbf{x}^{k-1}), \quad k = 1, 2, \ldots, K,$$

with $\mathbf{x}^0 = 0$ and usually with $K = 4$. As preconditioner we choose a stabilized incomplete lower-upper decomposition of the Matrix $\mathbf{A}$. This decomposition is performed in a block-wise sense. All degrees of freedom coupling in one node are strongly coupled. For the Navier-Stokes part (in three dimensions), this corresponds to small 4×4 blocks coupling pressure and the three velocities. For stabilization, we strengthen the diagonal by adding the weighted sum of all off-diagonals. This approximation is well suited as smoothing operation for various complex problems, see [BB00, BBRW99, BR07, KR] for description of the smoother and application to different problems. It is possible to use stronger iterations for enhancing the smoothing process.

The idea of this smoother can be seen as a mixture of Vanka-smoother and ILU-smoother. It is stronger, than a Vanka-smoother with Jacobi- or Gauss-Seidel coupling, as the inversion of the local blocks is embedded into an incomplete decomposition of the matrix. On the other hand, we use smaller block-sizes coupling only the degrees of freedom in single nodes (and not even those of one element). It will be worthwhile to analyze different smoothers for the sub-problems, as standard Vanka-smoothers will be better suited for parallelization.

7 Numerical testing

7.1 Multigrid-Iteration with exact smoother

In a first step of the numerical analysis, we apply the multigrid iteration of the previous section with exact solution of the subproblems within the multigrid-smoother. Solid problem of Section 6.2.1 and the two fluid problems for extension and transformed Navier-Stokes equations in Section 6.2.2 are solved by a direct solver. We apply no pre-smoothing and only one post-smoothing step on every mesh-level. The coupled problem on the coarsest level is solved with a monolithic direct solver.

Mesh Level (2d)	2	3	4	5	6	7
Avg. conv. rate	0.049	0.034	0.018	0.016	0.019	0.014
Memory usage	21 MB	71 MB	292 MB	1.2 GB	5.2 GB	49 GB
Avg. Time	0.07 sec	0.27 sec	1.18 sec	5.90 sec	35.93 sec	345 sec

Mesh Level (3d)	2	3	4	5
Avg. conv. rate	< 0.01	< 0.01	< 0.01	–
Memory usage	194 MB	2.1 GB	44 GB	> 256 GB
Avg. Time	2.51 sec	37.03 sec	1217 sec	–

Table 5: Convergence rates, memory usage and average computation time for linear solution with splitting smoother and exact subproblems.

In Table 5 we collect the results for the two benchmark problems. It is well seen, that the convergence rates are stable under mesh-refinement. Furthermore, convergence is very fast, in particular compared to the rates of the monolithic multigrid smoother given in Table 1. However we note, that the separate problems within the smoother are solved with a direct solver. Nevertheless, compared to a direct solution of the monolithic system, we could already substantially reduce the effort, as (in 3d) separate and smaller systems with three unknowns (extension) and four unknowns (Navier-Stokes and elasticity) are solved instead of one global system with seven unknowns.

7.2 Multigrid-Iteration with approximate smoothing in the sub-systems

Finally, we apply the final multigrid solver to both benchmark problems:

Mesh Level (2d)	2	3	4	5	6	7
Avg. conv. rate	0.078	0.042	0.043	0.040	0.048	0.054
Memory usage	17 MB	42 MB	142 MB	540 GB	2.1 GB	8.8 GB
Avg. Time	0.10 sec	0.43 sec	1.89 sec	8.47 sec	38.22 sec	171.92 sec

Mesh Level (3d)	2	3	4	5
Avg. conv. rate	0.049	0.052	0.059	0.068
Memory usage	115 MB	510 MB	4.5 GB	27 GB
Avg. Time	0.48 sec	4.10 sec	42.90 sec	401.2 sec

Table 6: Convergence rates, memory usage and average computation time for linear solution with splitting smoother and iterative smoother in subproblems.

In Table 6, we report on the performance of the multigrid solver with split smoothing and approximate solution of the subproblems as described in Section 6.3. The first glance shows three desired effects: the convergence rates are nearly robust with respect to the mesh size, memory usage is optimal (linear), and the computational time is nearly linear. Comparing the results of Table 6 with those for the standard multigrid solver in Table 1 or those using the direct solver in Table 3, we see a substantial improvement in both memory consumption and computational costs.

In Figure 2 we show a comparison of the memory performance of the different approaches. Here, we observe a great benefit of the splitting approach within the smoother and the avoidance of direct solvers, that always bring along fill-ins. Regarding the 3d problems, we observe a substantial improvement of the final multigrid solver with regard to the standard multigrid solver of Gascoigne 3D [BBM$^+$, KR]. This stems from the reduction of the overall matrix size: instead of one global 7×7 matrix, we only deal with smaller submatrices on either the fluid- or the solid-domain. The memory savings compared to direct solvers are dramatic, both for 2d and 3d problems.

Figure 3 shows a similar comparison regarding the average computational time required for solving the linear systems. Here, the similar performance of all methods in case of the 2d problem is a striking result. In particular the excellent performance of the direct solver UMFPACK [Dav14] must be appreciated. This result is even more surprising, as Figure 2 does show a significant and non-optimal increase of memory usage. Regarding the 3d test case, there is a substantial discrepancy between the different solver's performance. Direct inversion of the global matrix or use of direct solvers within the smoother process immediately ruins the performance. The standard multigrid solver of Gascoigne 3D [BBM$^+$, KR] shows a good performance, that is however sub-optimal, as the convergence rates

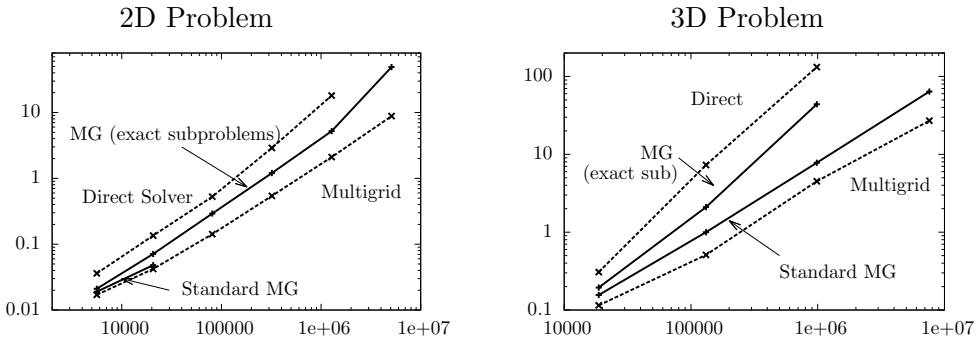


Figure 2: Memory usage in Gigabyte for the different solvers. 2D problem in left column and 3D problem in right one. The final multigrid solver shows linear memory consumption of about 1.8 Kilobyte per unknown for the 2d-problem and 4 Kilobyte per unknown in the 3d-case.

deteriorates on fine meshes, see Table 1. Only the final multigrid solver based on partitioned smoothing operations shows a nearly optimal (linear) scaling.

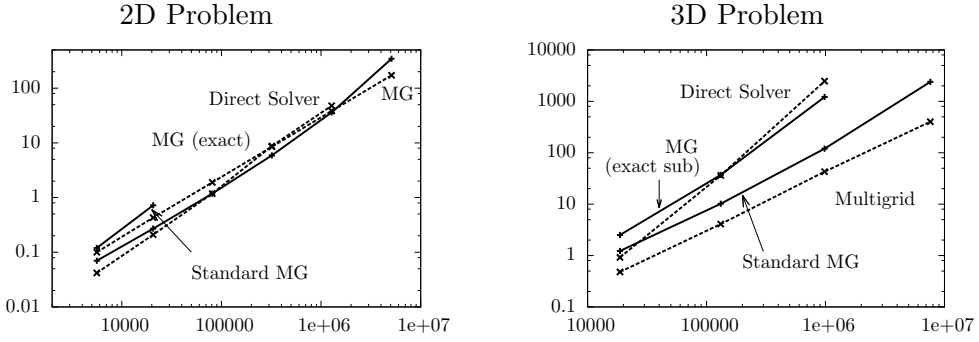
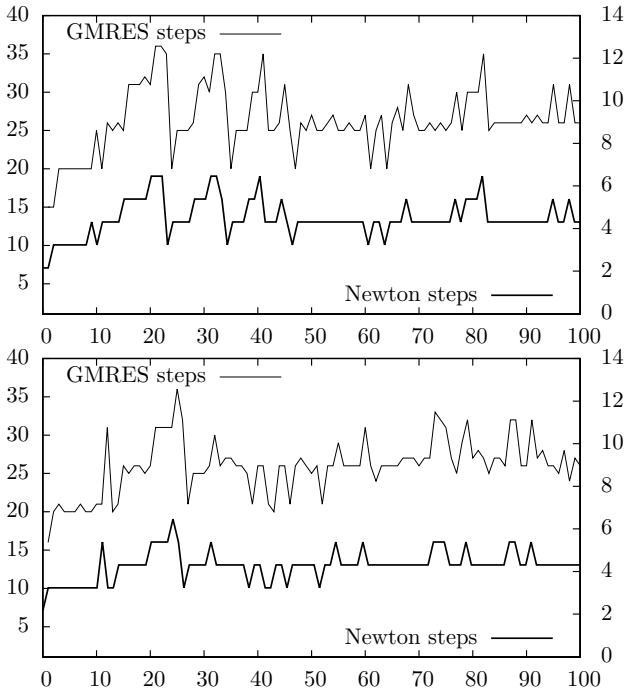


Figure 3: Computational time in seconds (lower row) for the different solvers. 2D problem in left column and 3D problem in right one. Nearly linear run-time for the final multigrid solver in 3D. In 2d, similar performance of all solvers.

Remark 8 (Comparison to numerical studies in the literature) *The two dimensional example discussed by Gee, Küttler and Wall in Section 8.2 of [GKW10] is comparable to the 2d-benchmark problem. Both configurations describe the self-excited motion of a flexible beam in a laminar flow. The authors [GKW10] used a discretization with about* 80 000 *degrees of freedom, similar to mesh-level 4, see Table 1. On this mesh, the geometric multigrid solver with partitioned smoother requires an average of* 1.89 *seconds in every time-step. An average of 5 Newton steps in every time-steps results in* 9.5 *seconds per time-step compared to an average of* 7 *seconds taken from [GKW10]. A direct comparison is difficult, as the authors of [GKW10] used a Newton tolerance of* 10^{-4} *vs.* 10^{-8} *in my work. Furthermore, the results in [GKW10] have been obtained by a parallel solver on 4 cores, whereas the present results use single core performance only.*

In [MM12] (Section 4.3.1) the authors investigate a partitioned scheme as preconditioned for a monolithic GMRES iteration. Comparable to Section 7.1, the authors investigate the performance of their solver, if all sub-systems are solved by direct inversion. They show nearly robust and good con-



vergence rates. However, in contrast to the results shown in Table 5, convergence does not improve on finer meshes. This supports the assumption, that a multigrid solver with a partitioned smoother (with exact solution of the subproblem) shows better robustness as preconditioner to a GMRES iteration, than a preconditioner, that is simply based on a partitioned iteration. Again, we refer to [vvd08], where exactly this relation was shown for a simple linear problem.

8.1 Robustness versus problem parameters

Next, we have a look at the robustness of the partitioned smoother with regard to different problem parameters. For example variations in the density ratio ρ_f/ρ_s could lead to instabilities due to the *added-mass effect* [van09, van11, CGN05]. In Table 7, we modify different parameters in separate computations. All remaining settings are kept as in the standard configuration, see Table 2. All computations are carried out on mesh-level three with 131 495 unknowns. We indicate the average linear convergence rate over a total of 20 time steps. The results collected in Table 7 show good robustness of the multigrid smoother. It is able to handle large variations of the density ratio as well as variations in the fluid velocity (that will lead to variation of the Reynolds number). Further, we are able to cover nearly incompressible materials without breakdown of the smoother. This will stem from the fact, that the partitioned smoother operation is based on incomplete block-wise decomposition of the matrices, that initially was designed for incompressible flows, see [BB00, KR]. A dramatic effect on the convergence rate is only found for the very small shear modulus $\mu_s = 1.4 \cdot 10^5$, that belongs to a very soft material. This choice results in very large deformation at the base Γ_{base}, where the elastic obstacle is attached to the fluid domain, see Figure 1.

8.2 Analysis of the iterative smoother

In this section, we will take a closer look at the performance of the iterative smoothing process. Altogether, the solution strategy requires a multitude of different parameters. In the previous analysis

Fluid density ρ_f	0.001	0.01	0.1	1
Convergence rate	0.047	0.050	0.046	0.047
Average inflow velocity $\bar{v}_{\text{in}}$	1 m/s	2 m/s	4 m/s	8 m/s
Convergence rate	0.047	0.045	0.045	0.046
Shear modulus μ_s	$2 \cdot 10^5$	$2 \cdot 10^6$	$2 \cdot 10^7$	$2 \cdot 10^8$
Convergence rate	0.092	0.047	0.045	0.045
Poisson's ratio ν_s	0.1	0.2	0.4	0.49
Convergence rate	0.048	0.049	0.047	0.046

Table 7: Convergence rate of the multigrid solver with partitioned smoother depending on variation of different problem parameters. 3d-benchmark problem on a mesh with 131 495 unknowns.

pre - post	0	1	2
0	—	0.047	0.048
1	0.052	0.039	0.023
2	0.050	0.022	0.031

Table 8: 3d-benchmark problem on a mesh with 131 495 unknowns. Convergence rates of the multigrid solver depending on the number of pre- and post-smoothing steps.

we chose the values

1. Newton tolerance of 10^{-8} and assembly of system matrix, if Newton's reduction rate is worse than 10^{-1}.
2. Relative tolerance of linear solver of 10^{-4}.
3. Multigrid with direct solution on the coarse mesh. One step of post-smoothing and no pre-smoothing in case of the splitting smoother, four steps of pre- and post-smoothing for the standard monolithic ILU smoother.
4. In the case of the iterative approximation of the subproblems, four steps of an ILU-preconditioned Richardson iteration for each of the three subproblems.

Table 8 compares different numbers of pre- and post-smoothing steps in the multigrid iteration. Here, this corresponds to the number of Gauss-Seidel iterations described in Section 6.2. The effect of increasing the number of iterations in the smoother is very little. This result corresponds to the findings of Brummelen and coworkers [vvd08] and the results presented in Section 7.1, dealing with the partitioned smoother based on exact solution of the subproblems: if these are approximated with sufficient accuracy, one step of post-smoothing is sufficient to yield good and robust convergence, see Table 5.

Remark 9 (Parallelization) *As the monolithic multigrid solver with partitioned smoother does not require direct inversions of any matrix, parallelization is possible. If the local problems described in Section 6.3 are approximated with a Vanka-type iteration for preconditioning, parallelization is straightforward. If however a more robust approximation based on a block-wise incomplete decomposition is to be used, parallelization has to follow the concept of domain decomposition. See [KR] for an implementation of the geometric multigrid solver used in* Gascoigne 3D *[BBM+] with applications to complex coupled flow problems.*

10 Conclusion

We have presented a monolithic geometric multigrid solver for fluid-structure interactions in ALE formulation. Essential part of the solver is a partitioned iteration in the smoother operator. With exception of the coarse mesh problem, no direct solver is used. In two prototypical benchmark problems, we could demonstrate nearly optimal (linear) performance of the solver in terms of computational time. Furthermore, and of great importance for 3d computations, the memory consumption grows only linearly in the problem size. The solver is very robust with respect to variation of the problem parameters.

It turns out, that great improvement to standard multigrid solvers with monolithic smoother is achieved. These monolithic standard smoothers even completely failed for the 2d test-case with large deformation and a substantial influence of the *added-mass effect*. As a surprising result, we observe optimal performance of the direct solver *UMFPACK* [Dav14] in the case of the 2d-benchmark problem. Here, nearly linear computational times are achieved. On very fine meshes however, the memory consumption of direct solvers increases quickly. Considering 3d-problems, direct solvers are no choice due to both computational time and memory consumption. Considering a problem with about 1 000 000 unknowns, our multigrid solver yields a speedup of 60 compared to a direct solver and a reduction of memory consumption by the factor 30. Regarding the computation on about 7 600 000 unknowns, the final multigrid solver yields a speedup of 6 compared to our robust standard multigrid solver. A direct solver is not applicable due to the enormous memory consumption.

Another benefit of the presented multigrid solver – that has not been exploited so far – is found in the possibility of parallelization, that can easily be applied, as no direct inversion of matrices is required any more. Application of the parallel multigrid solver of Gascoigne 3d [BBM$^+$, KR] will be subject of future work.

References

[BB00] R. Becker and M. Braack. Multigrid techniques for finite elements on locally refined meshes. *Numerical Linear Algebra with Applications*, 7:363–379, 2000. Special Issue.

[BB01] R. Becker and M. Braack. A finite element pressure gradient stabilization for the Stokes equations based on local projections. *Calcolo*, 38(4):173–199, 2001.

[BBM$^+$] R. Becker, M. Braack, D. Meidner, T. Richter, and B. Vexler. The finite element toolkit Gascoigne. http://www.gascoigne.uni-hd.de.

[BBRW99] R. Becker, M. Braack, R. Rannacher, and C. Waguet. Fast and reliable solution of the Navier-Stokes equations including chemistry. *Comput. Visual. Sci.*, 2(2 and 3):107–122, 1999.

[BC10] A. Barker and X.-C. Cai. Scalable parallel methods for monolithic coupling in fluidstructure interaction with application to blood flow modeling. *J. Comp. Phys.*, 229(3):642 – 659, 2010.

[Bna14] S. Bna. *Multilevel domain decomposition algorithms for monolithic fluid-structure interaction problems with application to haemodynamics.* PhD thesis, Universistá di Bologna, 2014.

[BR07] M. Braack and T. Richter. Solving multidimensional reactive flow problems with adaptive finite elements. In W. Jäger, R. Rannacher, and J. Warnatz, editors, *Reactive Flows, Diffusion and Transport*, pages 93–112. Springer Berlin Heidelberg, 2007.

[BS06] H.-J. Bungartz and M. Schäfer, editors. *Fluid-Structure Interaction. Modelling, Simulation, Optimisation*, volume 53 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin, Heidelberg, New York, 2006. ISBN-10: 3-540-34595-7.

[BS10] H.-J. Bungartz and M. Schäfer, editors. *Fluid-Structure Interaction II. Modelling, Simulation, Optimisation*. Lecture Notes in Computational Science and Engineering. Springer, 2010.

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

[CGN05] 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.

[Dav14] T.A. Davis. Umfpack, an unsymmetric-pattern multifrontal method. *ACM Transactions on Math. Soft.*, 30(2):196–199, 2014.

[Don82] J. Donea. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Comput. Methods Appl. Mech. Engrg.*, 33:689–723, 1982.

[Dun07] T. Dunne. *Adaptive Finite Element Approximation of Fluid-Structure Interaction Based on Eulerian and Arbitrary Lagrangian-Eulerian Variational Formulations*. PhD thesis, University of Heidelberg, 2007. urn:nbn:de:bsz:16-opus-79448.

[FGG07] M.A. Fernández, J.-F. Gerbeau, and C. Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *Int. J. Numer. Meth. Engrg.*, 69(4):794–821, 2007.

[FM05] M.A. Fernández and M. Moubachir. A newton method using exact jacobians for solving fluid-structure coupling. *Computers and Structures*, 83:127–142, 2005.

[FWR07] C. Förster, W. Wall, and E. Ramm. Artificial mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer methods Applied Mechanics and Engineering*, 196:1278–1293, 2007.

[GB95] T. Grätsch and K.J. Bathe. Goal-oriented error estimation in the analysis of fluid flows with structural interactions. *Computer Methods in Applied Mechanics and Engineering*, 2006:5673–5684, 195.

[GKW10] M.W. Gee, U. Küttler, and W.A. Wall. Truly monolithic algebraic multigrid for fluid-structure interaction. *Int. J. Numer. Meth. Engrg.*, 85:987–1016, 2010.

[Hei98] M. Heil. Stokes flow in an elastic tube – a large-displacement fluid-structure interaction problem. *Int. J. Numer. Math. Fluids.*, 28:254–265, 1998.

[Hei04] M. Heil. An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems. *Comput. Methods Appl. Mech. Engrg.*, 193:1–23, 2004.

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

[HLZ81] T.J.R. Hughes, W.K. Liu, and T.K. Zimmermann. Lagrangian-eulerian finite element formulations for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29:329–349, 1981.

[HRT92] J.G. Heywood, R. Rannacher, and S. Turek. Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations. *Int. J. Numer. Math. Fluids.*, 22:325–352, 1992.

[HT06a] J. Hron and S. Turek. A monolithic fem solver for an ale formulation of fluid-structure interaction with configuration for numerical benchmarking. In P. Wesseling, E. Onate, and J. Périaux, editors, *European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006*, pages 1–21. TU Delft, 2006.

[HT06b] J. Hron and S. Turek. A monolithic fem/multigrid solver for an ale formulation of fluid-structure interaction with applications in biomechanics. In H.-J. Bungartz and M. Schäfer, editors, *Fluid-Structure Interaction: Modeling, Simulation, Optimization*, Lecture Notes in Computational Science and Engineering, pages 146–170. Springer, 2006.

[HT06c] J. Hron and S. Turek. Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. In H.-J. Bungartz and M. Schäfer, editors, *Fluid-Structure Interaction: Modeling, Simulation, Optimization*, Lecture Notes in Computational Science and Engineering, pages 371–385. Springer, 2006.

[HWL12] G. Hou, J. Wang, and A. Layton. Numerical methods for fluid-structure interaction – a review. *Commun. Comput. Phys.*, 12(2):337–377, 2012.

[KGF+10] U. Küttler, M. Gee, Ch. Förster, A. Comerford, and W.A. Wall. upling strategies for biomedical fluidstructure interaction problems. *Int. J. Numer. Meth. Biomed. Engng.*, 26:305–321, 2010.

[KR] M. Kimmritz and T. Richter. Parallel multigrid method for finite element simulations of complex flow problems on locally refined meshes. *Numerical Linear Algebra with Applications*. doi: 10.1002/nla.744.

[KW08] U. Küttler and W.A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Comp. Mech.*, 43:61–72, 2008.

[LM01] P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacement. *Comput. Methods Appl. Mech. Engrg.*, 190:3039–3067, 2001.

[MM12] R.L. Muddle and M. Mihajlović. An efficient preconditioner for monolithically-coupled large-displacement fluid-structure interaction problems with pseudo-solid mesh updates. *J. Comp. Phys.*, 231:7315–7334, 2012.

[MS02] H.G. Matthies and J. Steindorf. Partitioned but strongly coupled iteration schemes for nonlinear fluid-structure interactions. *Comp. Struct.*, 80:1991–1999, 2002.

[MS03] H.G. Matthies and J. Steindorf. Partitioned strong coupling algorithms for fluid-structure interaction. *Comp. Struct.*, 81:805–812, 2003.

[MW01] D.P. Mok and W.A. Wall. Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. In W.A. Wall, K.-U. Bletzinger, and K. Schweizerhof, editors, *Trends in Computational Structural Mechanics*, pages 689–698. CIMNE, Barcelona, Spain, 2001.

[QQ07] A. Quaini and A. Quarteroni. A semi-implicit approach for fluid-structure interaction based on an algebraic fractional step method. *Math. Models Meth. Appl. Sciences*, 17:957–983, 2007.

[RDH+12] M. Razzaq, H. Damanik, J. Hron, A. Ouazzi, and S. Turek. {FEM} multigrid techniques for fluidstructure interaction with application to hemodynamics. *Applied Numerical Mathematics*, 62(9):1156 – 1170, 2012.

[Ric10] T. Richter. A posteriori error estimation and anisotropy detection with the dual weighted residual method. *Int. J. Numer. Math. Fluids.*, 62(1):90–118, 2010.

[Ric12] T. Richter. Goal oriented error estimation for fluid-structure interaction problems. *Comput. Methods Appl. Mech. Engrg.*, 223-224:28–42, 2012.

[RW] T. Richter and T. Wick. Finite elements fo fluid-structure interaction in ale and fully eulerian coordinates. *Computer Methods in Applied Mechanics and Engineering.* doi: 10.1016/j.cma.2010.04.016).

[RW15] T. Richter and T. Wick. On time discretizations of fluid-structure interactions. In T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, editors, *Multiple Shooting and Time Domain Decomposition Methods*, Contributions in Mathematical and Computational Science. Springer, to appear 2015.

[Saa96] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, 1996.

[Tez01] T.E. Tezduyar. Finite element methods for fluid dynamics with moving boundaries and interfaces. *Arch. Comput. Methods. Eng.*, 8:83–130, 2001.

[TM] Inc The MathWorks. Matlab and statistics toolbox release 2012b. Natick, Massachusetts, United States.

[van09] E van Brummelen. Added mass effects of compressible and incompressible flows in fluid-structure interaction. *J. Appl. Mech.*, 76, 2009.

[van11] E. van Brummelen. Partitioned iterative solution methods for fluid-structure interaction. *Int. J. Numer. Math. Fluids.*, 65:3–27, 2011.

[vvd08] E. van Brummelen, K. van der Zee, and R. de Borst. Space/time multigrid for a fluid-structure-interaction problem. *Applied Numerical Mathematics*, 58(12):1951–1971, 2008.

[WHWT05] T. Washio, T. Hisada, H. Watanabe, and T.E. Tezduyar. A robust preconditioner for fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 194(3941):4027 – 4047, 2005.

[Wic11] T. Wick. Fluid-structure interactions using different mesh motion techniques. *Computers and Structures*, 89:1456–1467, 2011.