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

D. Jodlbauer, U. Langer, T. Wick

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

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

AG Wissenschaftliches Rechnen
Institut für Angewandte Mathematik
Leibniz Universität Hannover
Welfengarten 1, 30167 Hannover, Germany

This is the peer reviewed version of the following article: Jodlbauer, D, Langer, U, Wick, T. Parallel block-preconditioned monolithic solvers for fluid-structure interaction problems. Int J Numer Methods Eng. 2019; 117: 623– 643, which has been published in final form at `https://doi.org/10.1002/nme.5970`. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Use of Self-Archived Versions. This article may not be enhanced, enriched or otherwise transformed into a derivative work, without express permission from Wiley or by statutory rights under applicable legislation. Copyright notices must not be removed, obscured or modified. The article must be linked to Wiley's version of record on Wiley Online Library and any embedding, framing or otherwise making available the article or pages thereof by third parties from platforms, services and websites other than Wiley Online Library must be prohibited.

International Journal for Numerical Methods in Engineering
Int. J. Numer. Meth. Engng. 0000; **00**:1–23
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld

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

D. Jodlbauer[1], U. Langer[2], 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*

SUMMARY

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 growth 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. Copyright © 0000 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: fluid-structure interaction; monolithic formulation; parallel solvers; physics-based block preconditioners

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., [1, 2]. 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., [3] and [4], respectively. More FSI applications are compiled in various books [5, 6, 7, 8, 9, 10, 11, 12].

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., [13, 2, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24] 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

*Correspondence to: E-mail: thomas.wick@ifam.uni-hannover.de

Copyright © 0000 John Wiley & Sons, Ltd.

with the high complexity, loss of robustness (specifically for further extensions of FSI; see, e.g., [25] or certain other chapters in [12]), instabilities due to so-called added-mass effects [26, 14, 27], 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., [28, 29, 5, 30, 31, 32, 33, 8, 34, 35, 36]. 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 [37, 38, 36] and [33, 39, 35], respectively. Likewise domain decomposition methods can be exploited [40, 41, 42]. 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 [28, 43, 31, 44, 45, 46, 42, 47, 34, 35] 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 movement (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 [40], 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 [41]; see also [48]. This Newton-Krylov-Schwarz FSI solver shows a very good parallel performance, 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 [45]. Strong and weak scalability studies were presented for a moderate number of cores. In the very recent publications [49] and [50], 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 [51]. In [52], a parallel framework for the partitioned FSI solution was presented.

In this paper, we follow basically the developments of [34, 35, 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 [29, 5] with large solid deformations. In 3d, we adopt another benchmark-like configuration [36]. 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 modeled 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 [54, 55]. 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 [56]. This leads to the following system of equations [37, 57, 56]:

Formulation 2.1 (Strong form of FSI equations)**.** Find $\hat v_f : I \times \hat\Omega_f \to \mathbb{R}^d, \hat p_f : I \times \hat\Omega_f \to \mathbb{R}, \hat u_f : I \times \hat\Omega_f \to \mathbb{R}^d, \hat v_s : I \times \hat\Omega_f \to \mathbb{R}^d, \hat u_s : I \times \hat\Omega_s \to \mathbb{R}^d$:

$$\begin{aligned}
\hat J\hat\rho_f\hat\partial_t\hat v_f + \hat J\hat\rho_f\hat\nabla\hat v_f\hat F^{-1}\cdot(\hat v_f - \partial_t\hat{\mathcal{A}}) - \mathrm{div}_R(\hat J\hat\sigma_f\hat F^{-T}) &= 0\\
\hat J\ \mathrm{tr}(\hat\nabla\hat v_f\hat F^{-1}) &= 0\\[1em]
\hat\rho_s\partial_t\hat v_s - \mathrm{div}_R(\hat F\hat\Sigma) &= 0\\
(\partial_t\hat u_s - \hat v_f) &= 0\\[1em]
-\mathrm{div}_R(\frac{1}{\hat J}\nabla\hat u_f) &= 0.
\end{aligned} \qquad (1)$$

Here, $\hat\nabla(\cdot)$ and $\mathrm{div}_R(\cdot)$ denote the gradient and the divergence operators in the reference domain, respectively. 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 [58] is adopted since we are interested in modeling large solid deformations.

The interface and boundary conditions are given by the equations

$$\begin{aligned}
\hat J\hat\sigma_f\hat F^{-T}\hat n_f + \hat F\hat\Sigma_s &= 0 \text{ on } I\times\hat\Gamma_I & \hat v_f &= 0 \text{ on } I\times\hat\Gamma_{top}\cup I\times\hat\Gamma_{bottom}\cup I\times\hat\Gamma_c & (2)\\
\hat v_f - \hat v_s &= 0 \text{ on } I\times\hat\Gamma_I & \hat v_f &= g \text{ on } I\times\hat\Gamma_{in} & (3)\\
\hat u_f - \hat u_s &= 0 \text{ on } I\times\hat\Gamma_I & \hat u_f &= 0 \text{ on } I\times\{\hat\Gamma_{top}\cup\hat\Gamma_{bottom}\cup\hat\Gamma_{in}\cup\hat\Gamma_{out}\cup\hat\Gamma_c\} & (4)\\
&& \hat u_s &= 0 \text{ on } I\times\hat\Gamma_{cf} & (5)\\
&& \hat v_s &= 0 \text{ on } I\times\hat\Gamma_{cf}. & (6)
\end{aligned}$$

We notice that the dynamic interface condition (2) only holds in a distributional sense. For details, we refer to the corresponding literature of trace theorems, e.g., [59, 60].

Furthermore,

$$\hat J(-\hat p_f\hat I + \hat\rho_f\hat\nu_f\hat\nabla\hat v_f\hat F^{-1})\hat F^{-T} = 0 \text{ on } I\times\hat\Gamma_{out}. \qquad (7)$$

Copyright © 0000 John Wiley & Sons, Ltd.

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) is given [61], 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 follows:

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

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

$$\hat{J} = \det(\hat{F}).$$

The solid stress tensor is chosen according to the Saint Venant Kirchhoff material law (STVK) as

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

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

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

Copyright © 0000 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

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., [62, 63]. 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.

Remark 3.1. In the definition of the spaces V_u, V_v, V_u^s, V_v^s, we observe that these spaces actually yield global H^1 spaces due to $\hat{u}_s = \hat{u}_f$ and $\hat{v}_s = \hat{v}_f$ on the interface $\hat{\Gamma}_I$, respectively.

Remark 3.2. As described in detail in [64, 11], the global extension of the function spaces yields that the kinematic interface conditions are prescribed in these function spaces as it is usually the case for Dirichlet-like conditions. The continuity of the dynamic interface condition, which is of Neumann-type, is fulfilled in an exact variational fashion and cancel in a monolithic modeling. Thus, the condition

$$\langle \hat{J}\hat{\sigma}_f \hat{F}^{-T}\hat{n}_f, \hat{\varphi}_f^v\rangle_{\hat{\Gamma}_i} + \langle \hat{F}\hat{\Sigma}_s\hat{n}_s, \hat{\varphi}_s^v\rangle_{\hat{\Gamma}_i} = 0 \quad \forall \hat{\varphi}_f^v \in H^1(\hat{\Omega}_f),\ \forall \hat{\varphi}_s^v \in H^1(\hat{\Omega}_s), \quad \hat{\varphi}_f^v = \hat{\varphi}_s^v \quad \text{on } \hat{\Gamma}_i, \tag{9}$$

is implicitly contained in the following system. Further literature and usage of this coupling condition is given in [37, 65, 66, 67, 68, 64, 11, 25].

With these preliminaries, we obtain the following FSI system:

Formulation 3.3 (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 $t \in I$:

$$\begin{aligned}
\Big(\hat{J}\hat{\rho}_f\hat{\partial}_t\hat{v}_f\ ,\ \hat{\varphi}_f^v\Big)_{\hat{\Omega}_f} + \Big(\hat{J}\hat{\rho}_f\hat{\nabla}\hat{v}_f\hat{F}^{-1}\cdot(\hat{v}_f - \hat{w})\ ,\ \varphi_f^v\Big)_{\hat{\Omega}_f} + \Big(\hat{J}\hat{F}^{-T}\hat{\sigma}_f\ ,\ \hat{\nabla}\hat{\varphi}_f^v\Big)_{\hat{\Omega}_f}& \\
-\Big\langle \hat{\rho}_f\hat{\nu}_f\hat{J}\hat{F}^{-T}\hat{\nabla}\hat{v}_f^T\hat{F}^{-T}\cdot\hat{n}_f\ ,\ \hat{\varphi}_f^v\Big\rangle_{\hat{\Gamma}_{out}} &= 0 \\
\Big(\hat{J}\ \mathrm{tr}(\hat{\nabla}\hat{v}_f\hat{F}^{-1})\ ,\ \hat{\varphi}_f^p\Big)_{\hat{\Omega}_f} &= 0 \\
\\
\Big(\hat{\rho}_s\hat{\partial}_t\hat{v}_s\ ,\ \hat{\varphi}_s^v\Big)_{\hat{\Omega}_s} + \Big(\hat{F}\hat{\Sigma}_s\ ,\ \hat{\nabla}\hat{\varphi}_s^v\Big)_{\hat{\Omega}_s} &= 0 \\
\Big(\hat{\partial}_t\hat{u}_s - \hat{v}_s\ ,\ \hat{\varphi}_s^u\Big)_{\hat{\Omega}_s} &= 0 \\
\\
\Big(\frac{1}{\hat{J}}\hat{\nabla}\hat{u}_f\ ,\ \hat{\nabla}\hat{\varphi}_f^u\Big)_{\hat{\Omega}_f} &= 0
\end{aligned} \tag{10}$$

for all test-functions $\Big((\hat{\varphi}_s^u, \hat{\varphi}_f^u), (\hat{\varphi}_s^v, \hat{\varphi}_f^v), \hat{\varphi}_f^p\Big)$ 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 3.4. We notice that we split all equations into the respective fluid and solid domains in order to highlight the different submodels. However, in a (fully) monolithic formulation, the equations are later summed up yielding compact equations defined on the entire domain; see for instance [69] for such a compact notation including the corresponding implementation.

Copyright © 0000 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

Remark 3.5 (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 [70]), which can be analyzed in the framework of monotone operators (chapter 9 of [70]) 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 [59, 71, 63, 62]. Of course a complete well-posedness analysis is not available yet in the literature.

Remark 3.6 (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 [72, 73, 74, 75].

3.2. Discretization in Time

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

Definition 3.7 (One-Step-Theta Scheme)**.** Given a differential equation $a(u)\partial_t u + A(u) = 0$, the one-step-theta scheme reads as follows:

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

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) [76], $\theta = 0.5$ (Crank-Nicolson), $\theta = 0.5 + \Delta t$ (Shifted Crank-Nicolson) [77], $\theta = 1$ (Implicit Euler), or the Fractional-Step-Theta scheme [78].

The explicit Euler method would require too small time steps in order to be stable, whereas the Implicit Euler method dampens the oscillations too much, when relatively[†] large time step sizes are adopted. 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 [64, 79, 80]. In [81] 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. These previous studies have shown as well (in addition to [82][Section 3.2] for pure Navier-Stokes flow) that in general the Crank-Nicolson scheme is less robust in long-term simulations. Moreover, post-processing for the pressure variable to obtain $2nd$ order accuracy in time in the temporal grid points for functionals of interest containing pressure information (such as drag and lift) is necessary. In this respect, an optimal time-stepping scheme is the Fractional-Step-Theta scheme, which has been investigated in great detail with respect to temporal robustness, accuracy and adaptivity in [80].

Remark 3.8 (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. [83].

3.3. Linearization

The nonlinearities within our FSI system are treated by Newton's method. Details are presented in [84].

For the differentiation of the whole FSI operator (12), 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.

[†] 'Large' has to be understood in relation to other material and model parameters and their respective characteristic time scales.

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

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:

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

Theorem 3.9 (FSI-related Derivatives). *It holds:*

$$\begin{aligned} \partial\hat{F} &= \hat{\nabla}\delta\hat{u} \\ \partial\hat{J} &= \hat{J}tr(\hat{F}^{-1}\hat{\nabla}\delta\hat{u}) \\ \partial\hat{F}^{-1} &= -\hat{F}^{-1}\hat{\nabla}\delta\hat{u}\hat{F}^{-1} \\ \partial\hat{F}^{-T} &= (\partial\hat{F}^{-1})^T \\ \partial tr(E) &= tr(\partial E) \end{aligned} \tag{11}$$

Proof

Unless trivial, see for example [85, 86]. □

The semi-linear form $A(U)(\Phi)$ and linear form $F(\Phi)$ are given as the sum of all equations in (10).

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

and

$$\begin{aligned} F(\Phi) = &\left(\hat{\rho}_s\hat{v}_s^{n-1}\,,\,\hat{\varphi}_s^v\right)_{\hat{\Omega}_s} - \Delta t\,(1-\theta)\left(\hat{F}^{n-1}\hat{\Sigma}_s^{n-1}\,,\,\hat{\nabla}\hat{\varphi}_s^v\right)_{\hat{\Omega}_s} \\ &+ \left(\hat{u}_s^{n-1}\,,\hat{\varphi}_s^u\right)_{\hat{\Omega}_s} + \Delta t\,(1-\theta)\ \left(\hat{v}_s^{n-1}\,,\,\hat{\varphi}_s^u\right)_{\hat{\Omega}_s} \\ &- \left(\hat{J}\hat{\rho}_f\hat{F}^{-1}\hat{\nabla}\hat{v}_f\cdot\hat{u}_f^{n-1}\,,\,\hat{\varphi}_f^v\right)_{\hat{\Omega}_f} \\ &- \Delta t\,(1-\theta)\left[\left(\hat{J}\hat{\rho}_f\hat{F}^{-1}\hat{\nabla}\hat{v}_f\cdot\hat{v}_f\,,\,\hat{\varphi}_f^v\right)_{\hat{\Omega}_f} + \left(\hat{J}\hat{\sigma}_f^v\hat{F}^{-T}\,,\,\hat{\nabla}\hat{\varphi}_f^v\right)_{\hat{\Omega}_f}\right]^{n-1} \\ &+ \Delta t\,(1-\theta)\left[\left\langle\hat{\rho}_f\hat{\nu}_f\hat{J}\hat{F}^{-T}\hat{\nabla}\hat{v}_f^T\hat{F}^{-T}\cdot\hat{n}_f\,,\hat{\varphi}_f^v\right\rangle_{\hat{\Gamma}_{out}}\right]^{n-1} \end{aligned} \tag{13}$$

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

$$\begin{aligned} V_h^s &:= \left\{ v \in H^1(\hat{\Omega}_s)^d : v\big|_T \in Q(k) \quad \forall\, T \subset \hat{\Omega}_s \right\}, \\ V_h^f &:= \left\{ v \in H^1(\hat{\Omega}_f)^d : v\big|_T \in Q(k) \quad \forall\, T \subset \hat{\Omega}_f \right\}, \\ L_h^f &:= \left\{ v \in L^2(\hat{\Omega}_f) : v\big|_T \in P(k-1)\; \forall\, T \subset \hat{\Omega}_f \right\} \end{aligned} \tag{14}$$

with the nodal basis functions

$$\begin{aligned} V_h^s &= span\{\varphi_{s,h}^v[j],\ j = 1, \dots, N_{v_s}\} = span\{\varphi_{s,h}^u[j],\ j = 1, \dots, N_{u_s}\}, \\ V_h^f &= span\{\varphi_{f,h}^v[j],\ j = 1, \dots, N_{v_f}\} = span\{\varphi_{f,h}^u[j],\ j = 1, \dots, N_{u_f}\}, \\ L_h^f &= span\{\varphi_{f,h}^p[j],\ j = 1, \dots, N_{p_f}\}. \end{aligned} \tag{15}$$

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

$$A_h := \begin{bmatrix} \mathcal{M} & \mathcal{C}_{ms} & 0 \\ \mathcal{C}_{sm} & \mathcal{S} & \mathcal{C}_{sf} \\ \mathcal{C}_{fm} & \mathcal{C}_{fs} & \mathcal{F} \end{bmatrix}, \tag{16}$$

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:

$$\begin{bmatrix} I & 0 & 0 \\ \mathcal{C}_{sm}\mathcal{M}^{-1} & I & 0 \\ \mathcal{C}_{fm}\mathcal{M}^{-1} & \tilde{\mathcal{C}}_{fs}\tilde{\mathcal{S}}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathcal{M} & 0 & 0 \\ 0 & \tilde{\mathcal{S}} & 0 \\ 0 & 0 & \mathcal{F} - \tilde{\mathcal{C}}_{fs}\tilde{\mathcal{S}}^{-1}\mathcal{C}_{sf} \end{bmatrix} \begin{bmatrix} I & \mathcal{M}^{-1}\mathcal{C}_{ms} & 0 \\ 0 & I & \tilde{\mathcal{S}}^{-1}\mathcal{C}_{sf} \\ 0 & 0 & I \end{bmatrix}, \tag{17}$$

where $\tilde{\mathcal{S}} = \mathcal{S} - \mathcal{C}_{sm}\mathcal{M}^{-1}\mathcal{C}_{ms}$ and $\tilde{\mathcal{C}}_{fs} = \mathcal{C}_{fs} - \mathcal{C}_{fm}\mathcal{M}^{-1}\mathcal{C}_{ms}$.

Using this LDU-factorization directly is obviously not very efficient since it involves the computation and storage of too many inverses 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

$$\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \mathcal{C}_{fm}\mathcal{M}^{-1} & \tilde{\mathcal{C}}_{fs}\mathcal{S}^{-1} & 1 \end{bmatrix} \begin{bmatrix} \mathcal{M} & 0 & 0 \\ 0 & \mathcal{S} & 0 \\ 0 & 0 & \mathcal{X} \end{bmatrix} \begin{bmatrix} I & \mathcal{M}^{-1}\mathcal{C}_{ms} & 0 \\ 0 & I & \mathcal{S}^{-1}\mathcal{C}_{sf} \\ 0 & 0 & I \end{bmatrix} \tag{18}$$

with $\mathcal{X} = \mathcal{F} - \tilde{\mathcal{C}}_{fs}\mathcal{S}^{-1}\mathcal{C}_{sf}$.

In order to apply solvers like sparse-LU or AMG methods, which require explicit 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 [34]. 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 $P^{-1}r$.

```
Solve $x_m = \mathcal{M}^{-1}r_m$
Solve $x_s = \mathcal{S}^{-1}r_s$
Solve $x_f = \mathcal{F}^{-1}(r_f - \mathcal{C}_{fm}x_m - \mathcal{C}_{fs}x_s)$
Update $x_s = x_s - \mathcal{S}^{-1}\mathcal{C}_{sf}x_f$
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 [87, 88] 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

$$\begin{bmatrix} \hat{\rho}_s M + \Delta t^2\, \theta^2 K_{vu} & 0 \\ -\Delta t\, \theta\, M & M \end{bmatrix} \begin{bmatrix} x_{v_s} \\ x_{u_s} \end{bmatrix} = \begin{bmatrix} r_{v_s} - \Delta t\, \theta K_{vu} M^{-1} r_{u_s} \\ r_{u_s} \end{bmatrix} \tag{19}$$

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 [89, 90], with the Trilinos package [87] for linear algebra operations and its multi-level package ML [88]. Partitioning of the mesh is done using ParMETIS [91].

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 (see Figure 1)
- split: each core owns either parts of the fluid or the solid domain, but not both (see Figure 2)
- default: no distinction is made between fluid and solid subdomains

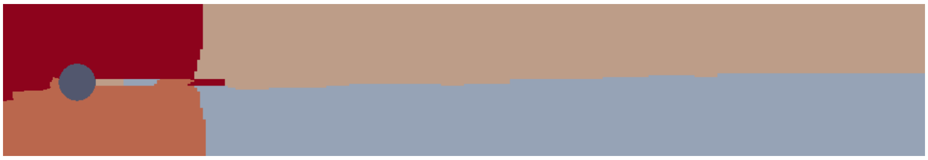

Figure 1. Shared configuration using four subdomains.

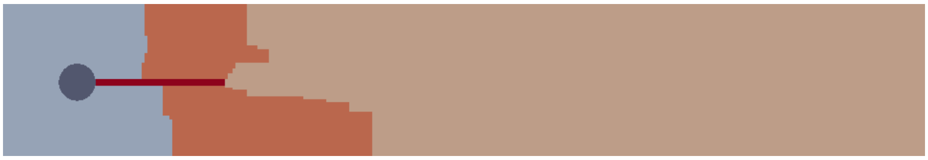

Figure 2. Split configuration using four subdomains. Load balancing issues are obvious.

Copyright © 0000 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

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.

5.2. *Interface Coupling*

We use global variables for the displacement, velocity and pressure (see also [69]). 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, the continuity of normal stresses, is fulfilled in a weak sense as defined in (9).

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 [29], 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 [36]. 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.

Remark 6.1. We notice that numerical results regarding robustness, h and Δt-dependence have been presented in [92, 53] for sequential computations. Consequently, in this work, we focus on the parallel scalability.

6.1. *Computational Resources and Solver Adjustments*

All tests regarding parallelization were done on the distributed memory cluster Radon1[‡] 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 less computational 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 $\|r_k\|_\infty > 0.1\,\|r_{k-1}\|_\infty$. 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.

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

6.2. *Example 1: FSI-2 Benchmark*

6.2.1. 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.

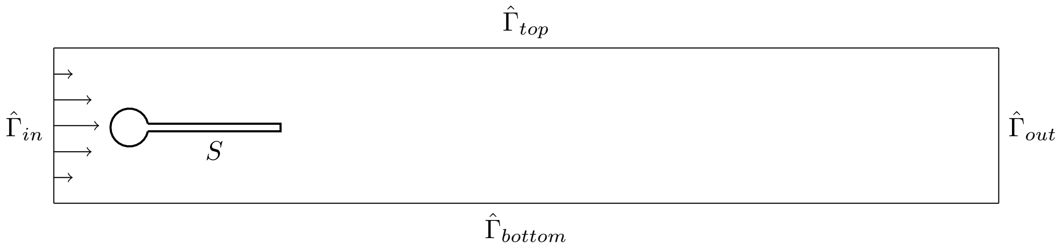

Figure 3. Geometry of the FSI-2 benchmark [29]: Elastic beam immersed in a flow around a cylinder.

In the following, all parameters and geometry data are given in SI units as originally listed in [29]. 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	$2.5m$
channel height	$0.41m$
cylinder center	$(0.2m, 0.2m)$
cylinder radius	$0.05m$
beam thickness	$0.02m$
beam length	$0.35m$
reference point	$(0.6m, 0.2m)$

Quantity	Value
inflow velocity $\bar{v}$	$1.0m/s$
density ρ_f	$10^3kg/m^3$
viscosity ν_f	$10^{-3}m^2/s$

Quantity	Value
solid density $\hat{\rho}_s$	$10^4kg/m^3$
Lamé λ	$2 \cdot 10^6 Pa$
Lamé μ	$0.5 \cdot 10^6 Pa$
Poisson ratio ν	0.4

We refer the reader to FeatFlow[§] or [29] for a more detailed description of FSI-2 benchmark.

The number of dofs range from $15k$ $(r = 1)$ to $240k$ $(r = 3)$ using $Q(2) - Q(2) - P(1)$ elements. The dof counts for the respective $Q(1) - Q(1) - P(0)$ elements are exactly shifted by one refinement level, i.e. $15k$ $(r = 2)$ to $240k$ $(r = 4)$. We notice that the $Q(1) - Q(1) - P(0)$ discretization may be unstable in the pressure variable (when the velocity approximations are still satisfactory), and the inf-sup condition is violated or only fulfilled with a h-dependent constant; see e.g., [93][Sections 3.3 and 3.4] or [94][Section 6.2.2].

6.2.2. 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 [29, 5, 56].

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:

$$(F_D, F_L) = \int_S \hat{\sigma}_f \hat{n}_f ds, \tag{20}$$

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 for different elements, refinements and time step sizes are shown in Figures 4 – 7. All configurations yield similar results, but a slight shift in time was observed. To allow for better comparison of frequency and amplitude, the plots were shifted accordingly. As previously mentioned, the pressure in the $Q(1) - Q(1) - P(0)$ discretization is unstable. Indeed, our results slightly differ from [29, 5, 56]. However, we still obtain surprisingly satisfying findings in the four functionals of interest.

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

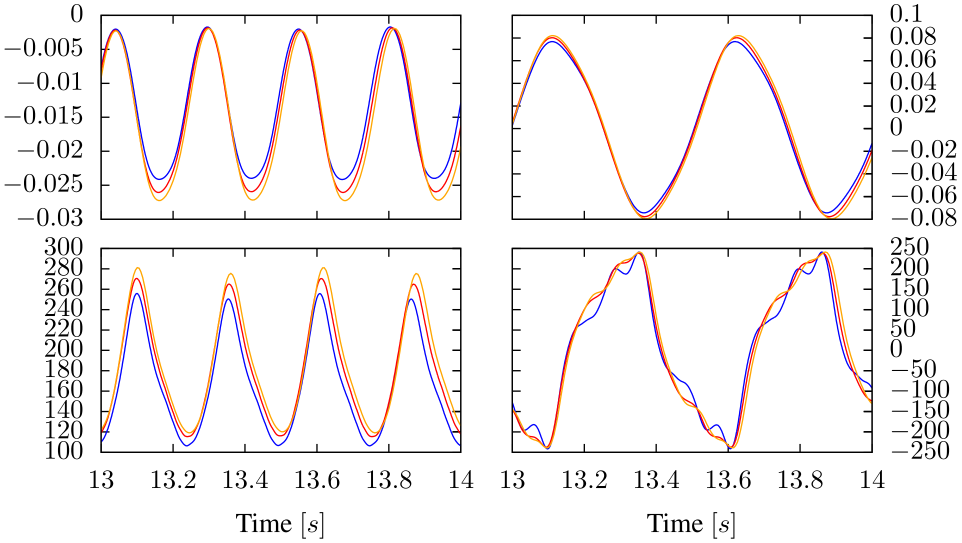

Figure 4. FSI-2 benchmark: Drag, lift (bottom row) and x/y-displacements (top row) using different levels r of refinements for $Q(1) - Q(1) - P(0)$ (blue: $r = 2$, red: $r = 3$, orange: $r = 4$). Time was shifted by $0.12s$ for $r = 3$ and $0.14s$ for $r = 4$ to allow better comparisons of frequencies and amplitudes of the oscillations. Indeed, a correct code should yield not only the convergence in the four functional values, but it also should yield the same frequencies and amplitudes under spatial and temporal refinement.

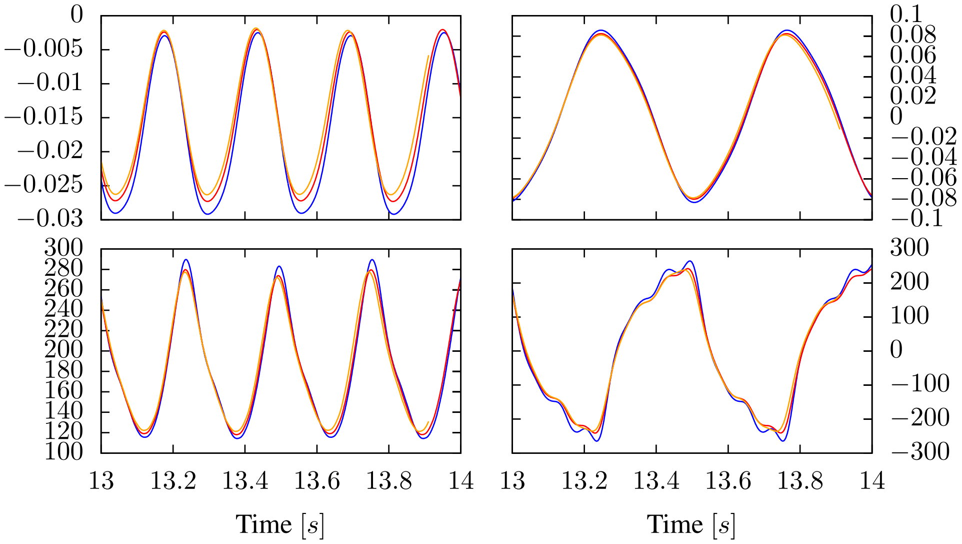

Figure 5. FSI-2 benchmark: Drag, lift (bottom row) and x/y-displacements (top row) using different levels r of refinements for $Q(2) - Q(2) - P(1)$ (blue: $r = 1$, red: $r = 2$, orange: $r = 3$). Time was shifted by $0.035s$ for $r = 2$ and $0.003s$ for $r = 3$ to allow better comparison of frequencies and amplitudes of the oscillations.

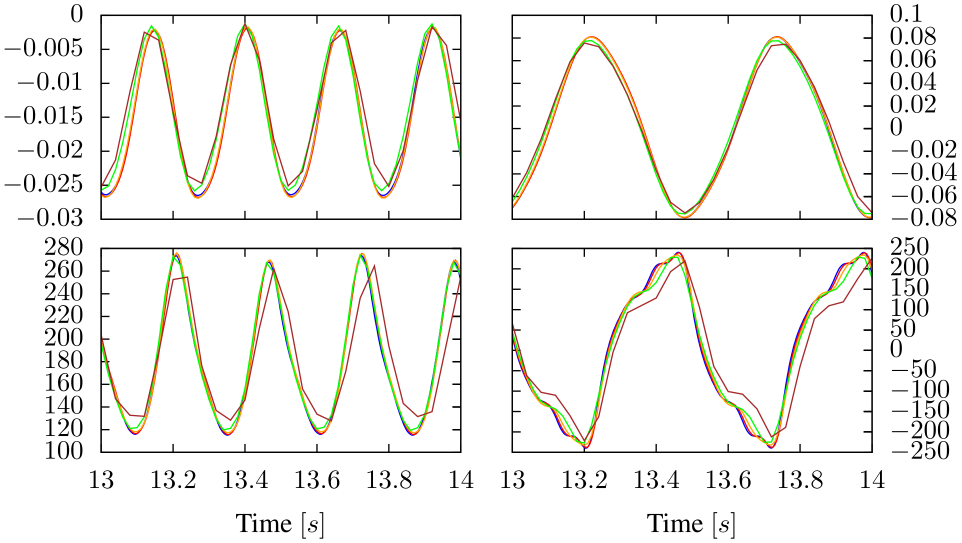

Figure 6. FSI-2 benchmark: Drag, lift (bottom row) and x/y-displacements (top row) with $Q(1) - Q(1) - P(0)$ elements on a 3 times refined grid. Time was shifted to allow better comparison of frequencies and amplitudes of the oscillations. Different time step sizes dt are shown: $dt = 0.0025s$ (blue), $dt = 0.005s$, shift = $0.01s$ (red), $dt = 0.01s$, shift = $0.03s$ (orange), $dt = 0.02s$, shift = $0.08s$ (green), $dt = 0.04s$, shift = $0.28s$ (brown).

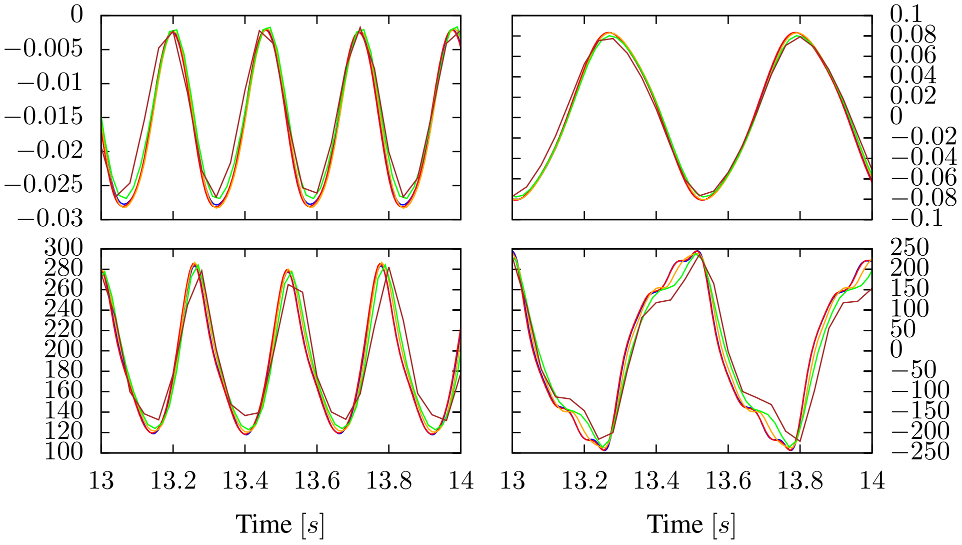

Figure 7. FSI-2 benchmark: Drag, lift (bottom row) and x/y-displacements (top row) with $Q(2) - Q(2) - P(1)$ elements on a 2 times refined grid. Time was shifted to allow better comparison of frequencies and amplitudes of the oscillations. Different time step sizes dt are shown: $dt = 0.0025s$ (blue), $dt = 0.005$, shift = $0.015s$ (red), $dt = 0.01s$, shift = $0.03s$ (orange), $dt = 0.02s$, shift = $0.07s$ (green), $dt = 0.04s$, shift = $0.28s$ (brown).

Copyright © 0000 John Wiley & Sons, Ltd.

6.2.3. Parallel Performance Studies For comparison, Figure 8 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 $\mathcal{O}(1/n)$ and $\mathcal{O}(1/\sqrt{n})$, 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 $\mathcal{O}(1/\sqrt{n})$ scalability as shown in Figure 9. This behavior is in agreement with the results presented in [45]. We note that both methods do not yield the optimal scalability of $\mathcal{O}(1/n)$. Most likely, this is due to the coupled block-structure of FSI problems. Indeed, replacing the inner solvers by perfectly scaling, but poorly convergent Richardson iterations (without any preconditioner), the same $\mathcal{O}(1/\sqrt{n})$ scalability can be observed.

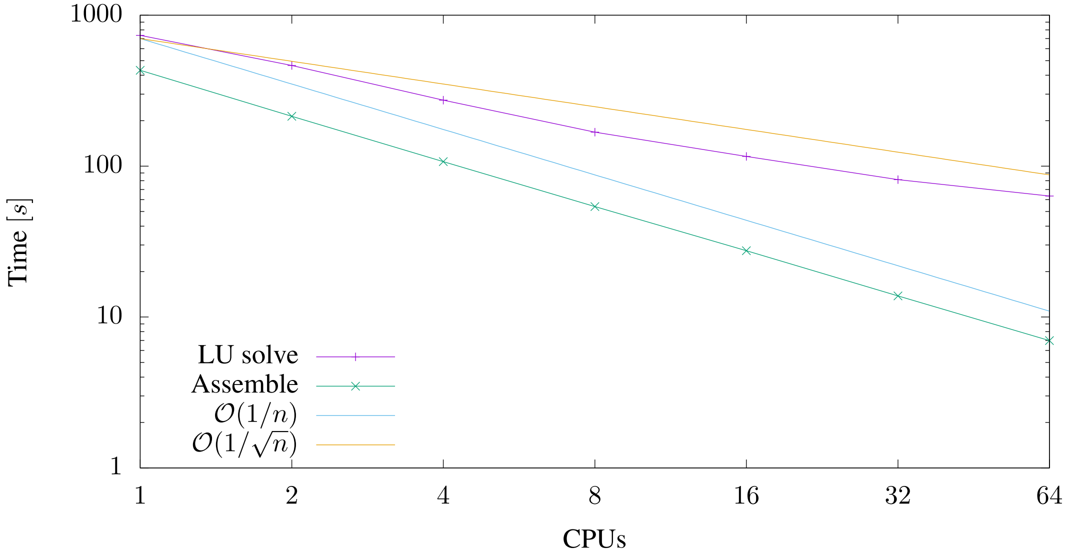

Figure 8. FSI-2 benchmark: Strong scalability of the sparse direct solver MUMPS (Multifrontal Massively Parallel Sparse Direct Solver) for approximately $1.8 \cdot 10^6$ dofs using $Q(2) - Q(2) - P(1)$ elements and time step size $dt = 0.01s$. Average time given in seconds for the solution of one linear system.

Copyright © 0000 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

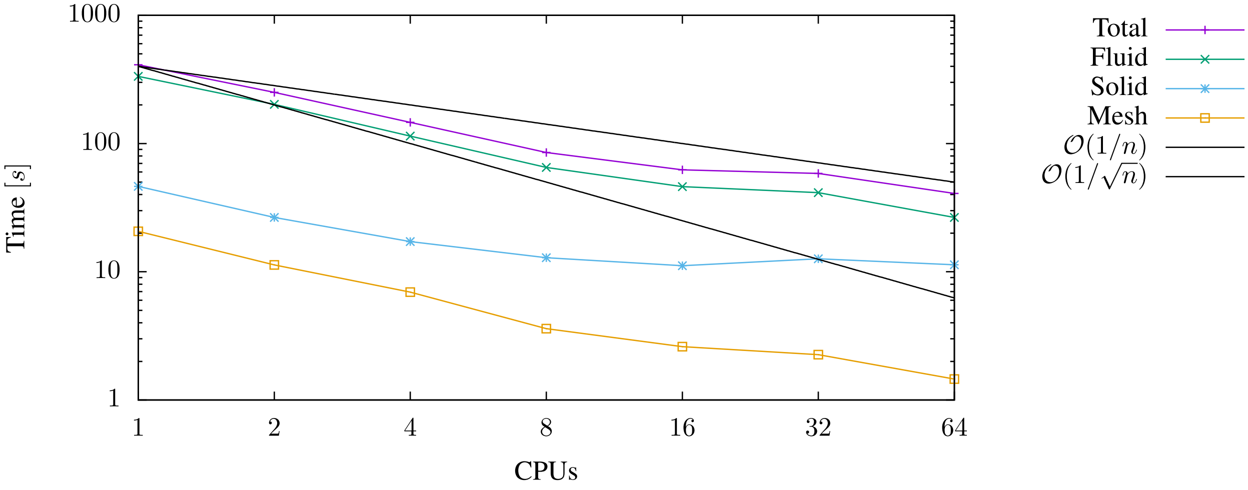

Figure 9. FSI-2 benchmark: Strong scalability using the preconditioned GMRES scheme for approximately $16 \cdot 10^6$ dofs using $Q(2) - Q(2) - P(1)$ elements and time step size $dt = 0.01s$. Average time given in seconds for the solution of one linear system.

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.

We finally extract some explicit time measurement numbers. 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.

6.3. Example 2: Flow Around an Elastic Obstacle

6.3.1. Description This numerical test features a $3d$ flow around an elastic obstacle and is taken from [36]. As in the first example, all parameters and geometry data are given in SI units. 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 inflow is smoothened similar to the previous test until $t = 0.5s$. The geometrical and material parameters and $\bar{v}$ are provided in the Tables I and II. The geometry is illustrated in Figure 10. Within

Copyright © 0000 John Wiley & Sons, Ltd.

DOI: 10.1002/fld

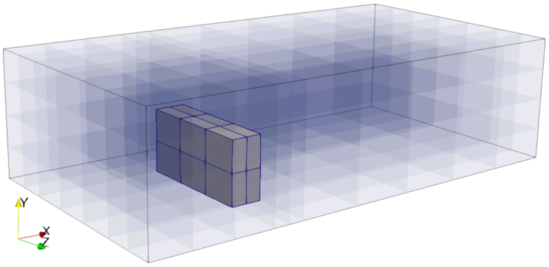

Figure 10. Graphical illustration of the geometry of Example 2. The elastic obstacle is displayed in grey color. The explanation of the axes is given in Table I.

this example, we used $Q(2) - Q(2) - P(1)$ elements, which correspond to $32k$ dofs on the coarsest grid ($r = 0$) up to $14m$ dofs for $r = 3$.

Quantity	Value
channel length (x-direction) L	$1.5m$
channel height (y-direction) H	$0.4m$
channel width (z-direction) $2H$	$0.8m$
obstacle height h	$0.2m$
obstacle width	$0.4m$
obstacle thickness	$0.1m$
inflow velocity $\bar{v}$	$1.0m/s$

Table I. Geometry and problem data for Example 2.

Quantity	Value
solid density $\hat{\rho}_s$	$10^3 kg/m^3$
Lamé λ	$2 \cdot 10^6 Pa$
Lamé μ	$0.5 \cdot 10^6 Pa$
Poisson ratio ν	0.4

Table II. Material parameters for Example 2.

6.3.2. Evaluation of Quantities of Interest As in Section 6.2.2, 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 11 – 13 using the scheme depicted in Table III. Since the problem is symmetric along the plane $z = 0$, the displacement in z-direction of P_1 and P_4 should be zero. Indeed, Figure 13 shows values in the order of numerical noise for these displacements.

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

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

6.3.3. Performance Studies We first start with comments on the nonlinear and linear solvers followed by an analysis of the parallel performance. Figure 14 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 during h-refinement. This happens because we limit the number of iterations for the subsolvers instead of prescribing a fixed tolerance. The applied solvers for the subproblems are not entirely h-robust, which leads to the observed increase of outer iterations. If we solve the subproblems sufficiently accurate, the number of iterations remains almost constant. The respective results of the 2d benchmarks (for sequential computations) are discussed in [53].

As we can see in Figure 15, 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 $\mathcal{O}(1/\sqrt{n})$. As in Example 1, the scaling is not optimal, which is again most likely due to the coupled structure of FSI.

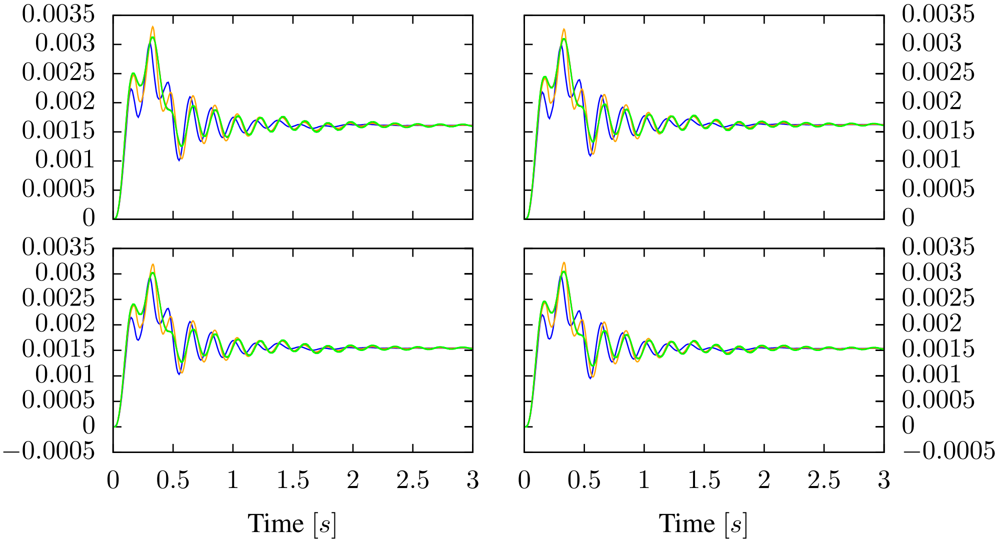

Figure 11. Example 2: x-displacement at points P_1, P_2 (top) and P_3, P_4 (bottom) for different time step sizes dt, $Q(2)-Q(2)-P(1)$ elements, and $r=0$ (blue: $dt=10^{-2}s$, orange: $dt=5\cdot10^{-3}s$, red: $dt=10^{-3}s$, green: $dt=5\cdot10^{-4}s$). Note that the green and red line coincide almost perfectly.

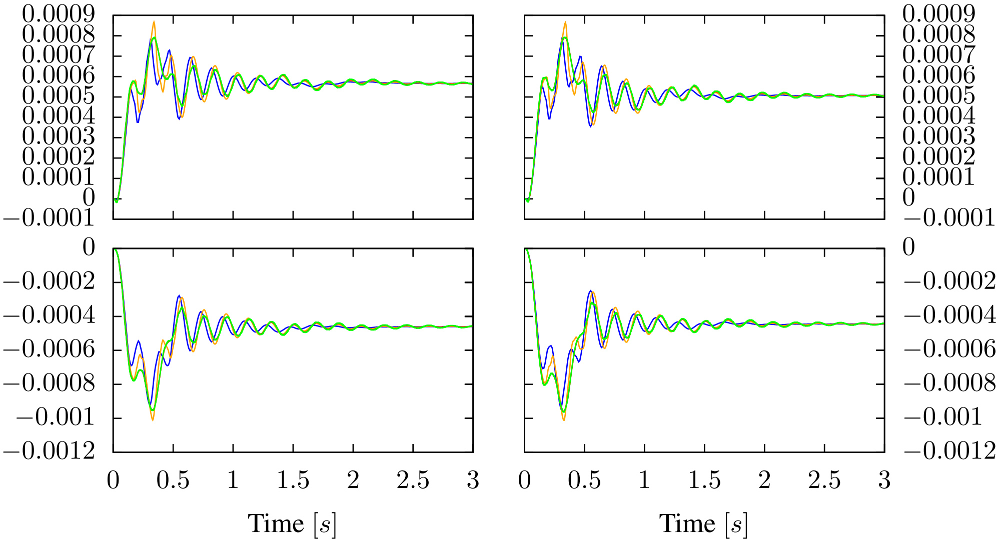

Figure 12. Example 2: y-displacement at points P_1, P_2 (top) and P_3, P_4 (bottom) for different time step sizes, $Q(2)-Q(2)-P(1)$ elements and $r=0$ (blue: $dt=10^{-2}s$, orange: $dt=5\cdot10^{-3}s$, red: $dt=10^{-3}s$, green: $dt=5\cdot10^{-4}s$). Note that the green and red line coincide almost perfectly.

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 $14\cdot10^6$ dofs at a single time step is

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

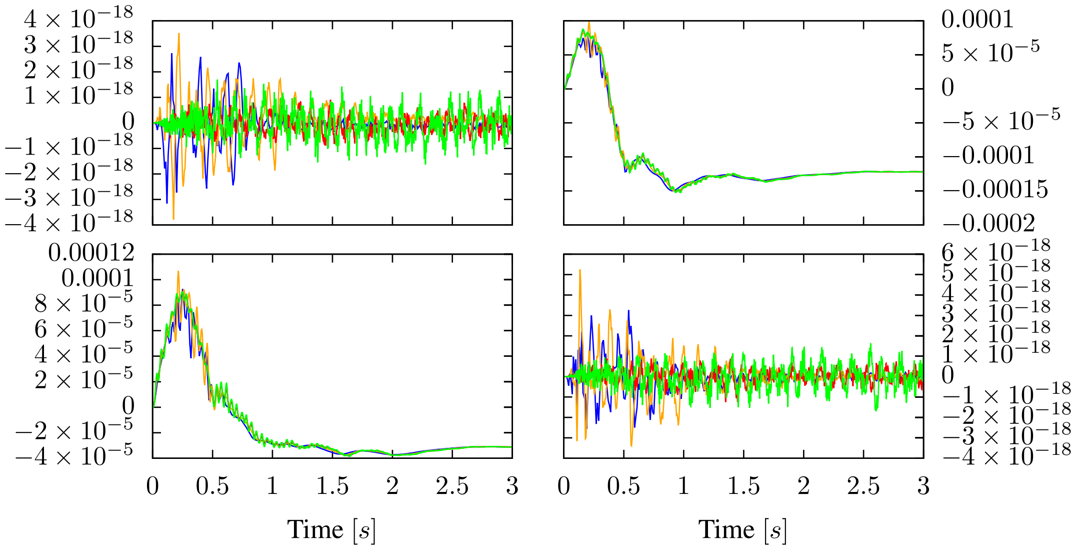

Figure 13. Example 2: z-displacement at points P_1, P_2 (top) and P_3, P_4 (bottom) for different time step sizes, $Q(2) - Q(2) - P(1)$ elements and $r = 0$ (blue: $dt = 10^{-2}s$, orange: $dt = 5 \cdot 10^{-3}s$, red: $dt = 10^{-3}s$, green: $dt = 5 \cdot 10^{-4}s$). Due to the symmetry of the problem, P_1 and P_4 yield almost zero z-displacements as expected. Note that the green and red line coincide almost perfectly for P_2 and P_3.

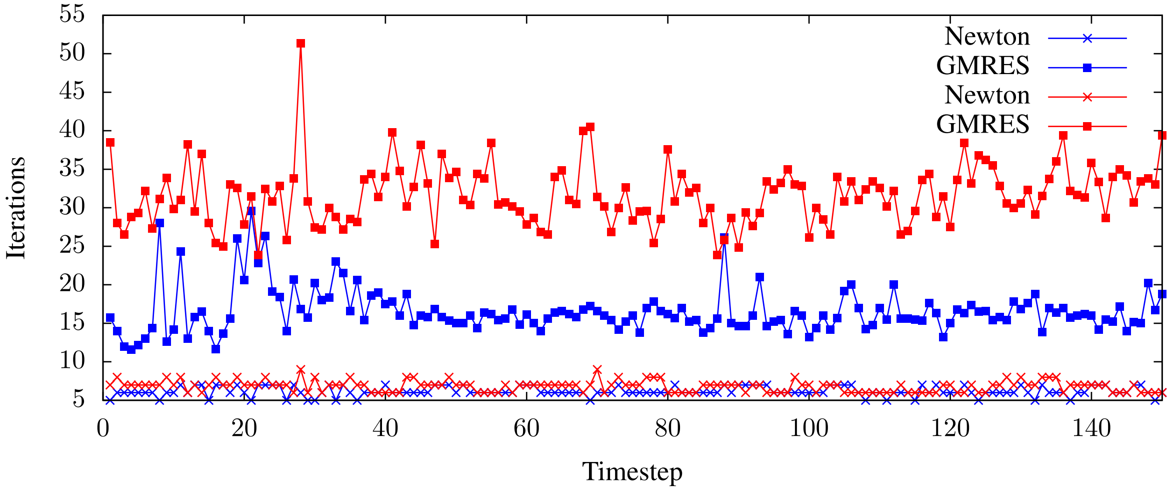

Figure 14. Number of iterations (Newton, average GMRES) required for the $3d$ test problem (blue: $r = 1$, red $r = 2$). The number of iterations for the subsolvers are limited, which yields an increase in the number of iterations. Solving them sufficiently accurate yields (almost) constant number of outer GMRES iterations.

7962 seconds, i.e., approximately 133 minutes. On 256 cores, the computational cost decreases to 1223 seconds, i.e., 20.4 minutes. Thus we achieve a reduction by 84% of the computational time.

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

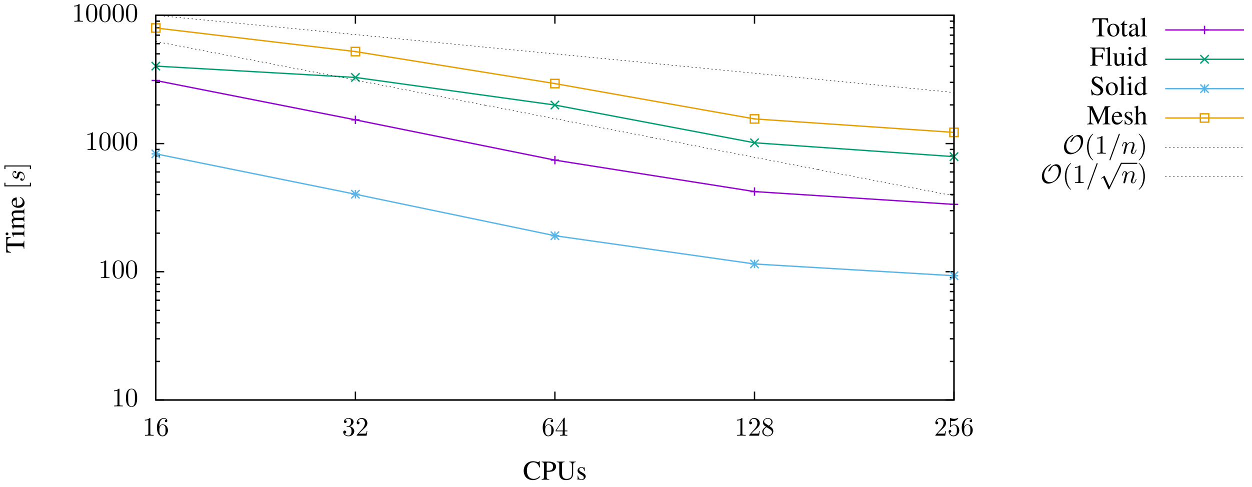

Figure 15. Example 2: Strong scalability using the preconditioned GMRES for approximately $14 \cdot 10^6$ dofs $(r = 3)$ in $3d$ $(Q(2) - Q(2) - P(1)$ elements) and time step size $dt = 0.005s$. Average time given in seconds for the solution of one linear system.

7. CONCLUSIONS

In this work, we developed a framework for the parallel solution of monolithic fluid-structure interaction (FSI) problems. The FSI problem 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 work. Indeed, block-wise preconditioners perform well in the serial case, but often lack perfect parallel scalability. This confirms the findings in [45, 95]. 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 [33], 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’.

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

References

1. Hsu MC, Bazilevs Y. Fluid–structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics* 2012; **50**(6):821–833.
2. Piperno S, Farhat C. Partitioned procedures for the transient solution of coupled aeroelastic problems - part ii: energy transfer analysis and three-dimensional applications. *Comput. Methods Appl. Mech. Engrg.* 2001; **190**:3147–3170.
3. Formaggia L, Quarteroni A, Veneziani A. *Cardiovascular Mathematics: Modeling and simulation of the circulatory system*. Springer-Verlag, Italia, Milano, 2009.
4. Bazilevs Y, Hsu MC, Zhang Y, Wang W, Liang X, Kvamsdal T, Brekken R, Isaksen JG. A fully-coupled fluid-structure interaction simulation of cerebral aneurysms. *Computational Mechanics* 2010; **46**:3–16, doi: 10.1007/s00466-009-0421-4.
5. Bungartz HJ, Schäfer M. *Fluid-Structure Interaction: Modelling, Simulation, Optimization*, *Lecture Notes in Computational Science and Engineering*, vol. 53. Springer: Berlin Heidelberg, 2006.
6. Formaggia L, Quarteroni A, Veneziani A. *Cardiovascular Mathematics: Modeling and simulation of the circulatory system*. Springer-Verlag, Italia, Milano, 2009.
7. Galdi G, Rannacher R. *Fundamental Trends in Fluid-Structure Interaction*. World Scientific, 2010.
8. Bazilevs Y, Takizawa K, Tezduyar T. *Computational Fluid-Structure Interaction: Methods and Applications*. John Wiley & Sons, Ltd, 2013.
9. Bodnar T, Galdi G, Necasova S. *Fluid-Structure Interaction and Biomedical Applications*. Birkhaeuser, 2014.
10. Bazilevs Y, Takizawa K ((eds.)). *Advances in Computational Fluid-Structure Interaction and Flow Simulation: New Methods and Challenging Computations*, Birkhäuser: Cham, Switzerland, 2016.
11. Richter T. *Fluid-structure interactions: models, analysis, and finite elements*. Springer, 2017.
12. Frei S, Holm B, Richter T, Wick T, Yang H. *Fluid-structure interactions: Fluid-Structure Interaction: Modeling, Adaptive Discretisations and Solvers*. Radon Series on Computational and Applied Mathematics 20, de Gruyter, 2017.
13. Mok D, Wall W. *Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures*. Trends in Computational Structural Mechanics, CIMNE: Barcelona, 2001.
14. Förster C, Wall WA, Ramm E. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**(7):1278 – 1293, doi:http://dx.doi.org/10.1016/j.cma.2006.09.002.
15. Santos NDD, Gerbeau JF, Bourgat J. A partitioned fluid-structure algorithm for elastic thin valves with contact. *Comp. Methods Appl. Mech. Engrg.* 2008; **197**(19-20):1750–1761.
16. Wall W, Genkinger S, Ramm E. A strong coupling partitioned approach for fluid-structure interaction with free surfaces. *Computers and Fluids* 2007; **36**:169–183.
17. Vierendeels J, Dumont K, Verdonck P. A partitioned strongly coupled fluid-structure interaction method to model heart valve dynamics. *J. Comp. Appl. Math.* 2008; .
18. Badia S, Nobile F, Vergara C. Fluid-structure partitioned procedures based on Robin transmission conditions. *J. Comp. Phys.* 2008; **227**:7027–7051.
19. Degroote J, Bruggeman P, Haelterman R, Vierendeels J. Stability of a coupling technique for partitioned solvers in FSI applications. *Comput. Struct.* 2008; :2224–2234.
20. Degroote J, Haelterman R, Annerel S, Bruggeman P, Vierendeels J. Performance of partitioned procedures in fluid-structure interaction. *Comput. Struct.* 2010; :446–457.
21. van Brummelen EH. Partitioned iterative solution methods for fluid-structure interaction. *International Journal for Numerical Methods in Fluids* 2010; **65**(1-3):3–27.
22. Langer U, Yang H. Partitioned solution algorithms for fluid-structure interaction problems with hyperelastic models. *Journal of Computational and Applied Mathematics* 2015; **276**(0):47 – 61, doi:http://dx.doi.org/10.1016/j.cam.2014.08.007. URL `http://www.sciencedirect.com/science/article/pii/S0377042714003641`.
23. Langer U, Yang H. Numerical simulation of fluid-structure interaction problems with hyperelastic models I: A partitioned approach. *Journal of Computational and Applied Mathematics* 2015; **276**:47–61.
24. Bungartz HJ, Lindner F, Mehl M, Uekermann B. A plug-and-play coupling approach for parallel multi-field simulations. *Computational Mechanics* Jun 2015; **55**(6):1119–1129.
25. van Brummelen EH, Roudbari MS, Simsek G, van der Zee K. *Binary-fluid-solid interaction based on the Navier-Stokes-Cahn-Hilliard equations*, *Radon Series on Computational and Applied Mathematics*, vol. 20. de Gruyter: Berlin, 2017.
26. Causin P, Gerbeau JF, Nobile F. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Comput. Methods Appl. Mech. Engrg.* 2005; **194**:4506–4527.
27. van Brummelen EH. Added mass effects of compressible and incompressible flows in fluid-structure interaction. *ASME. J. Appl. Mech.* 2009; **76**(2):021 206–021 206–7.
28. Heil M. An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**(1-2):1 – 23.
29. Hron J, Turek S. *Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow*. Springer Berlin Heidelberg: Berlin, Heidelberg, 2006; 371–385.
30. Bungartz HJ, Mehl M, Schäfer M. *Fluid Structure Interaction II: Modelling, Simulation, Optimization*, vol. 73. Springer Science & Business Media, 2010.
31. Badia S, Quaini Q, Quarteroni A. Splitting methods based on algebraic factorization for fluid-structure interaction. *SIAM Journal on Scientific Computing* 2008; **30**(4):1778–1805.
32. van Brummelen EH, van der Zee KG, de Borst R. Space/time multigrid for a fluid–structure-interaction problem. *Appl. Numer. Math.* 2008; **58**(12):1951–1971, doi:10.1016/j.apnum.2007.11.012.

33. Gee MW, Küttler U, Wall WA. Truly monolithic algebraic multigrid for fluid-structure interaction. *Int. J. Numer. Meth. Engng.* 2010; **85**:987–1016.
34. Langer U, Yang H. Robust and efficient monolithic fluid-structure-interaction solvers. *International Journal for Numerical Methods in Engineering* 2016; **108**(4):303–325.
35. Langer U, Yang H. Numerical simulation of fluid-structure interaction problems with hyperelastic models: A monolithic approach. *Mathematics and Computers in Simulation* 2018; **145**:186–208, doi: https://doi.org/10.1016/j.matcom.2016.07.008.
36. Richter T. A monolithic geometric multigrid solver for fluid-structure interactions in ale formulation. *International Journal for Numerical Methods in Engineering* 2015; **104**(5):372–390, doi:10.1002/nme.4943.
37. Hron J, Turek S. *A monolithic FEM/Multigrid solver for ALE formulation of fluid structure with application in biomechanics*, vol. 53. Springer-Verlag, 2006; 146–170.
38. Razzaq M, Damanik H, Hron J, Ouazzi A, Turek S. FEM multigrid techniques for fluid-structure interaction with application to hemodynamics. *Appl. Numer. Math.* 2012; **62**(9):1156–1170.
39. Mayr M, Klöppel T, Wall W, Gee M. A temporal consistent monolithic approach to fluid-structure interaction enabling single field predictors. *SIAM J. Sci. Comput.* 2015; **37**(1):B30–B59.
40. Barker A, Cai X. Scalable parallel methods for monolithic coupling in fluid-structure interaction with application to blood flow modeling. *J. Comput. Phys.* 2010; **229**(3):642–659.
41. Wu Y, Cai X. A fully implicit domain decomposition based ale framework for three-dimensional fluid-structure interaction with application in blood flow computation. *J. Comput. Physics* 2014; **258**:524–537.
42. Balzani D, Deparis S, Fausten S, Forti D, Heinlein A, Klawonn A, Quarteroni A, Rheinbach O, Schröder J. 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.* 2016; **32**(10).
43. Heil M, Hazel A, Boyle J. Solvers for large-displacement fluid-structure interaction problems: segregated versus monolithic approaches. *Comput. Mech.* 2008; **43**(1):91–101.
44. Badia S, Quaini A, Quarteroni A. Modular vs. non-modular preconditioners for fluid-structure systems with large added-mass effect. *Comput. Methods Appl. Mech. Engrg.* 2008; **197**(49-50):4216–4232.
45. Crosetto P, Deparis S, Fourestey G, Quarteroni A. Parallel algorithms for fluid-structure interaction problems in haemodynamics. *SIAM Journal on Scientific Computing* 2011; **33**(4):1598–1622, doi:10.1137/090772836.
46. Crosetto P, Reymond P, Deparis S, Kontaxakis D, Stergiopulos N, Quarteroni A. Fluid–structure interaction simulation of aortic blood flow. *Computers & Fluids* 2011; **43**(1):46 – 57, doi: http://dx.doi.org/10.1016/j.compfluid.2010.11.032. Symposium on High Accuracy Flow Simulations. Special Issue Dedicated to Prof. Michel DevilleSymposium on High Accuracy Flow Simulations.
47. Muddle R, Mihajlović M, Heil M. An efficient preconditioner for monolithically-coupled large-displacement fluid-structure interaction problems with pseudo-solid mesh updates. *J. Comput. Phys.* 2012; **231**(21):7315–7334.
48. Kong F, Cai XC. 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. DOI: 10.1177/1094342016646437.
49. Deparis S, Forti D, Grandperrin G, Quarteroni A. Facsi: A block parallel preconditioner for fluid-structure interaction in hemodynamics. *Journal of Computational Physics* 2016; **327**:700–718.
50. Forti D, Quarteroni A, Deparis S. A parallel algorithm for the solution of large-scale nonconforming fluid-structure interaction problems in hemodynamics. *J. Comput. Math.* 2017; **35**(3):363–380.
51. Deparis S, Forti D, Gervasio P, Quarteroni A. Internodes: an accurate interpolation-based method for coupling the galerkin solutions of pdes on subdomains featuring non-conforming interfaces. *Computers & Fluids* 2016; **141**:22–41.
52. Mehl M, Uekermann B, Bijl H, Blom D, Gatzhammer B, van Zuijlen A. Parallel coupling numerics for partitioned fluid-structure interaction simulations. *Computers & Mathematics with Applications* 2016; **71**(4):869 – 891.
53. Jodlbauer D, Wick T. A monolithic FSI solver applied to the FSI 1,2,3 benchmarks. *Fluid-Stucture Interaction: Modeling, Adaptive Discretizations and Solvers*, *Radon Series on Computational and Applied Mathematics*, vol. 20, Frei S, Holm B, Richter T, Wick T, Yang H (eds.). chap. 6, de Gruyter: Berlin, Boston, 2017; 193–234.
54. Hughes TJR, Liu WK, Zimmermann T. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Comput. Methods Appl. Mech. Engrg.* 1981; **29**:329–349.
55. Donea J, Giuliani S, Halleux JP. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Comput. Methods Appl. Mech. Engrg.* 1982; **33**:689–723.
56. Wick T. Fluid-structure interactions using different mesh motion techniques. *Computers & Structures* 2011; **89**(13 – 14):1456 – 1467.
57. Dunne T, Richter T, Rannacher R. *Numerical simulation of fluid-structure interaction based on monolithic variational formulations*. Comtemporary Challenges in Mathematical Fluid Mechanics, Springer: World Scientific, Singapore, 2010; 1–75.
58. Stein K, Tezduyar T, Benney R. Mesh moving techniques for fluid-structure interactions with large displacements. *J. Appl. Mech.* 2003; **70**:58–63.
59. Temam R. *Navier-Stokes equations; theory and numerical analysis*. Reprint. with corr. edn., AMS Chelsea publishing, American Math. Soc.: Providence, RI, 2001.
60. Lions JL, Magenes E. *Non-Homogeneous Boundary Value Problems and Applications*. Grundlehren der mathematischen Wissenschaften 181, Springer, 1972.
61. Heywood JG, Rannacher R, Turek S. Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations. *International Journal of Numerical Methods in Fluids* 1996; **22**:325–352.
62. Ciarlet P. Mathematical elasticity, volume i: Three-dimensional elasticity. *Acta Applicandae Mathematica* 1990; **18**(2):190–195, doi:10.1007/BF00046568. URL `https://doi.org/10.1007/BF00046568`.
63. Ball JM. Convexity conditions and existence theorems in nonlinear elasticity. *Archive for Rational Mechanics and Analysis* 1976; **63**(4):337–403, doi:10.1007/BF00279992. URL `https://doi.org/10.1007/BF00279992`.

64. Wick T. Adaptive Finite Element Simulation of Fluid-Structure Interaction with Application to Heart-Valve Dynamics. PhD Thesis, University of Heidelberg 2011. URL `http://www.ub.uni-heidelberg.de/archiv/12992`.
65. Bukac M, Canic S, Glowinski R, Muha B, Quaini A. A modular, operator-splitting scheme for fluid-structure interaction problems with thick structures. *International Journal for Numerical Methods in Fluids* 2014; **74**(8):577–604, doi:10.1002/fld.3863. URL `http://dx.doi.org/10.1002/fld.3863`.
66. Fernández MA, Gerbeau JF. *Algorithms for fluid-structure interaction problems*. Springer Milan: Milano, 2009; 307–346.
67. Bazilevs Y, Calo VM, Hughes T, Zhang Y. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Comput. Mech.* 2008; **43**:3–37.
68. Tallec PL, Mouro J. Fluid structure interaction with large structural displacements. *Comput. Methods Appl. Mech. Engrg.* 2001; **190**:3039–3067.
69. Wick T. Solving monolithic fluid-structure interaction problems in arbitrary lagrangian eulerian coordinates with the deal.ii library. *Archive of Numerical Software* 2013; **1**(1):1–19, doi:10.11588/ans.2013.1.10305.
70. Evans LC. *Partial differential equations*. American Mathematical Society, 2010.
71. Grandmont C. Existence for a three-dimensional steady state fluid-structure interaction problem. *Journal of Mathematical Fluid Mechanics* 2002; **4**(1):76–94, doi:10.1007/s00021-002-8536-9. URL `https://doi.org/10.1007/s00021-002-8536-9`.
72. Ignatova M, Kukavica I, Lasiecka I, Tuffaha A. On well-posedness for a free boundary fluid-structure model. *J. Math. Phys.* November 2012; **53**(11):115 624.
73. Grandemont C. Existence of weak solutions for the unsteady interaction of a viscous fluid with an elastic plate. *SIAM J. Math. Anal.* 2008; **40**(2):716–737.
74. Coutand D, Shkoller S. Motion of an elastic solid inside an incompressible viscous fluid. *Arch. Rational Mech. Anal.* 2005; **176**:25–102.
75. Coutand D, Shkoller S. The interaction between quasilinear elastodynamics and the Navier-Stokes equations. *Arch. Rational Mech. Anal.* 2006; **179**:303–352.
76. Wood WL. *Practical Time-stepping Schemes*. Clarendon Press: Gloucestershire, 1990.
77. Heywood JG, Rannacher R. Finite-element approximation of the nonstationary Navier-Stokes problem part iv: Error analysis for second-order time discretization. *SIAM Journal on Numerical Analysis* 1990; **27**(2):353–384.
78. Bristeau MO, Glowinski R, Periaux J. Numerical methods for the Navier-Stokes equations. *Comput. Phys. Rep.* 1987; **6**:73–187.
79. Richter T, Wick T. On time discretizations of fluid-structure interactions. *Multiple Shooting and Time Domain Decomposition Methods*, Carraro T, Geiger M, Körkel S, Rannacher R (eds.), Contributions in Mathematical and Computational Science, 2015; 377–400. URL `http://www.springer.com/us/book/9783319233208`.
80. Failer L, Wick T. Adaptive time-step control for nonlinear fluid-structure interaction. *Journal of Computational Physics* 2018; **366**:448 – 477.
81. Wick T. *Variational-Monolithic ALE Fluid-Structure Interaction: Comparison of Computational Cost and Mesh Regularity Using Different Mesh Motion Techniques*. Springer International Publishing: Cham, 2017; 261–275.
82. Turek S. *Efficient solvers for incompressible flow problems: An algorithmic approach*. 1999.
83. Weickert J. *Navier-stokes Equations as a Differential Algebraic System*. Preprint-Reihe des Chemnitzer SFB 393, Techn. Univ., 1996.
84. Fernández MA, Moubachir M. A Newton method using exact Jacobians for solving fluid–structure coupling. *Computers & Structures* 2005; **83**(2–3):127 – 142, doi:http://dx.doi.org/10.1016/j.compstruc.2004.04.021.
85. Holzapfel GA. *Nonlinear solid mechanics; a continuum approach for engineering*. Reprint. edn., John Wiley & Sons, Ltd, 2010.
86. Antman SS. *Nonlinear problems of elasticity*, *Applied mathematical sciences*, vol. 107. Springer: New York, 1995.
87. Heroux M, Bartlett R, Hoekstra VHR, Hu J, Kolda T, Lehoucq R, Long K, Pawlowski R, Phipps E, Salinger A, *et al.*. An Overview of Trilinos. *Technical Report SAND2003-2927*, Sandia National Laboratories 2003.
88. Gee MW, Siefert CM, Hu JJ, Tuminaro RS, Sala MG. ML 5.0 smoothed aggregation user's guide. *Technical Report SAND2006-2649*, Sandia National Laboratories 2006.
89. Bangerth W, Hartmann R, Kanschat G. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.* 2007; **33**(4):24/1–24/27.
90. Bangerth W, Davydov D, Heister T, Heltai L, Kanschat G, Kronbichler M, Maier M, Turcksin B, Wells D. The `deal.II` library, version 8.4. *Journal of Numerical Mathematics* 2016; **24**.
91. Karypis G, Kumar V. A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing* 1999; **20**(1):359–392.
92. Jodlbauer D. 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/.
93. Girault V, Raviart PA. *Finite Element method for the Navier-Stokes equations*. Number 5 in Computer Series in Computational Mathematics, Springer-Verlag, 1986.
94. Rannacher R. *Probleme der Kontinuumsmechanik und ihre numerische Behandlung*. Heidelberg University Publishing, 2017.
95. Forti D, Quarteroni A, Deparis S. A parallel algorithm for the solution of large-scale nonconforming fluid-structure interaction problems in hemodynamics. *J. Comput. Math.* 2017; **35**(3):363–380.