ÖAW AUSTRIAN ACADEMY OF SCIENCES

Johann Radon Institute for
Computational and Applied Mathematics
Austrian Academy of Sciences (ÖAW)

RICAM
JOHANN · RADON · INSTITUTE
FOR COMPUTATIONAL AND APPLIED MATHEMATICS

Parallel Block-Preconditioned Monolithic Solvers for Fluid-Structure-Interaction Problems

D. Jodlbauer, U. Langer, T. Wick

RICAM-Report 2018-01

Parallel Block-Preconditioned Monolithic Solvers for Fluid-Structure-Interaction Problems

D. Jodlbauer[1], U. Langer[2], and T. Wick[3]

[1]Doctoral Program on Computational Mathematics, Johannes Kepler University, Altenbergerstr. 69, A-4040 Linz, Austria

[2]Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Altenbergerstr. 69, A-4040 Linz, Austria

[3]Institut für Angewandte Mathematik, Leibniz Universität Hannover, Welfengarten 1, 30167 Hannover, Germany

Abstract

In this work, we consider the solution of fluid-structure interaction problems using a monolithic approach for the coupling between fluid and solid subproblems. The coupling of both equations is realized by means of the arbitrary Lagrangian-Eulerian framework and a nonlinear harmonic mesh motion model. Monolithic approaches require the solution of large, ill-conditioned linear systems of algebraic equations at every Newton step. Direct solvers tend to use too much memory even for a relatively small number of degrees of freedom, and, in addition, exhibit superlinear grow in arithmetic complexity. Thus, iterative solvers are the only viable option. To ensure convergence of iterative methods within a reasonable amount of iterations, good and, at the same time, cheap preconditioners have to be developed. We study physics-based block preconditioners, which are derived from the block LDU-factorization of the FSI Jacobian, and their performance on distributed memory parallel computers in terms of two- and three-dimensional test cases permitting large deformations.

1 Introduction

Fluid-structure interaction problems (FSI) are important in many technical and life science applications. Air flow around the wings of an aircraft or the flow through rotating turbine blades are two of such typical technical examples, see, e.g., [49, 63]. In hemodynamics, the numerical simulation of the vascular blood flow is another FSI example where the interactions between blood flow and the walls of the vessels must be taken into account. The simulation of the human heart and the analysis of aneurysms are important medical applications, see, e.g., [30] and [10], respectively. More FSI applications are compiled in various books [16, 31, 36, 12, 13, 11, 66, 34].

The traditional approach to the solution of FSI problems makes use of available fluid and solid solvers in an alternate iteration between fluid and solid, and an information exchange across the

interface via the interface conditions. This class of solvers are called *partitioned solvers*, see, e.g., [61, 63, 35, 68, 73, 72, 2, 22, 23, 57, 56] and the references therein for recent developments of partitioned methods. Beside the advantage of using available solvers, often provided by different codes, there are several disadvantages of partitioned solvers. These disadvantages are connected with the high complexity, loss of robustness due to so-called added-mass effects, the difficulties connected with the error control of the fluid and solid iterations, and, last but not least, the sequentiality of the alternating iteration process.

These drawbacks are the main reasons why *monolithic solvers* have attracted more and more attention during the last decade; see, e.g., [41, 48, 16, 15, 4, 71, 37, 12, 58, 59, 65]. There are different approaches to construct monolithic solvers for the linear FSI system that arises at every linearization step. Monolithic geometric and algebraic multigrids can be used as solvers or preconditioners in connection with Krylov space methods like GMRES (generalized minimal residuals), see [47, 64, 65] and [37, 60, 59], respectively. Likewise domain decomposition methods can be exploited [9, 79, 6]. Another starting point for deriving efficient preconditioners for Krylov subspace solvers is the block LDU-factorization of the linearized FSI matrix. Different arrangements of the blocks and different approximations of the blocks in the factorization lead to different inexact block LDU-factorizations that can serve as preconditioners in Krylov subspace solvers, see [41, 42, 4, 3, 20, 21, 6, 62, 58, 59] and the references cited there. In the engineering community, this class of preconditioners are also called physics-based block preconditioners. One important advantage of physics-based block preconditioners is their modularity that allows the reuse of available solid, fluid and mesh movment (elliptic) solvers similar to the partitioned approach, but now as a part of the preconditioner.

There are many publications on FSI problems with exciting applications from different areas, but there are to date only a few publications studying the parallel performance of FSI solvers. In [9], an overlapping domain decomposition (additive Schwarz) preconditioner for the GMRES solver that provides an inexact solve of the Jacobian system at each Newton step is proposed. This parallel solution technique, which was developed and tested for two-dimensional FSI problems in this paper, has been extended to three dimensions in [79], see also [55]. This Newton-Krylov-Schwarz FSI solver shows a very good parallel perfomance, at least, for the examples studied in these papers. All numerical tests were performed for characteristic tube-like geometries typically arising in blood flow simulations and under the assumption of geometric and material linear elastic behavior of the solid, whereas the fluid is assumed to be a Newtonian fluid. For the same class of FSI problems, the parallel performance of physics-based block preconditioners were investigated in [20]. Strong and weak scalability studies were presented for a moderate number of cores. In the very recent publications [25] and [32], these studies have been extended to a larger number of cores, again including scalability tests, and to a nonconforming fluid-structure coupling via the internodes technique proposed in [24].

In this paper, we follow basically the developments of [58, 59, 53] to construct a monolithic GMRES solver preconditioned by a physics-based block preconditioner. The major novelty is the extension to high performance computing and a fully parallelized programming code. This code is dimension-independent and can simulate two-dimensional as well as three-dimensional configurations. In 2d, we

consider a challenging FSI benchmark problem [48, 16] with large solid deformations. In 3d, we adopt another benchmark-like configuration [65]. The main goals are to verify the functionals of interest to show that our modeling yields the correct physical results, and scalability tests in order to show the performance of the parallelization.

The remainder of this paper is organized as follows: In Section 2, we recall the governing FSI equation in the Arbitrary Lagrangian-Eulerian (ALE) setting. Its line variational formulation, time discretization, Newton linearization, and spacial discretization are shortly described in Section 3. Section 4 introduces the approximate block-LDU-preconditioner that preconditions the GMRES solver that is used to solve the huge system of algebraic equations arising at each Newton step. The parallel implementation is described in Section 5. In Section 6, we present our numerical results for two- and three-dimensional benchmark problems including convergence and parallel performance studies. Finally, we draw our conclusions.

2 FSI Equations

We denote the computational domain of the fluid-structure interaction problem by $\Omega \subset \mathbb{R}^d$, $d = 2, 3$. This domain is supposed to be time independent but consists of two time dependent subdomains $\Omega_f(t)$ and $\Omega_s(t)$ with a moving interface $\Gamma_i(t) = \partial\Omega_f(t) \cap \partial\Omega_s(t)$. The initial (or later reference) domains are denoted by $\hat{\Omega}_f$ and $\hat{\Omega}_s$, respectively, with the interface $\hat{\Gamma}_i$. Further, we denote the outer boundary with $\partial\hat{\Omega} = \hat{\Gamma} = \hat{\Gamma}_D \cup \hat{\Gamma}_N$ where $\hat{\Gamma}_D$ and $\hat{\Gamma}_N$ denote Dirichlet and Neumann boundaries, respectively. Our philosophy is to transform and solve all equations in $\hat{\Omega}$. A prototype setting of an FSI configuration is displayed in Figure 3 in Section 6. In addition to the spatial domains, we introduce the time interval $I := (0, T]$ where $T > 0$ is the end time value.

Let us now briefly recall the governing equations for fluid-structure interaction. Usually, fluid and solid equations are modelled in different coordinate systems, namely Eulerian and Lagrangian coordinates. In order to couple those equations on a common interface, it is necessary to use a common coordinate system. A popular choice within the FSI framework is the (arbitrary) extension of the Lagrange coordinates from the solid domain into the fluid domain, also called ALE-coordinates [50, 26]. Such an extension is given via the solution of an additional auxiliary problem, e.g. a harmonic extension in our case. A comparison of various extension methods for fluid-structure interaction problems can be found in [76]. This leads to the following system of equations [47, 27, 76]:

Formulation 2.1 (Strong form of FSI equations)**.** Find $\hat{v}_f : I \times \hat{\Omega}_f \to \mathbb{R}^d, \hat{p}_f : I \times \hat{\Omega}_f \to \mathbb{R}, \hat{u}_f : I \times \hat{\Omega}_f \to \mathbb{R}^d, \hat{v}_s : I \times \hat{\Omega}_f \to \mathbb{R}^d, \hat{u}_s : I \times \hat{\Omega}_s \to \mathbb{R}^d$:

$$\begin{aligned}
\hat{J}\hat{\rho}_f\hat{\partial}_t\hat{v}_f + \hat{J}\hat{\rho}_f\hat{\nabla}\hat{v}_f\hat{F}^{-1}\cdot(\hat{v}_f - \partial_t\hat{\mathcal{A}}) - \mathrm{div}_R(\hat{J}\hat{\sigma}_f\hat{F}^{-T}) &= 0 \\
\hat{J}\ \mathrm{tr}(\hat{\nabla}\hat{v}_f\hat{F}^{-1}) &= 0 \\
\\
\hat{\rho}_s\partial_t\hat{v}_s - \mathrm{div}_R(\hat{F}\hat{\Sigma}) &= 0 \\
(\partial_t\hat{u}_s - \hat{v}_f) &= 0 \\
\\
-\mathrm{div}_R(\frac{1}{\hat{J}}\nabla\hat{u}_f) &= 0.
\end{aligned} \tag{1}$$

In the rest of this section, we explain all equations, terms and variables in more detail. The first set of equation describe the incompressible Navier-Stokes equations, the second set of equation describe nonlinear elastodynamics, and in the third set, a nonlinear harmonic mesh motion model [69] is adopted since we are interested in modeling large solid deformations.

The interface and boundary conditions are given by the equations

$$\hat{J}\hat{\sigma}_f\hat{F}^{-T}\hat{n}_f + \hat{F}\hat{\Sigma}_s = 0 \text{ on } I\times\hat{\Gamma}_I \qquad \hat{v}_f = 0 \text{ on } I\times\hat{\Gamma}_{top}\cup I\times\hat{\Gamma}_{bottom}\cup I\times\hat{\Gamma}_c \tag{2}$$

$$\hat{v}_f - \hat{v}_s = 0 \text{ on } I\times\hat{\Gamma}_I \qquad \hat{v}_f = g \text{ on } I\times\hat{\Gamma}_{in} \tag{3}$$

$$\hat{u}_f - \hat{u}_s = 0 \text{ on } I\times\hat{\Gamma}_I \qquad \hat{u}_f = 0 \text{ on } I\times\{\hat{\Gamma}_{top}\cup\hat{\Gamma}_{bottom}\cup\hat{\Gamma}_{in}\cup\hat{\Gamma}_{out}\cup\hat{\Gamma}_c\} \tag{4}$$

$$\hat{u}_s = 0 \text{ on } I\times\hat{\Gamma}_{cf} \tag{5}$$

$$\hat{v}_s = 0 \text{ on } I\times\hat{\Gamma}_{cf} \tag{6}$$

and

$$\hat{J}(-\hat{p}_f\hat{I} + \hat{\rho}_f\hat{\nu}_f\hat{\nabla}\hat{v}_f\hat{F}^{-1})\hat{F}^{-T} = 0 \text{ on } I\times\hat{\Gamma}_{out}. \tag{7}$$

Thus, the flow regime is driven by the prescribed velocity profile g at the inflow boundary $\hat{\Gamma}_{in}$. On the outflow boundary $\hat{\Gamma}_{out}$, the do-nothing condition (7) are given [45], whereas homogeneous boundary conditions are stated otherwise. The initial conditions for the displacements and velocities are assumed to be homogeneous.

The first equation in (1) is the incompressible Navier-Stokes system in ALE-coordinates with the ALE-mapping defined as following:

$$\hat{u}(t,\hat{x}) := \begin{cases} \hat{u}_s(t,\hat{x}) & \hat{x}\in\hat{\Omega}_s \\ \hat{u}_f(t,\hat{x}) & \hat{x}\in\hat{\Omega}_f \end{cases} \qquad \text{and} \qquad \hat{\mathcal{A}}(t,\hat{x}) := \hat{x} + \hat{u}(t,\hat{x}) \text{ for } \hat{x}\in\hat{\Omega}, \tag{8}$$

followed by the equations for the solid. Of course in the solid domain the ALE-mapping is nothing else than the standard coordinate transformation between Lagrangian and Eulerian variables.

The last equation in (1) defines the nonlinear harmonic extension of the solid-displacement into the fluid-domain, yielding the additional mesh-motion variable given by $\hat{u}_f$. The solid displacement and velocity are denoted by $\hat{u}_s$ and $\hat{v}_s$. The fluid velocity and pressure in ALE-coordinates are given by $\hat{u}_f, \hat{p}_f$.

The term $\hat{F} = \hat{I} + \hat{\nabla}\hat{u}$ denotes the gradient of the ALE mapping, and its determinant is given by $\hat{J}$. The solid stress tensor is chosen according to the Saint Venant Kirchhoff material law (STVK) as

$$\hat{\Sigma} = 2\mu\hat{E} + \lambda \mathrm{tr}(\hat{E})\hat{I},$$

with the strain tensor $\hat{E}$ given by $\hat{E} = \frac{1}{2}(\hat{\nabla}\hat{u} + \hat{\nabla}\hat{u}^T + \hat{\nabla}\hat{u}^T\hat{\nabla}\hat{u})$ and the well-known Lamé parameters λ and μ. The fluid stress tensor is given by

$$\hat{\sigma}_f = -\hat{p}_f\hat{I} + \hat{\rho}_f\hat{\nu}_f(\hat{\nabla}\hat{v}_f\hat{F}^{-1} + \hat{F}^{-T}\hat{\nabla}\hat{v}_f^T).$$

Here, $\hat{I}$ is the identity matrix. Moreover, the kinematic viscosity of the fluid is denoted by $\hat{\nu}_f$. Material densities of fluid and solid are denoted by $\hat{\rho}_f$ and $\hat{\rho}_s$, respectively. Interface conditions consist of the coupling of stresses, which propagate forces from one subdomain into the other, as well as the continuity of velocities and displacements resulting from the no-slip condition and the definition of the ALE-extension.

3 Variational Formulation and Discretization

Within the next subsections, we discuss the usual steps to derive a discretized version of the nonlinear, time-dependent FSI system (1) - (7).

3.1 Line Variational Formulation

As usual the finite element discretization starts from a variational formulation of our problem. Special care has to be taken on the interface when defining the test and trial functions. In order to circumvent Bochner function spaces, we define the equations on the time-space continuous level for almost all times and only specify the spatial spaces in more detail. For almost all times t, we seek

$$\begin{aligned}(\hat{u}_s, \hat{u}_f) \in V_u := \{&\hat{u}_s \in V_u^s, \hat{u}_f \in H^1(\hat{\Omega}_f)^d : \hat{u}_s = 0 \text{ on } \partial\hat{\Omega}_s\backslash\hat{\Gamma}_I, \hat{u}_f = 0 \text{ on } \partial\hat{\Omega}_f\backslash\hat{\Gamma}_I, \hat{u}_s = \hat{u}_f \text{ on } \hat{\Gamma}_I\},\\ (\hat{v}_s, \hat{v}_f) \in V_v := \{&\hat{v}_s \in V_v^s, \hat{v}_f \in H^1(\hat{\Omega}_f)^d : \hat{v}_s = 0 \text{ on } \partial\hat{\Omega}_s\backslash\hat{\Gamma}_I, \hat{v}_f = 0 \text{ on } \partial\hat{\Omega}_f\backslash(\hat{\Gamma}_I \cup \hat{\Gamma}_{in}),\\ &\hat{v}_f = g(t) \text{ on } \hat{\Gamma}_{in}, \hat{v}_s = \hat{v}_f \text{ on } \hat{\Gamma}_I\}\end{aligned}$$

and $\hat{p}_f \in V_p := L^2(\hat{\Omega}_f)$. The vector-valued spaces V_u^s and V_v^s denote the respective function spaces for the solid displacement and velocity, taking into account the non-linear structure of the equations, see, e.g., [17, 5]. The test spaces are similar to the trial spaces: for the velocity fields, we take

$$(\hat{\varphi}_s^v, \hat{\varphi}_f^v) \in V_v^0 := V_v^0 := V_v \text{ with } g(t) \equiv 0.$$

For the displacements, we employ the modification $\hat{\varphi}_v^u = 0$ on $\hat{\Gamma}_I$. This avoids a non-physical coupling from the fluid mesh back to the solid displacements. The pressure test space coincides with its trial space.

This leads to the variational formulation of our FSI-equations. Note that the interface terms vanish because of the continuity of stresses and the respective test-functions.

Formulation 3.1 (Line variational formulation of FSI-ALE)**.** Find $((\hat{u}_s, \hat{u}_f), (\hat{v}_s, \hat{v}_f), \hat{p}_f)$ in $V_u \times V_v \times V_p$ such that the following variational equations are satisfied for almost all times $t \in I$:

$$\begin{aligned}
\left(\hat{J}\hat{\rho}_f \hat{\partial}_t \hat{v}_f \ , \ \hat{\varphi}_f^v\right)_{\hat{\Omega}_f} + \left(\hat{J}\hat{\rho}_f \hat{\nabla}\hat{v}_f \hat{F}^{-1} \cdot (\hat{v}_f - \hat{w}) \ , \ \hat{\varphi}_f^v\right)_{\hat{\Omega}_f} + \left(\hat{J}\hat{F}^{-T}\hat{\sigma}_f \ , \ \hat{\nabla}\hat{\varphi}_f^v\right)_{\hat{\Omega}_f} \\
- \left\langle \hat{\rho}_f \hat{\nu}_f \hat{J}\hat{F}^{-T}\hat{\nabla}\hat{v}_f^T \hat{F}^{-T} \cdot \hat{n}_f \ , \hat{\varphi}_f^v \right\rangle_{\hat{\Gamma}_{out}} = 0 \\
\left(\hat{J} \ \mathrm{tr}(\hat{\nabla}\hat{v}_f \hat{F}^{-1}) \ , \ \hat{\varphi}_f^p\right)_{\hat{\Omega}_f} = 0 \\
\left(\hat{\rho}_s \hat{\partial}_t \hat{v}_s \ , \ \hat{\varphi}_s^v\right)_{\hat{\Omega}_s} + \left(\hat{F}\hat{\Sigma}_s \ , \ \hat{\nabla}\hat{\varphi}_s^v\right)_{\hat{\Omega}_s} = 0 \\
\left(\hat{\partial}_t \hat{u}_s - \hat{v}_s \ , \ \hat{\varphi}_s^u\right)_{\hat{\Omega}_s} = 0 \\
\left(\frac{1}{\hat{J}}\hat{\nabla}\hat{u}_f \ , \ \hat{\nabla}\hat{\varphi}_f^u\right)_{\hat{\Omega}_f} = 0
\end{aligned} \tag{9}$$

for all test-functions $\left((\hat{\varphi}_s^u, \hat{\varphi}_f^u), (\hat{\varphi}_s^v, \hat{\varphi}_f^v), \hat{\varphi}_f^p\right)$ in $V_u \times V_v^0 \times V_p$, where $(\cdot,\cdot)_\Omega$ denotes the $L^2(\Omega)$ inner product for vector-functions, and $\langle \cdot, \cdot \rangle_{\hat{\Gamma}_{out}}$ is nothing but the duality product of the trace of functions from $H^1(\hat{\Omega}_f)^d$ on $\hat{\Gamma}_{out}$ with functionals from the dual space.

Remark 3.2 (Well-posedness of the nonlinear fluid mesh motion problem)**.** The nonlinear fluid mesh motion problem is a quasi-linear problem (for the definition of 'quasi-linear' we refer to [28]), which can be analyzed in the framework of monotone operators (chapter 9 of [28]) as long as we can guarantee that $\hat{J} > 0$. Then, the solution $\hat{u}_f$ is indeed a H^1 function. We notice that the chosen function spaces for the fluid and solid subproblems are conforming with the available theory [70, 40, 5, 17]. Of course a complete well-posedness analysis is not available yet in the literature.

Remark 3.3 (Well-posedness of fluid-structure interaction)**.** We assume in the following that there exists a unique smooth solution for the variational FSI problem. For more information on the assumptions imposed on initial data and regularity of the domain to guarantee existence and uniqueness for FSI, we refer for example to [51, 39, 18, 19].

3.2 Discretization in Time

For the time discretization, we employ a one-step theta scheme as given below.

Definition 3.4 (One-Step-Theta Scheme)**.** Given a differential equation $a(u)\partial_t u + A(u) = 0$, the one-step-theta scheme reads as follows:

$$\left(\theta \ a(u^n) + (1-\theta) \ a(u^{n-1}) \ (u(t^n) - u(t^{n-1})\right) - \Delta t \ \theta \ A(u^n) - \Delta t \ (1-\theta) \ A(u^{n-1}) = 0$$

with $\theta \in [0, 1]$ and $\Delta t := t^n - t^{n-1}$.

Different values of θ result in time-stepping schemes with different properties. Popular choices are $\theta = 0$ (Explicit Euler) [78], $\theta = 0.5$ (Crank-Nicolson), $\theta = 0.5 + \Delta t$ (Shifted Crank-Nicolson) [44], $\theta = 1$ (Implicit Euler), or the fractional-step-theta scheme [14].

The explicit Euler method would require too small timesteps in order to be stable, whereas the Implicit Euler method dampens the oscillations too much. Hence, we focus on the Crank-Nicolson method and its shifted variant, since these methods provide the correct results. Detailed computational comparisons of these time-stepping schemes for fluid-structure interaction problems were performed in [75, 67]. In [77] it has been shown that the choice of θ significantly influences the solid displacement. Choosing $\theta = 0.6$ (or even larger up to $\theta = 1$) does not yield any deformation of the elastic beam in the FSI-2 benchmark.

Remark 3.5 (Implicit Pressure)**.** The pressure term within the Navier-Stokes equations is treated fully implicit, i.e., just as in the case $\theta = 1$, independent of the actual choice of θ. This is motivated by the theory of differential algebraic equations, see e.g. [74].

3.3 Linearization

The nonlinearities within our FSI system are treated by Newton's method. Details are presented in [29].

Algorithm 1 Newton Linearization

Let $A(u)(\varphi)$ be a semi-linear form (linear with respect to the second argument), $F(\varphi)$ a linear form. Then the solution of $A(u)(\varphi) = F(\varphi)$ can be obtained by the following iteration:

1: Initial guess u_0
2: **for** $k = 0, 1, \ldots$ until convergence **do**
3: Solve $A'(u_k)(\varphi, \delta u_k) = F(\varphi) - A(u_k)(\varphi)$ for δu_k
4: Update $u_{k+1} := u_k + \delta u_k$
5: **end for**

For the differentiation of the whole FSI operator (11), we need the derivatives of quantities like $\hat{J}, \hat{F}, \hat{F}^{-1}$ and others. Since we will use the notation A' for the full Jacobian, we denote the following derivatives by $\partial_u \hat{F}(u)$ or simply $\partial \hat{F}$ etc., if it is clear by which variable we are differentiating.

Theorem 3.6 (FSI-related Derivatives)**.** *It holds:*

$$\begin{aligned} \partial \hat{F} &= \hat{\nabla} \delta \hat{u} \\ \partial \hat{J} &= \hat{J}\, tr(\hat{F}^{-1} \hat{\nabla} \delta \hat{u}) \\ \partial \hat{F}^{-1} &= -\hat{F}^{-1} \hat{\nabla} \delta \hat{u} \hat{F}^{-1} \\ \partial \hat{F}^{-T} &= (\partial \hat{F}^{-1})^T \\ \partial tr(E) &= tr(\partial E) \end{aligned} \tag{10}$$

Proof. Unless trivial, see for example [46, 1]. □

The semi-linear form $A(U)(\Phi)$ and linear form $F(\Phi)$ are given as the sum of all equations in (9).

$$
\begin{aligned}
A(U)(\Phi) = &\left(\hat{\rho}_f \hat{J}^\theta (\hat{v}_f - \hat{v}_f^{n-1}) \ , \ \hat{\varphi}_f^v\right)_{\hat{\Omega}_f} - \left(\hat{J}\hat{\rho}_f \hat{F}^{-1}\hat{\nabla}\hat{v}_f \cdot \hat{u}_f \ , \ \hat{\varphi}_f^v\right)_{\hat{\Omega}_f} \\
&+ \Delta t \ \theta \left[\left(\hat{J}\hat{\rho}_f \hat{F}^{-1}\hat{\nabla}\hat{v}_f \cdot \hat{v}_f \ , \ \hat{\varphi}_f^v\right)_{\hat{\Omega}_f} + \left(\hat{J}\hat{\sigma}_f^v \hat{F}^{-T} \ , \ \hat{\nabla}\hat{\varphi}_f^v\right)_{\hat{\Omega}_f}\right] \\
&- \Delta t \ \theta \left\langle \hat{\rho}_f \hat{\nu}_f \hat{J}\hat{F}^{-T}\hat{\nabla}\hat{v}_f^T \hat{F}^{-T} \cdot \hat{n}_f \ , \hat{\varphi}_f^v \right\rangle_{\hat{\Gamma}_{out}} \\
&+ \left(\hat{J}(-\hat{p}_f)\hat{F}^{-T} \ , \ \hat{\nabla}\hat{\varphi}_f^v\right)_{\hat{\Omega}_f} \\
&+ (\hat{\rho}_s \hat{v}_s^n \ , \ \hat{\varphi}_s^v)_{\hat{\Omega}_s} + \Delta t \ \theta \left(\hat{F}\hat{\Sigma}_s \ , \ \hat{\nabla}\hat{\varphi}_s^v\right)_{\hat{\Omega}_s} \\
&+ (\hat{u}_s \ , \ \hat{\varphi}_s^u)_{\hat{\Omega}_s} - \Delta t \ \theta \ (\hat{v}_s \ , \ \hat{\varphi}_s^u)_{\hat{\Omega}_s} \\
&+ \left(\frac{1}{\hat{J}}\hat{\nabla}\hat{u}_f \ , \ \hat{\nabla}\hat{\varphi}_f^u\right)_{\hat{\Omega}_f}
\end{aligned}
\tag{11}
$$

and

$$
\begin{aligned}
F(\Phi) = &\left(\hat{\rho}_s \hat{v}_s^{n-1} \ , \ \hat{\varphi}_s^v\right)_{\hat{\Omega}_s} - \Delta t \ (1-\theta) \left(\hat{F}^{n-1}\hat{\Sigma}_s^{n-1} \ , \ \hat{\nabla}\hat{\varphi}_s^v\right)_{\hat{\Omega}_s} \\
&+ \left(\hat{u}_s^{n-1} \ , \ \hat{\varphi}_s^u\right)_{\hat{\Omega}_s} + \Delta t \ (1-\theta) \ \left(\hat{v}_s^{n-1} \ , \ \hat{\varphi}_s^u\right)_{\hat{\Omega}_s} \\
&- \left(\hat{J}\hat{\rho}_f \hat{F}^{-1}\hat{\nabla}\hat{v}_f \cdot \hat{u}_f^{n-1} \ , \ \hat{\varphi}_f^v\right)_{\hat{\Omega}_f} \\
&- \Delta t \ (1-\theta) \left[\left(\hat{J}\hat{\rho}_f \hat{F}^{-1}\hat{\nabla}\hat{v}_f \cdot \hat{v}_f \ , \ \hat{\varphi}_f^v\right)_{\hat{\Omega}_f} + \left(\hat{J}\hat{\sigma}_f^v \hat{F}^{-T} \ , \ \hat{\nabla}\hat{\varphi}_f^v\right)_{\hat{\Omega}_f}\right]^{n-1} \\
&+ \Delta t \ (1-\theta) \left[\left\langle \hat{\rho}_f \hat{\nu}_f \hat{J}\hat{F}^{-T}\hat{\nabla}\hat{v}_f^T \hat{F}^{-T} \cdot \hat{n}_f \ , \hat{\varphi}_f^v \right\rangle_{\hat{\Gamma}_{out}}\right]^{n-1}
\end{aligned}
\tag{12}
$$

with the test-function $\Phi := (\hat{\varphi}_f^v, \hat{\varphi}_f^u, \hat{\varphi}_f^p, \hat{\varphi}_s^v, \hat{\varphi}_s^u)$ and the solution variable $U := (\hat{v}_f, \hat{u}_f, \hat{p}_f, \hat{v}_s, \hat{u}_s)$.

Due to the nested non-linearities, which all require multiple applications of the product rule, the computation of all single terms of the Jacobian is quite lengthy. However, when implementing the derivatives, we do not actually need to expand all terms. Therefore, we do not write them down explicitly here for the convenience of the reader.

3.4 Spatial Discretization

Now we are in the position to discretize the Jacobian and Newton residual given in Algorithm 1. We are going to use a quadrilateral ($2d$) or hexahedral ($3d$) subdivision $\mathcal{T}_h$ of the reference domain $\overline{\hat{\Omega}} = \bigcup_{T\in\mathcal{T}_h} \overline{T}$ with $Q(k)$ shape functions for displacements and velocities, and discontinuous $P(k-1)$ elements for the pressure, where h denotes the usual discretization parameter. The subdivision matches the interface, i.e., every element T is either part of the fluid or the solid domain, but not both.

As indicated above we use the following discrete function spaces

$$
\begin{aligned}
V_h^s &:= \left\{ v \in H^1(\hat{\Omega}_s)^d : v\big|_T \in Q(k) \quad \forall \ T \subset \hat{\Omega}_s \right\}, \\
V_h^f &:= \left\{ v \in H^1(\hat{\Omega}_f)^d : v\big|_T \in Q(k) \quad \forall \ T \subset \hat{\Omega}_f \right\}, \\
L_h^f &:= \left\{ v \in L^2(\hat{\Omega}_f) : v\big|_T \in P(k-1) \ \forall \ T \subset \hat{\Omega}_f \right\}
\end{aligned}
\tag{13}
$$

with the nodal basis functions

$$\begin{aligned} V_h^s &= span\{\varphi_{s,h}^v[j],\ j=1,\dots,N_{v_s}\} = span\{\varphi_{s,h}^u[j],\ j=1,\dots,N_{u_s}\},\\ V_h^f &= span\{\varphi_{f,h}^v[j],\ j=1,\dots,N_{v_f}\} = span\{\varphi_{f,h}^u[j],\ j=1,\dots,N_{u_f}\},\\ L_h^f &= span\{\varphi_{f,h}^p[j],\ j=1,\dots,N_{p_f}\}. \end{aligned} \tag{14}$$

Using the ansatz

$$\delta U := \sum_{j=1}^{N_{FSI}} \delta U_j \Phi_j$$

for the Newton correction at each step k leads to the linear system $A_h \delta U_h = F_h$, with the system matrix

$$A_h := \begin{bmatrix} \mathcal{M} & \mathcal{C}_{ms} & 0 \\ \mathcal{C}_{sm} & \mathcal{S} & \mathcal{C}_{sf} \\ \mathcal{C}_{fm} & \mathcal{C}_{fs} & \mathcal{F} \end{bmatrix}, \tag{15}$$

where $\mathcal{M}, \mathcal{S}$ and $\mathcal{F}$ denote the discrete versions of the mesh-motion, solid and fluid equations, respectively. The coupling terms $\mathcal{C}_{**}$ arise because of the ALE transformation and interface coupling conditions.

4 Approximate Block-LDU - Preconditioner

4.1 Approximate factorization

By treating our block system as a simple 3×3 matrix, we can (formally) apply an LDU-factorization, yielding the following decomposition:

$$\begin{bmatrix} I & 0 & 0 \\ \frac{\mathcal{C}_{sm}}{M} & I & 0 \\ \frac{\mathcal{C}_{fm}}{M} & \frac{\mathcal{C}_{fs}-\frac{\mathcal{C}_{fm}}{M}}{\mathcal{S}-\frac{\mathcal{C}_{sm}\mathcal{C}_{ms}}{\mathcal{M}}} & 1 \end{bmatrix} \begin{bmatrix} \mathcal{M} & 0 & 0 \\ 0 & \mathcal{S}-\frac{\mathcal{C}_{sm}\mathcal{C}_{ms}}{\mathcal{M}} & 0 \\ 0 & 0 & \mathcal{F}-\frac{\mathcal{C}_{sf}\left(\mathcal{C}_{fs}-\frac{\mathcal{C}_{fm}\mathcal{C}_{ms}}{\mathcal{M}}\right)}{\mathcal{S}-\frac{\mathcal{C}_{sm}\mathcal{C}_{ms}}{\mathcal{M}}} \end{bmatrix} \begin{bmatrix} I & \frac{\mathcal{C}_{ms}}{\mathcal{M}} & 0 \\ 0 & I & \frac{\mathcal{C}_{sf}}{\mathcal{S}-\frac{\mathcal{C}_{sm}\mathcal{C}_{ms}}{\mathcal{M}}} \\ 0 & 0 & I \end{bmatrix} \tag{16}$$

making slight abuse of the notation by using $\frac{A}{B}$ as shortcut for AB^{-1} or $B^{-1}A$. Using this LDU-factorization directly is obviously not very efficient since it involves the computation of too many inverses (displayed as fractions) and matrix-matrix products.

In order to simplify the computation, we drop the term $\mathcal{C}_{sm}$ from our system. This is justified by the fact that this coupling term corresponds to the term which is set to zero on the interface to avoid the (non-physical) coupling from the mesh into the solid equations.

The simplified LDU decomposition is then given as

$$\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \frac{\mathcal{C}_{fm}}{\mathcal{M}} & \frac{\tilde{\mathcal{C}}_{fs}}{\mathcal{S}} & 1 \end{bmatrix} \begin{bmatrix} \mathcal{M} & 0 & 0 \\ 0 & \mathcal{S} & 0 \\ 0 & 0 & \mathcal{X} \end{bmatrix} \begin{bmatrix} I & \frac{\mathcal{C}_{ms}}{\mathcal{M}} & 0 \\ 0 & I & \frac{\mathcal{C}_{sf}}{\mathcal{S}} \\ 0 & 0 & I \end{bmatrix} \tag{17}$$

with $\tilde{\mathcal{C}}_{fs} = \mathcal{C}_{fs} - \frac{\mathcal{C}_{fm}\mathcal{C}_{ms}}{\mathcal{M}}$ and $\mathcal{X} = \mathcal{F} - \frac{\mathcal{C}_{fs}\tilde{\mathcal{C}}_{sf}}{\mathcal{S}}$.

In order to apply solvers like sparse-LU or AMG methods, which require explict knowledge about the matrix entries, we would have to compute $\mathcal{X}$ explicitly. However, this involves the explicit computation of $\mathcal{S}^{-1}$, which we want to avoid. One possibility is to replace $\mathcal{S}$ by a block-diagonal approximation that can be easily inverted, as shown in [58]. In our application, we simply ignore the perturbation $\frac{\mathcal{C}_{sf}\tilde{\mathcal{C}}_{fs}}{\mathcal{S}}$, and observe almost no difference compared to the block-diagonal approximation. This factorization can now be applied to our block-system as given in Algorithm 2.

Remark 4.1. The LDU-factorization depends on the ordering of the blocks and is therefore not unique, giving rise to many different preconditioners. For a comparison of some of them as well as further numerical results we refer to our previous work [53].

Algorithm 2 Evaluation of $P^{-1}r$.

1: Solve $x_m = \mathcal{M}^{-1}r_m$
2: Solve $x_s = \mathcal{S}^{-1}r_s$
3: Solve $x_f = \mathcal{F}^{-1}(r_f - \mathcal{C}_{fm}x_m - \mathcal{C}_{fs}x_s)$
4: Update $x_s = x_s - \mathcal{S}^{-1}\mathcal{C}_{sf}x_f$
5: Update $x_m = x_m - \mathcal{M}^{-1}\mathcal{C}_{ms}x_s$

The approximate solution of the subproblems, i.e., application of the inverses in Algorithm 2, is discussed in the next sections.

4.2 Solving the Mesh Subproblem

In our configuration, the mesh-motion equation is a scaled Laplace-type equation. Thus, available AMG methods like those provided by the Trilinos package [43, 38] are good candidates for the approximation of $\mathcal{M}^{-1}$.

4.3 Solving the Solid Subproblem

For the solution of the solid equations, we employ a Schur-complement approach, which eliminates the equations related to the solid velocity $\hat{v}_s$. Here, matrices M_{**} denote mass-matrices, and K denotes the matrix resulting from $\left(\hat{F}\hat{\Sigma}_s\ ,\ \hat{\nabla}\hat{\varphi}_s^v\right)_{\hat{\Omega}_s}$. The solution of the linear system

$$\begin{bmatrix} \hat{\rho}_s M_{vv} & \Delta t\ \theta K_{vu} \\ -\Delta t\ \theta M_{uv} & M_{uu} \end{bmatrix} \begin{bmatrix} x_{v_s} \\ x_{u_s} \end{bmatrix} = \begin{bmatrix} r_{v_s} \\ r_{u_s} \end{bmatrix}$$

is equivalent to solving

$$\begin{bmatrix} \hat{\rho}_s M_{vv} + \Delta t^2\ \theta^2 K_{vu} M_{uu}^{-1} M_{uv} & 0 \\ -\Delta t\ \theta M_{uv} & M_{uu} \end{bmatrix} \begin{bmatrix} x_{v_s} \\ x_{u_s} \end{bmatrix} = \begin{bmatrix} r_{v_s} - \Delta t\ \theta K_{vu} M_{uu}^{-1} r_{u_s} \\ r_{u_s} \end{bmatrix}$$

Since we use equal-order elements for the displacement and velocity variables, the mass matrices M_{**} are all equal (possibly after reordering the dofs). Hence, we can simplify the system above to

$$\begin{bmatrix} \hat{\rho}_s M + \Delta t^2\ \theta^2 K_{vu} & 0 \\ -\Delta t\ \theta\ M & M \end{bmatrix} \begin{bmatrix} x_{v_s} \\ x_{u_s} \end{bmatrix} = \begin{bmatrix} r_{v_s} - \Delta t\ \theta K_{vu} M^{-1} r_{u_s} \\ r_{u_s} \end{bmatrix} \tag{18}$$

This system is then solved in a similar fashion as the global system:

Algorithm 3 Evaluation of $P^{-1}r$.

1: Solve $x_{v_s} = (\hat{\rho}_s M + \Delta t^2\ \theta^2 K_{vu})^{-1}(r_{v_s} - \Delta t\ \theta K_{vu} M^{-1} r_{u_s})$

2: Solve $x_{u_s} = M^{-1}(r_{u_s} + \Delta t\ \theta M x_{v_s})$

We do not require the exact realization of the application of the occurring inverses, but again use an AMG-solver to obtain reasonable approximations to $(\hat{\rho}_s M + \Delta t^2\ \theta^2 K_{vu})^{-1}$ and M^{-1}.

4.4 Solving the Fluid Subproblem

For the fluid inverse $\mathcal{F}^{-1}$, we employ the same Schur-complement approach as for the solid system (an Uzawa-like method). Therein, we use an AMG-preconditioned GMRES solver to approximate the action of the occurring inverses. The additional solver was necessary to cope with large solid deformations, because a standalone ML-AMG (multi-level algebraic multigrid) method did no longer suffice.

5 Parallel Implementation

The implementation uses the C++ library deal.II [8, 7], with the Trilinos package [43] for linear algebra operations and its multi-level package ML [38]. Partitioning of the mesh is done using ParMETIS [54].

5.1 Mesh Partitioning

In a distributed setting, each core only stores parts of the problem. Hence, the mesh is split into several subdomains using ParMETIS. In our tests, we considered the following splitting strategies:

- shared: each core owns parts of both, the fluid and the solid domain
- split: each core owns either parts of the fluid or the solid domain, but not both
- default: no distinction is made between fluid and solid subdomains

Figure 1: Shared configuration using four subdomains.

Obviously, the split-strategy suffers from load balancing issues if a small number of cores is used, since it is not possible to obtain an almost uniform distribution of the dofs. This kind of problem is avoided using the shared-type partitioning. On the other hand, the split-strategy may require less communication than the shared-strategy due to the separation of fluid and solid. In addition to the owned cells, each core also obtains information about the neighboring cells, which are referred to as ghost cells or ghost layer.

Figure 2: Split configuration using four subdomains. Load balancing issues are obvious.

5.2 Interface coupling

We use global variables for the displacement, velocity and pressure. The benefit of this approach is the fact that all interface conditions, namely the continuity of shape-functions, velocities and displacements, are automatically fulfilled. The natural interface condition (2), the continuity of normal stresses, is fulfilled in a weak sense.

5.3 Distribution of dofs

Degrees of freedoms are distributed in a similar manner as the mesh. Each CPU owns the dofs located in the interior of its associated subdomain. Dofs located on the interface between different cores are assigned to one of the adjacent ones. Hence, each dof is owned by exactly one core, although possibly more cores may acquire information about this dof. The set of all dofs that are required by a specific core is called the set of locally relevant dofs within the deal.II library.

6 Numerical Results

In this section, we consider two test cases to demonstrate the computational performance of the solvers presented in Section 4. In the first example, we focus on the FSI-2 benchmark [48], which exhibits large solid deformations and is a well-known difficult test problem. The second example, which describes the flow around an obstacle, is a three-dimensional configuration inspired by [65]. In both test cases, we first focus on the correct physics and reproduce the quantities of interest published in the literature. Then, we discuss the parallel performance of the preconditioner from Section 4. We notice that numerical results regarding robustness, h and Δt-dependence have been presented in [52, 53] for sequential computations. Consequently, in this work, we focus on the parallel scalability.

All tests regarding parallelization were done on the distributed memory cluster Radon1[1] at RICAM, Linz. Radon1 consists of 64 compute nodes each with two 8-core Intel Haswell processors (Xeon E5-2630v3, 2.4Ghz) and 128 GB of memory.

In all our tests, all partitioning strategies (shared, split, default) yielded similar scalability results. However, the shared-type partition required significantly less time than the others. Unless stated otherwise, all figures below are done using the faster shared-type strategy.

For solving the nonlinear problem we employ a Quasi-Newton scheme, which reassembles the Jacobian only if the reduction of the last Newton iteration is less than 10, i.e. assemble if $\|r_k\|_\infty > 0.1\,\|r_{k-1}\|_\infty$.

[1] https://www.ricam.oeaw.ac.at/hpc/overview/

The Newton solver is stopped once the nonlinear residual r satisfies $\|r_k\|_\infty < 10^{-6}\|r_0\|_\infty$, where r_0 is the initial Newton residual.

GMRES iterates until a reduction of 10^3 is achieved, which yields only a slight increase in the same number of Newton iterations compared to solving the linear systems with a direct method.

6.1 Example 1: FSI-2 benchmark

6.1.1 Description

The geometry consists of a channel with some given inflow velocity profile on the left boundary. Inside the channel is a fixed cylindric obstacle with an elastic beam attached to it, as depicted in Figure 3.

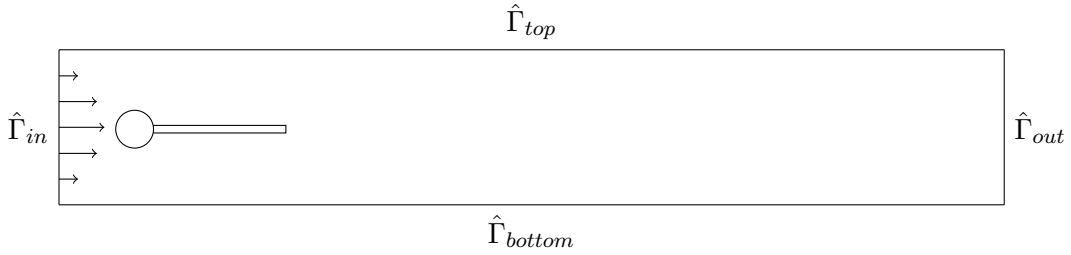

Figure 3: Geometry of the FSI-2 benchmark [48]: Elastic beam immersed in a flow around a cylinder.

The inflow is given by $\hat{v}_f(t,(0,y)) = 6\frac{y(H-y)}{H^2}s(t)\bar{v}$, where the height of the channel H is given by $H = 0.41$, and $s(t) = \frac{1}{2}(1-\cos(\frac{\pi}{2}t))$ is nothing but a time-dependent smoothing factor, and $\bar{v} = 1$ is the mean inflow velocity. The other quantities of the geometry are given as follows:

Quantity	Value
channel length	2.5
channel height	0.41
cylinder center	$(0.2, 0.2)$
cylinder radius	0.05
beam thickness	0.02
beam length	0.35
reference point	$(0.6, 0.2)$

Quantity	Value
inflow velocity $\bar{v}$	1.0
density	10^3
viscosity	10^{-3}

Quantity	Value
solid density $\hat{\rho}_s$	10^4
Lamé λ	$2 \cdot 10^6$
Lamé μ	$0.5 \cdot 10^6$
Poisson ratio	0.4

We refer the reader to FeatFlow[2] or [48] for a more detailed description of FSI-2 benchmark.

[2] http://www.featflow.de/en/benchmarks.html

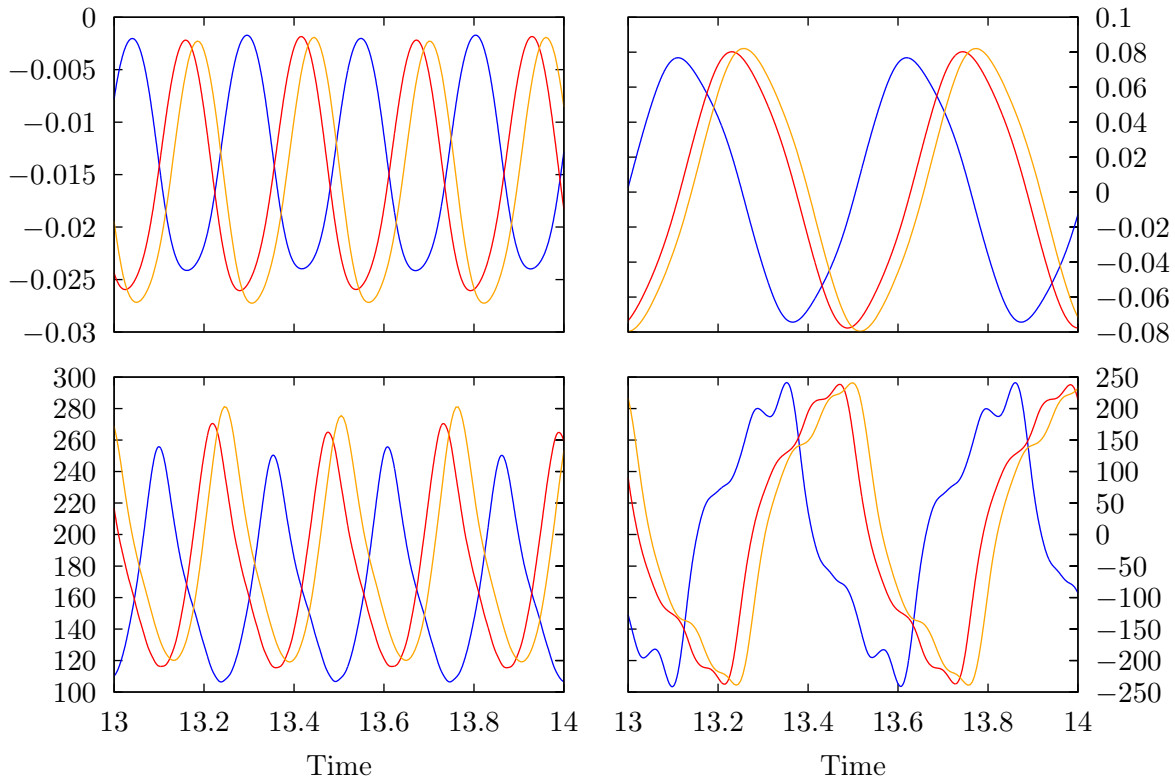

Figure 4: FSI-2 benchmark: Drag, lift (bottom row) and x/y-displacements (top row) using different levels r of refinements for $Q(1) - Q(1) - P(0)$ (blue: $r = 2$, red: $r = 3$, orange: $r = 4$).

6.1.2 Computing and comparison of displacements, drag and lift

To check the correctness of our code, we first compare the evolution of the deflection of the tip of the elastic beam, drag and lift with the results published in the literature [48, 16, 76].

The displacement is measured at the reference point $(0.6, 0.2)$, located at the very end of the beam. Drag F_D and lift F_L are computed according to the following formula:

$$(F_D, F_L) = \int_S \hat{\sigma}_f \hat{n}_f ds, \tag{19}$$

where S denotes the boundary of flag and circle with the outer unit normal vector $\hat{n}_f$ pointing inside the solid domain. The results of these computations are shown in Figure 4, using a $Q(1) - Q(1) - P(0)$ discretization, and in Figure 5 for $Q(2) - Q(2) - P(1)$. All configurations yield similar results, but a slight shift in time can be observed.

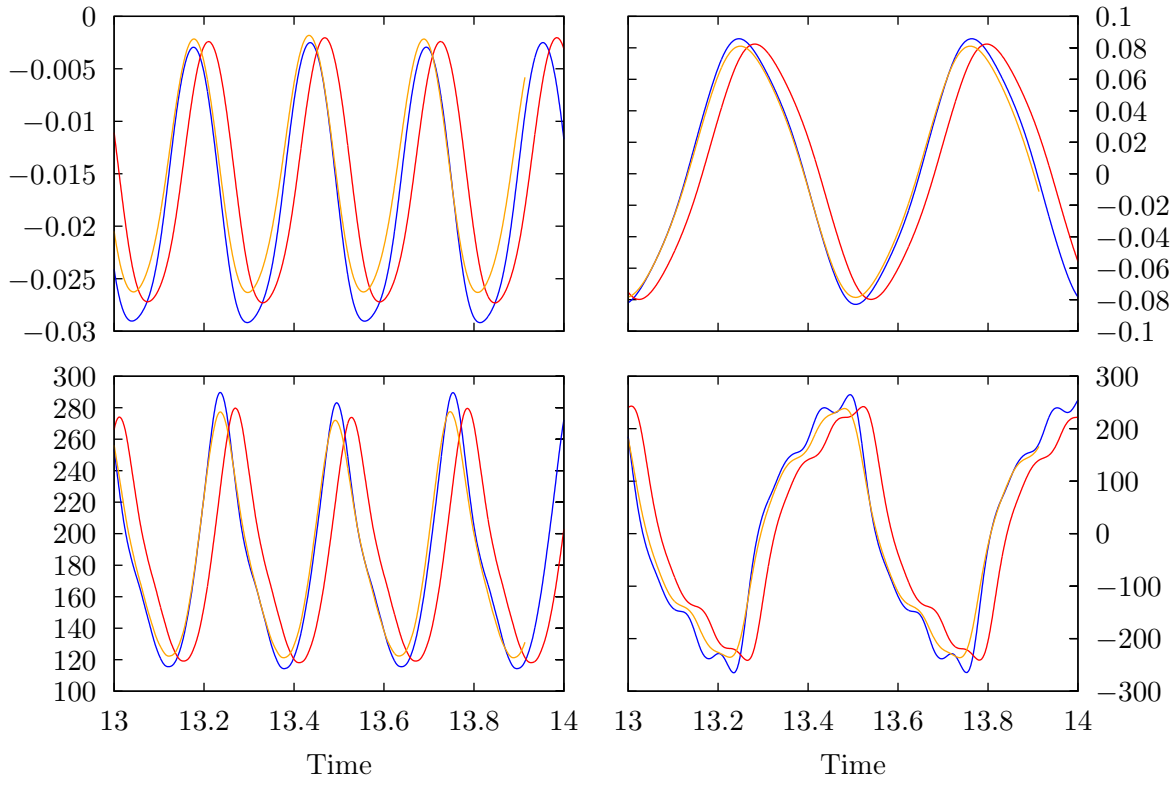

Figure 5: FSI-2 benchmark: Drag, lift (bottom row) and x/y-displacements (top row) using different levels r of refinements for $Q(2) - Q(2) - P(1)$ (blue: $r = 1$, red: $r = 2$, orange: $r = 3$).

6.1.3 Parallel performance studies

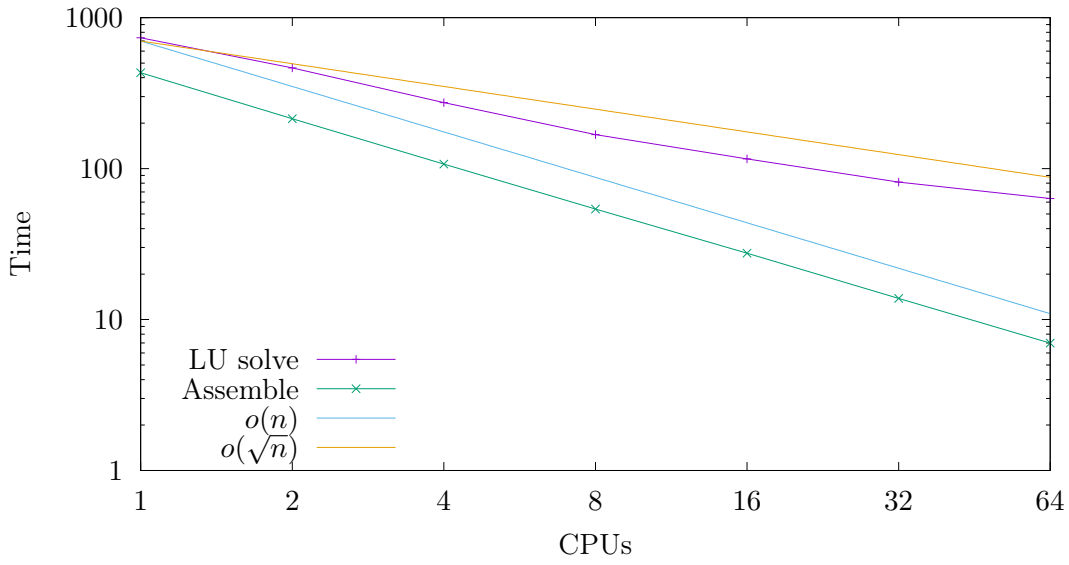

Figure 6: FSI-2 benchmark: Strong scalability of the sparse direct solver MUMPS.

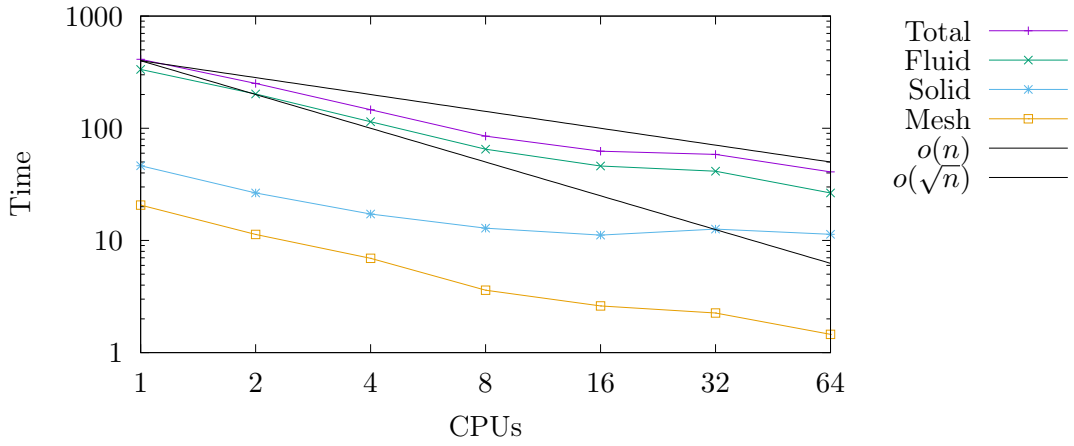

Figure 7: FSI-2 benchmark: Strong scalability using the preconditioned GMRES scheme for approximately $16 \cdot 10^6$ dofs. Average time given in seconds for the solution of one linear system.

For comparison, Figure 6 shows the parallel performance of the sparse direct solver MUMPS solving one monolithic linear system from the FSI-2 simulation. For a small number of cores, the speedup is somewhere between $\mathcal{O}(n)$ and $\mathcal{O}(\sqrt{n})$, but decays when more CPUs are added. Furthermore, it displays the expected perfect scalability of the assembling procedure, rendering the linear solver as the only remaining bottleneck in serial and parallel computations.

Our preconditioned GMRES yields $\mathcal{O}(\sqrt{n})$ scalability as shown in Figure 7. This behavior is in agreement with the results presented in [20]. We note that both methods do not yield the optimal

scalability of $\mathcal{O}(n)$.

The time required for solving the linear system is dominated by the solution time spent for the fluid sub-problem. The mesh problem is the easiest one to solve, requiring just a few AMG-cycles, hence contributing only little to the overall runtime. The solid problem does not scale very well, most likely due to its small size in comparison to the other problems.

Specifically, using 1 core, the total CPU time to solve the problem with $16 \cdot 10^6$ dofs at a single time step is 411 seconds, i.e., approximately 6.8 minutes. On 4 cores, the computational cost decreases to 146 seconds, i.e., 2.4 minutes. Thus we achieve a reduction by 64% of the computational time. The further decrease using 64 cores is less significant and drops to 41 seconds per solution of the linear system.

6.2 Example 2: Flow around an elastic obstacle

6.2.1 Description

This numerical test features a $3d$ flow around an elastic obstacle and is motivated from [65]. The computational domain is given by $(0, L) \times (0, H) \times (-H, H)$, with the solid inclusion $(0.4, 0.5) \times (0, h) \times (-0.2, 0.2)$. Similar to the previous test, an inflow velocity is prescribed on the yz-plane by $\hat{v}_f(t, (0, y, z)) = \frac{81}{16} \frac{y(H-y)(H^2-z^2)}{H^4} s(t)\bar{v}$. The geometrical parameters and $\bar{v}$ are given in Table 1, whereas the fluid and solid parameters are chosen the same as in the FSI-2 benchmark. The geometry is illustrated in Figure 8.

Quantity	Value
channel length (x-direction) L	1.5
channel height (y-direction) H	0.4
channel width (z-direction) $2H$	0.8
obstacle height h	0.3
obstacle width	0.4
obstacle thickness	0.1
inflow velocity $\bar{v}$	3.0

Table 1: Geometry and problem data for Example 2.

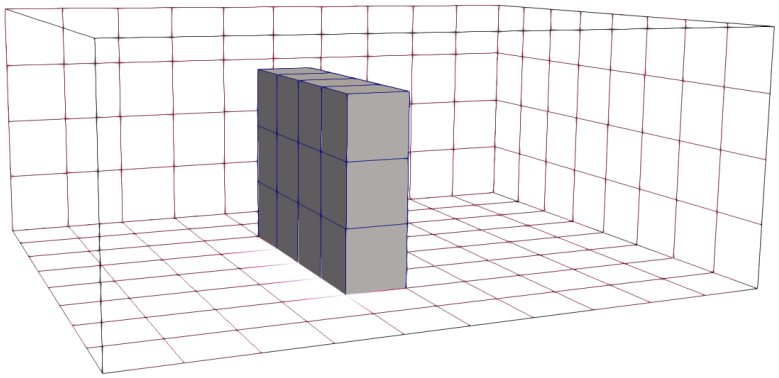

Figure 8: Graphical illustration of the geometry of Example 2. The elastic obstacle is displayed in dark color. The explanation of the axes is given in Table 1.

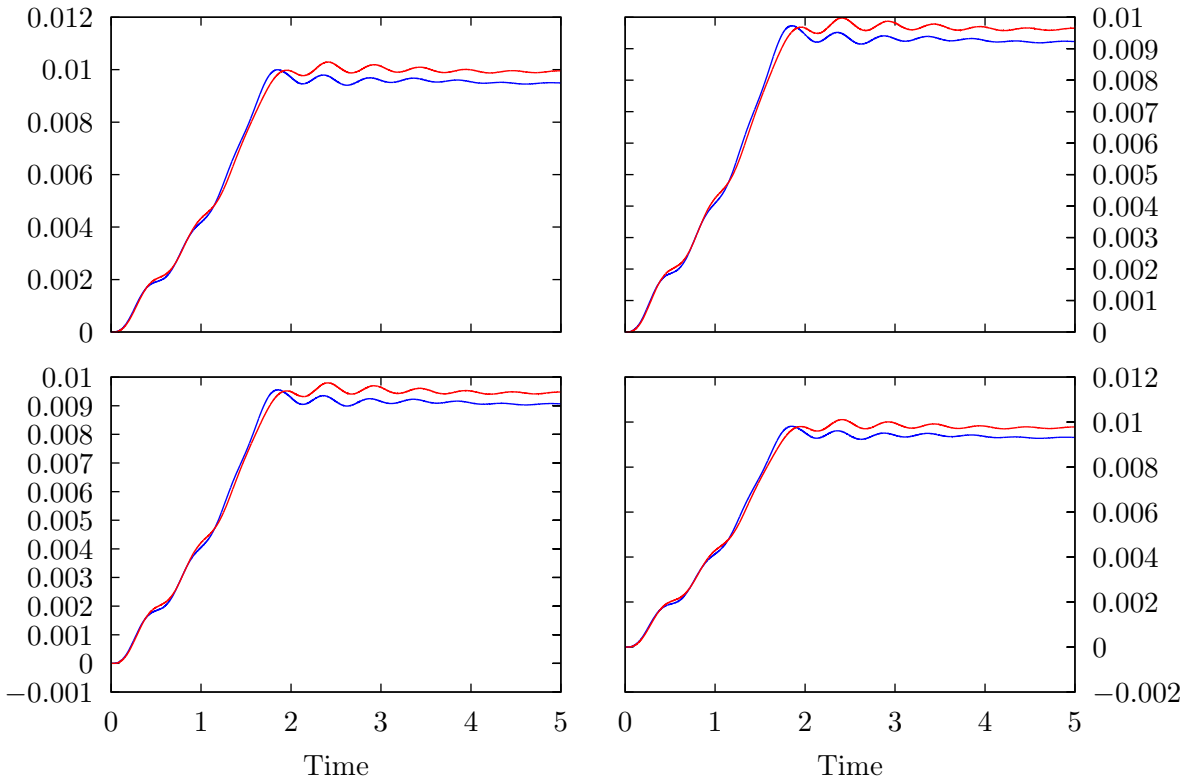

Figure 9: Example 2: x-component of displacement $u(P_i)$ using different levels r of refinements (blue: $r = 1$, red $r = 2$).

6.2.2 Evaluation of quantities of interest

As in Section 6.1.2, we first compute the physical quantities of interest. This includes point evaluations at the upper boundary surface ($y = h$) of the solid obstacle at points given below The results are presented component-wise in Figures 9 – 11 using the scheme depicted in Table 2.

$P_1 = (0.4, h, \quad 0.0)$	$P_2 = (0.4, h, -0.2)$
$P_3 = (0.5, h, -0.2)$	$P_4 = (0.5, h, \quad 0.0)$

Table 2: Example 2: Evaluation points for the displacement of the elastic obstacle.

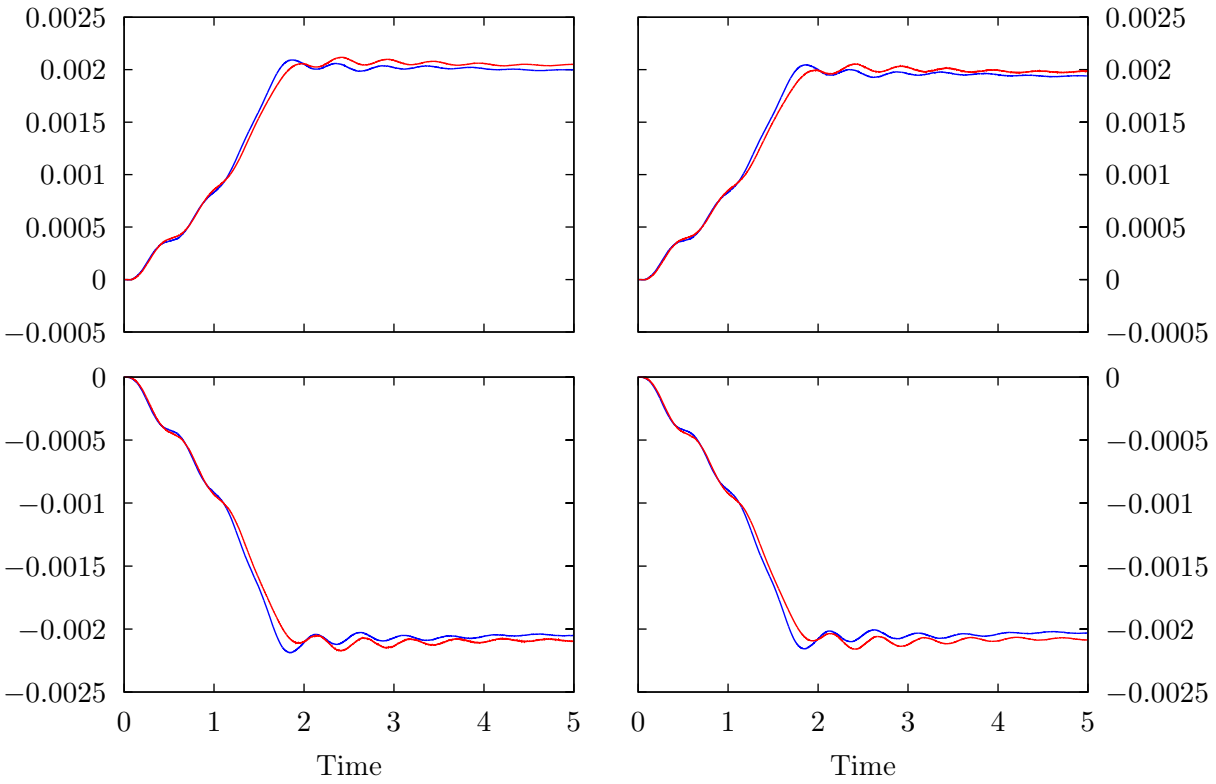

Figure 10: Example 2: y-component of displacement $u(P_i)$ using different levels r of refinements (blue: $r = 1$, red $r = 2$).

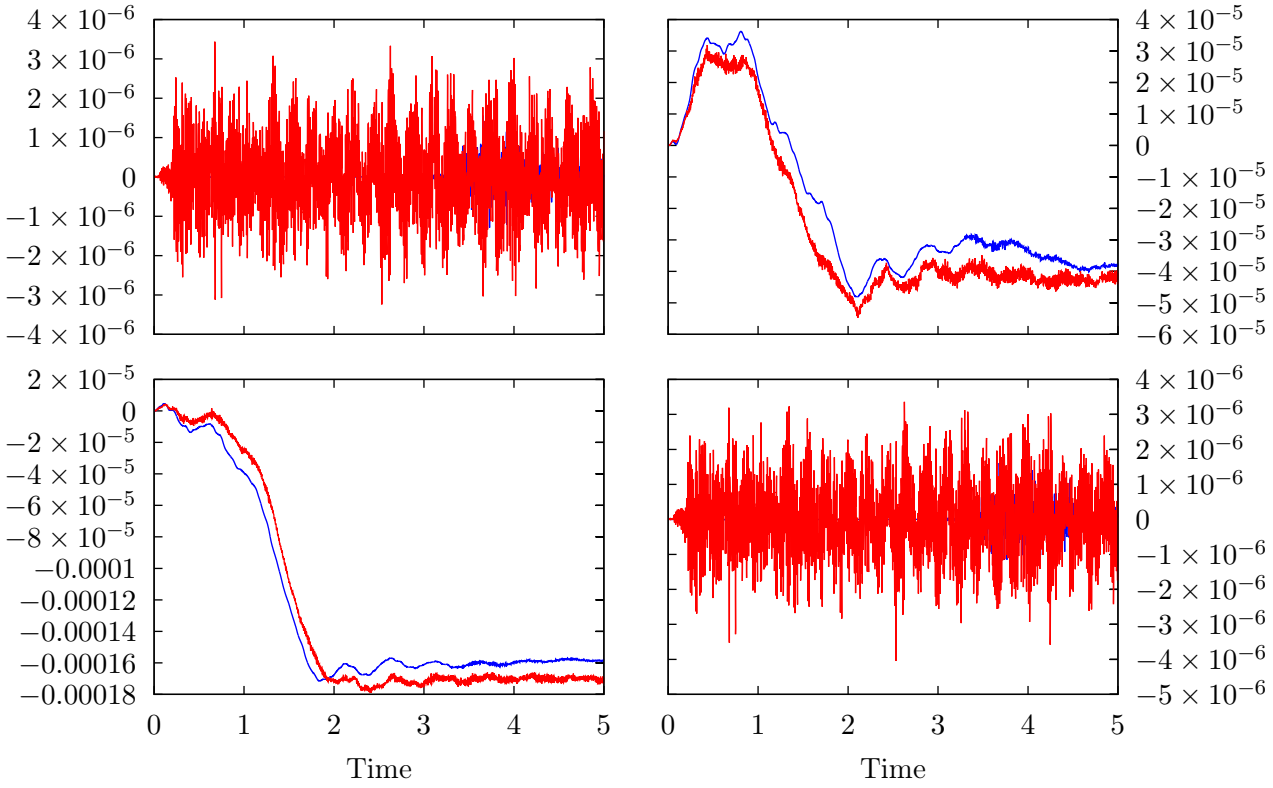

Figure 11: Example 2: z-component of displacement $u(P_i)$ using different levels r of refinements (blue: $r = 1$, red $r = 2$). We notice that the oscillations are of order 10^{-6} and thus numerical noise. Thus, the z-displacements are approximately zero, which was expected due to the symmetry of the configuration.

6.2.3 Performance studies

We first start with comments on the nonlinear and linear solvers followed by an analysis of the parallel performance. Figure 12 shows that the number of Newton iterations remains approximately constant at $5-10$ iterations throughout the computations. We note that in $3d$, the number of iterations increases slightly during h-refinement. The respective results of the 2d benchmarks are discussed in [53].

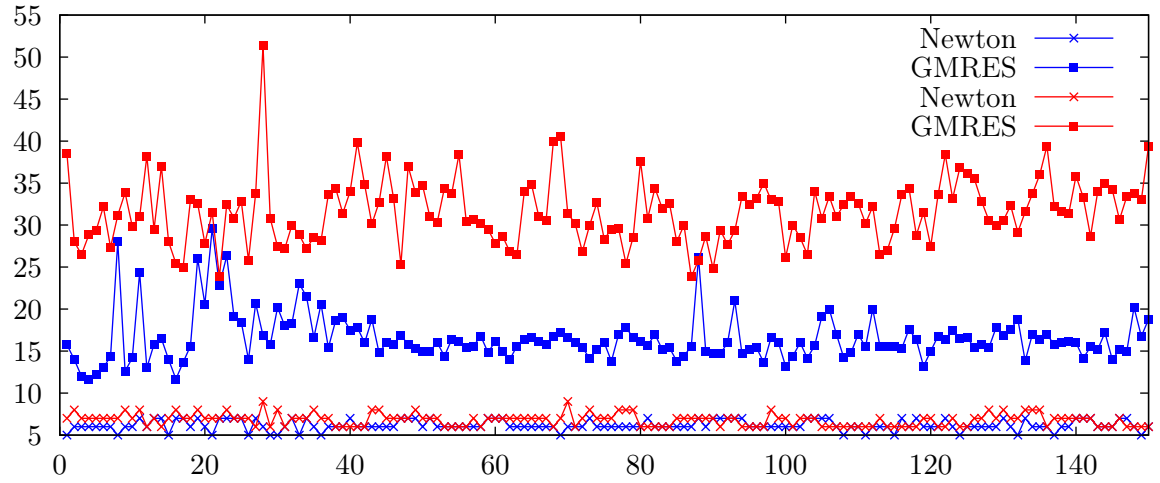

Figure 12: Number of iterations (Newton, average GMRES) required for the $3d$ test problem (blue: $r = 1$, red $r = 2$).

As we can see in Figure 13, the parallel performance in the $3d$ case yields similar results as in the case of the 2d benchmark problem FSI-2. First, these findings show that our code is dimension-independent and can be employed for 2d and 3d simulations. Second, the obtained scalability is again in the range of $\mathcal{O}(\sqrt{n})$. Due to the higher computational cost in $3d$, tests were done on 16 cores upwards. Specifically, for 16 cores, the total CPU time to solve the linear problem with $1.8 \cdot 10^6$ dofs at a single time step is 2605 seconds, i.e., approximately 43 minutes. On 256 cores, the computational cost decreases to 431 seconds, i.e., 7.2 minutes. Thus we achieve a reduction by 84% of the computational time.

7 Conclusions

In this work, we developed a framework for the parallel solution of monolithic fluid-structure interaction (FSI) problems. The FSI problems is formulated with the help of the arbitrary Lagrangian-Eulerian technique. To cope with large solid deformations, we adopted a nonlinear harmonic mesh motion model. The key goals have been on the development of approximate block-LDU preconditioners in which we used Schur complement arguments. The parallel implementation is based on a combination of different software packages, which are mainly joined in the C++ package deal.II. To date only very few other studies have been published with satisfying results for FSI problems using high performance parallel computing and showing satisfactory scalability. This has been an important motivation of this

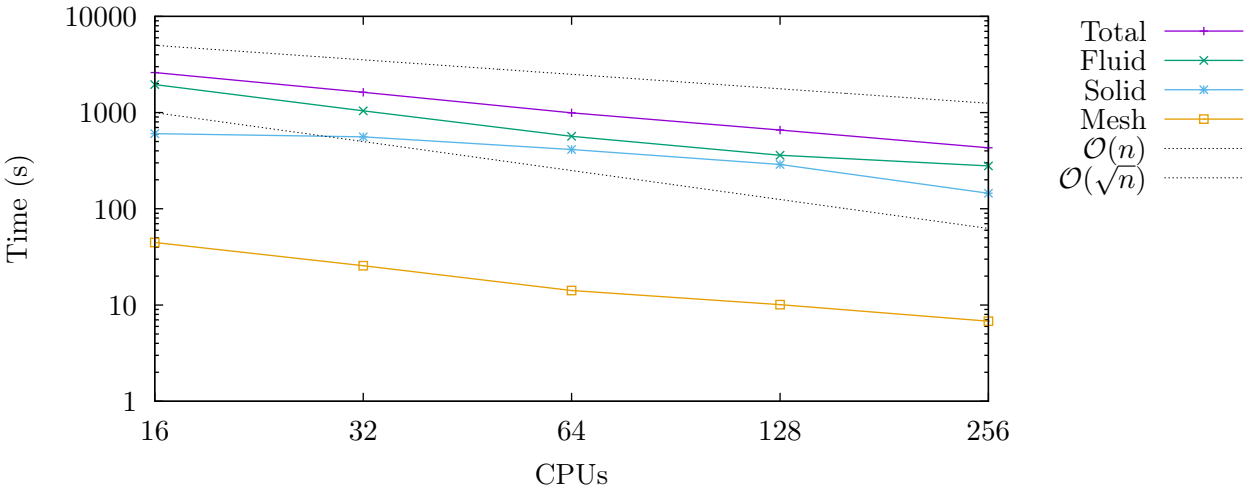

Figure 13: Example 2: Strong scalability using the preconditioned GMRES for approximately $1.8 \cdot 10^6$ dofs in $3d$. Average time given in seconds for the solution of one linear system.

work. Indeed, block-wise preconditioners perform pretty well in the serial case, but often lack perfect parallel scalability. This confirms the findings in [20, 33]. Additionally, this type of preconditioner may be applied to any coupled problem exhibiting a 3×3 block-structure similar to the FSI system, provided that solvers for the subproblems are available. A similar preconditioner is employed in [37], where additionally such a block-wise strategy is used as a smoother inside an AMG method. However, no scalability results have been reported therein. In our numerical tests, we have provided detailed studies for a challenging 2D benchmark problem and 3D test case. Both configurations are time-dependent and exhibit large solid deformations. In view of these aspects, the outcome of our results is more than satisfying.

8 Acknowledgments

This work has been supported by the Austrian Science Fund (FWF) grant No. P-29181 'Goal-Oriented Error Control for Phase-Field Fracture Coupled to Multiphysics Problems'.

References

[1] S. S. Antman. *Nonlinear problems of elasticity*, volume 107 of *Applied mathematical sciences*. Springer, New York, 1995.

[2] S. Badia, F. Nobile, and C. Vergara. Fluid-structure partitioned procedures based on Robin transmission conditions. *J. Comp. Phys.*, 227:7027–7051, 2008.

[3] S. Badia, A. Quaini, and A. Quarteroni. Modular vs. non-modular preconditioners for fluid-structure systems with large added-mass effect. *Comput. Methods Appl. Mech. Engrg.*, 197(49-50):4216–4232, 2008.

[4] S. Badia, Q. Quaini, and A. Quarteroni. Splitting methods based on algebraic factorization for fluid-structure interaction. *SIAM Journal on Scientific Computing*, 30(4):1778–1805, 2008.

[5] J. M. Ball. Convexity conditions and existence theorems in nonlinear elasticity. *Archive for Rational Mechanics and Analysis*, 63(4):337–403, 1976.

[6] D. Balzani, S. Deparis, S. Fausten, D. Forti, A. Heinlein, A. Klawonn, A. Quarteroni, O. Rheinbach, and J. Schröder. Numerical modeling of fluid-structure interaction in arteries with anisotropic polyconvex hyperelastic and anisotropic viscoelastic material models at finite strains. *Int. J. Numer. Methods Biomed. Engrg.*, 32(10), 2016.

[7] W. Bangerth, D. Davydov, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, and D. Wells. The `deal.II` library, version 8.4. *Journal of Numerical Mathematics*, 24, 2016.

[8] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, 33(4):24/1–24/27, 2007.

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

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

[11] Y. Bazilevs and K. Takizawa, editors. *Advances in Computational Fluid-Structure Interaction and Flow Simulation: New Methods and Challenging Computations*, Cham, Switzerland, 2016. Birkhäuser.

[12] Y. Bazilevs, K. Takizawa, and T. Tezduyar. *Computational Fluid-Structure Interaction: Methods and Applications*. John Wiley & Sons, Ltd, 2013.

[13] T. Bodnar, G. Galdi, and S. Necasova. *Fluid-Structure Interaction and Biomedical Applications*. Birkhaeuser, 2014.

[14] M. O. Bristeau, R. Glowinski, and J. Periaux. Numerical methods for the Navier-Stokes equations. *Comput. Phys. Rep.*, 6:73–187, 1987.

[15] H. J. Bungartz, M. Mehl, and M. Schäfer. *Fluid Structure Interaction II: Modelling, Simulation, Optimization*, volume 73. Springer Science & Business Media, 2010.

[16] H. J. Bungartz and M. Schäfer. *Fluid-Structure Interaction: Modelling, Simulation, Optimization*, volume 53 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin Heidelberg, 2006.

[17] P. Ciarlet. Mathematical elasticity, volume i: Three-dimensional elasticity. *Acta Applicandae Mathematica*, 18(2):190–195, 1990.

[18] D. Coutand and S. Shkoller. Motion of an elastic solid inside an incompressible viscous fluid. *Arch. Rational Mech. Anal.*, 176:25–102, 2005.

[19] D. Coutand and S. Shkoller. The interaction between quasilinear elastodynamics and the Navier-Stokes equations. *Arch. Rational Mech. Anal.*, 179:303–352, 2006.

[20] P. Crosetto, S. Deparis, G. Fourestey, and A. Quarteroni. Parallel algorithms for fluid-structure interaction problems in haemodynamics. *SIAM Journal on Scientific Computing*, 33(4):1598–1622, 2011.

[21] P. Crosetto, P. Reymond, S. Deparis, D. Kontaxakis, N. Stergiopulos, and A. Quarteroni. Fluid–structure interaction simulation of aortic blood flow. *Computers & Fluids*, 43(1):46 – 57, 2011. Symposium on High Accuracy Flow Simulations. Special Issue Dedicated to Prof. Michel DevilleSymposium on High Accuracy Flow Simulations.

[22] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels. Stability of a coupling technique for partitioned solvers in FSI applications. *Comput. Struct.*, pages 2224–2234, 2008.

[23] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, and J. Vierendeels. Performance of partitioned procedures in fluid-structure interaction. *Comput. Struct.*, pages 446–457, 2010.

[24] S. Deparis, D. Forti, P. Gervasio, and A. Quarteroni. Internodes: an accurate interpolation-based method for coupling the galerkin solutions of pdes on subdomains featuring non-conforming interfaces. *Computers & Fluids*, 141:22–41, 2016.

[25] S. Deparis, D. Forti, G. Grandperrin, and A. Quarteroni. Facsi: A block parallel preconditioner for fluid-structure interaction in hemodynamics. *Journal of Computational Physics*, 327:700–718, 2016.

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

[27] T. Dunne, T. Richter, and R. Rannacher. *Numerical simulation of fluid-structure interaction based on monolithic variational formulations*, pages 1–75. Comtemporary Challenges in Mathematical Fluid Mechanics. Springer, World Scientific, Singapore, 2010.

[28] L. C. Evans. *Partial differential equations*. American Mathematical Society, 2010.

[29] M. A. Fernández and M. Moubachir. A Newton method using exact Jacobians for solving fluid–structure coupling. *Computers & Structures*, 83(2–3):127 – 142, 2005.

[30] L. Formaggia, A. Quarteroni, and A. Veneziani. *Cardiovascular Mathematics: Modeling and simulation of the circulatory system*. Springer-Verlag, Italia, Milano, 2009.

[31] L. Formaggia, A. Quarteroni, and A. Veneziani. *Cardiovascular Mathematics: Modeling and simulation of the circulatory system*. Springer-Verlag, Italia, Milano, 2009.

[32] D. Forti, A. Quarteroni, and S. Deparis. A parallel algorithm for the solution of large-scale nonconforming fluid-structure interaction problems in hemodynamics. *J. Comput. Math.*, 35(3):363–380, 2017.

[33] D. Forti, A. Quarteroni, and S. Deparis. A parallel algorithm for the solution of large-scale nonconforming fluid-structure interaction problems in hemodynamics. *J. Comput. Math.*, 35(3):363–380, 2017.

[34] S. Frei, B. Holm, T. Richter, T. Wick, and H. Yang. *Fluid-structure interactions: Fluid-Structure Interaction: Modeling, Adaptive Discretisations and Solvers*. de Gruyter, 2017.

[35] C. Förster, W. A. Wall, and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1278 – 1293, 2007.

[36] G. Galdi and R. Rannacher. *Fundamental Trends in Fluid-Structure Interaction*. World Scientific, 2010.

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

[38] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala. ML 5.0 smoothed aggregation user's guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.

[39] C. Grandemont. Existence of weak solutions for the unsteady interaction of a viscous fluid with an elastic plate. *SIAM J. Math. Anal.*, 40(2):716–737, 2008.

[40] C. Grandmont. Existence for a three-dimensional steady state fluid-structure interaction problem. *Journal of Mathematical Fluid Mechanics*, 4(1):76–94, 2002.

[41] M. Heil. An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(1-2):1 – 23, 2004.

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

[43] M. Heroux, R. Bartlett, V. H. R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, and A. Williams. An Overview of Trilinos. Technical Report SAND2003-2927, Sandia National Laboratories, 2003.

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

[45] J. G. Heywood, R. Rannacher, and S. Turek. Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations. *International Journal of Numerical Methods in Fluids*, 22:325–352, 1996.

[46] G. A. Holzapfel. *Nonlinear solid mechanics; a continuum approach for engineering.* John Wiley & Sons, Ltd, reprint. edition, 2010.

[47] J. Hron and S. Turek. *A monolithic FEM/Multigrid solver for ALE formulation of fluid structure with application in biomechanics*, volume 53, pages 146–170. Springer-Verlag, 2006.

[48] J. Hron and S. Turek. *Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow*, pages 371–385. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[49] M.-C. Hsu and Y. Bazilevs. Fluid–structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, 50(6):821–833, 2012.

[50] T. J. R. Hughes, W. K. Liu, and T. Zimmermann. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Comput. Methods Appl. Mech. Engrg.*, 29:329–349, 1981.

[51] M. Ignatova, I. Kukavica, I. Lasiecka, and A. Tuffaha. On well-posedness for a free boundary fluid-structure model. *J. Math. Phys.*, 53(11):115624, November 2012.

[52] D. Jodlbauer. Robust preconditioners for fluid-structure-interaction problems. Master's thesis, Johannes Kepler University Linz, Institute of Computational Mathematics, 2016. Available on https://www.numa.uni-linz.ac.at/Teaching/Diplom/.

[53] D. Jodlbauer and T. Wick. A monolithic FSI solver applied to the FSI 1,2,3 benchmarks. In S. Frei, B. Holm, T. Richter, T. Wick, and H. Yang, editors, *Fluid-Stucture Interaction: Modeling, Adaptive Discretizations and Solvers*, volume 20 of *Radon Series on Computational and Applied Mathematics*, chapter 6, pages 193–234. de Gruyter, Berlin, Boston, 2017.

[54] G. Karypis and V. Kumar. A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1999.

[55] F. Kong and X.-C. Cai. Scalability study of an implicit solver for coupled fluid-structure interaction problems on unstructured meshes in 3d. *The International Journal of High Performance Computing Applications*, 2016. DOI: 10.1177/1094342016646437.

[56] U. Langer and H. Yang. Numerical simulation of fluid-structure interaction problems with hyperelastic models I: A partitioned approach. *Journal of Computational and Applied Mathematics*, 276:47–61, 2015.

[57] U. Langer and H. Yang. Partitioned solution algorithms for fluid-structure interaction problems with hyperelastic models. *Journal of Computational and Applied Mathematics*, 276(0):47 – 61, 2015.

[58] U. Langer and H. Yang. Robust and efficient monolithic fluid-structure-interaction solvers. *International Journal for Numerical Methods in Engineering*, 108(4):303–325, 2016.

[59] U. Langer and H. Yang. Numerical simulation of fluid-structure interaction problems with hyperelastic models: A monolithic approach. *Mathematics and Computers in Simulation*, 145:186–208, 2018.

[60] M. Mayr, T. Klöppel, W. Wall, and M. Gee. A temporal consistent monolithic approach to fluid-structure interaction enabling single field predictors. *SIAM J. Sci. Comput.*, 37(1):B30–B59, 2015.

[61] D. Mok and W. Wall. *Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures.* Trends in Computational Structural Mechanics. CIMNE, Barcelona, 2001.

[62] R. Muddle, M. Mihajlović, and M. Heil. An efficient preconditioner for monolithically-coupled large-displacement fluid-structure interaction problems with pseudo-solid mesh updates. *J. Comput. Phys.*, 231(21):7315–7334, 2012.

[63] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems - part ii: energy transfer analysis and three-dimensional applications. *Comput. Methods Appl. Mech. Engrg.*, 190:3147–3170, 2001.

[64] M. Razzaq, H. Damanik, J. Hron, A. Ouazzi, and S. Turek. FEM multigrid techniques for fluid-structure interaction with application to hemodynamics. *Appl. Numer. Math.*, 62(9):1156–1170, 2012.

[65] T. Richter. A monolithic geometric multigrid solver for fluid-structure interactions in ale formulation. *International Journal for Numerical Methods in Engineering*, 104(5):372–390, 2015.

[66] T. Richter. *Fluid-structure interactions: models, analysis, and finite elements.* Springer, 2017.

[67] 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*, pages 377–400, Contributions in Mathematical and Computational Science, 2015.

[68] N. D. D. Santos, J.-F. Gerbeau, and J. Bourgat. A partitioned fluid-structure algorithm for elastic thin valves with contact. *Comp. Methods Appl. Mech. Engrg.*, 197(19-20):1750–1761, 2008.

[69] K. Stein, T. Tezduyar, and R. Benney. Mesh moving techniques for fluid-structure interactions with large displacements. *J. Appl. Mech.*, 70:58–63, 2003.

[70] R. Temam. *Navier-Stokes equations; theory and numerical analysis.* AMS Chelsea publishing. American Math. Soc., Providence, RI, reprint. with corr. edition, 2001.

[71] E. H. van Brummelen, K. G. van der Zee, and R. de Borst. Space/time multigrid for a fluid–structure-interaction problem. *Appl. Numer. Math.*, 58(12):1951–1971, 2008.

[72] J. Vierendeels, K. Dumont, and P. Verdonck. A partitioned strongly coupled fluid-structure interaction method to model heart valve dynamics. *J. Comp. Appl. Math.*, 2008.

[73] W. Wall, S. Genkinger, and E. Ramm. A strong coupling partitioned approach for fluid-structure interaction with free surfaces. *Computers and Fluids*, 36:169–183, 2007.

[74] J. Weickert. *Navier-stokes Equations as a Differential Algebraic System.* Preprint-Reihe des Chemnitzer SFB 393. Techn. Univ., 1996.

[75] T. Wick. *Adaptive Finite Element Simulation of Fluid-Structure Interaction with Application to Heart-Valve Dynamics.* PhD thesis, University of Heidelberg, 2011.

[76] T. Wick. Fluid-structure interactions using different mesh motion techniques. *Computers & Structures*, 89(13–14):1456 – 1467, 2011.

[77] T. Wick. *Variational-Monolithic ALE Fluid-Structure Interaction: Comparison of Computational Cost and Mesh Regularity Using Different Mesh Motion Techniques*, pages 261–275. Springer International Publishing, Cham, 2017.

[78] W. L. Wood. *Practical Time-stepping Schemes.* Clarendon Press, Gloucestershire, 1990.

[79] Y. Wu and X. Cai. A fully implicit domain decomposition based ale framework for three-dimensional fluid-structure interaction with application in blood flow computation. *J. Comput. Physics*, 258:524–537, 2014.