Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

Efficient monolithic solvers for fluid-structure interaction applied to flapping membranes

Daniel Jodlbauer Ulrich Langer Thomas Wick

DK-Report No. 2021-06 04 2021

A–4040 LINZ, ALTENBERGERSTRASSE 69, AUSTRIA

Supported by

Austrian Science Fund (FWF) Upper Austria

DK sponsors:

- **Johannes Kepler University Linz** (JKU)
- **Austrian Science Fund** (FWF)
- **Upper Austria**

Efficient monolithic solvers for fluid-structure interaction applied to flapping membranes*

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

[1] *Doctoral Program "Computational Mathematics", Johannes Kepler University Linz, Altenberger Str. 69, A-4040 Linz, Austria*

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

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

April 6, 2021

Abstract

In this contribution, we apply our recently developed monolithic solver with physics-based block-preconditioning to a flapping membrane example. In this application, the very thin solid flaps lead to a challenging problem, which require a well-chosen mesh motion model for which robust linear solvers are needed. The block preconditioner is revisited algorithmically and substantiated with some numerical studies in terms of channel flow with flapping membranes and elastic solid boundaries, which are inspired from hemodynamic applications.

1 Introduction

This work is devoted to the efficient solution of variational-monolithic fluid-structure interaction (FSI) initial-boundary value problems. Solvers for such monolithic systems were developed, e.g., in [1–9]. Due to the interface coupling conditions, the development of robust scalable parallel solvers remains a challenging task, and to the best of our knowledge only semi-cost optimal parallel approaches could be derived [8, 10]. The main purpose of this work consists in further numerical studies of the solver, developed in [8], for a benchmark problem that is motivated by hemodynamic applications. Specifically, we consider channel flow with elastic membranes and elastic solid walls. This situation is challenging because of the thin elastic flaps and was the motivation for fluid-structure interaction models such as immersed methods [11, 12]. However, we use arbitrary Lagrangian-Eulerian coordinates (see e.g., [13]), because of its high accuracy of the coupling conditions as the interface is tracked. For a careful evaluation of the performance of our physics-based block FSI preconditioner from [8], we use sparse direct solvers for the mesh, solid, and fluid subproblems. These sparse direct solvers should be replaced by iterative solvers in the case of large-scale problems with a high number of degrees of freedom. Therein, the flow part with well-known saddle-point structure becomes very critical, which was not yet the case for our solver applied to the FSI benchmarks in [6, 8]. The performance of our block FSI preconditioner

*This work has been supported by the Austrian Science Fund (FWF) grant P29181 'Goal-Oriented Error Control for Phase-Field Fracture Coupled to Multiphysics Problems', and by the Doctoral Program W1214-03 at the Johannes Kepler University Linz.

and overall linear GMRES solver is evaluated in terms of iteration numbers as well as memory storage. Moreover, iteration numbers of the nonlinear Newton solver are monitored. Finally, a computational convergence analysis for flap tip displacements, drag and lift for different spatial mesh levels is conducted.

2 FSI Model

Let the function spaces $\widehat{X}$ (including extensions of non-homogeneous Dirichlet conditions) and $\widehat{X}^0$ (homogeneous Dirichlet conditions) be given. Our variational-monolithic arbitrary Lagrangian-Eulerian FSI model from [14] (see also [8]) reads in space-time formulation as follows: Find a global vector-valued velocity $\hat{v}$, global vector-valued displacements $\hat{u} = \hat{u}_s + \hat{u}_f$, and a scalar-valued fluid pressure $\hat{p}_f$, i.e., $\hat{U} := (\hat{v}, \hat{u}, \hat{p}_f) \in \hat{X}$ such that the fluid/solid momentum equation

$$\begin{aligned}\int_I \Big((\hat{J}\hat{\rho}_f \partial_t \hat{v}, \hat{\psi}^v)_{\hat{\Omega}_f} + (\hat{\rho}_f \hat{J}(\hat{F}^{-1}(\hat{v} - \hat{w}) \cdot \hat{\nabla})\hat{v}), \hat{\psi}^v)_{\hat{\Omega}_f} + (\hat{J}\hat{\sigma}_f \hat{F}^{-T}, \hat{\nabla}\hat{\psi}^v)_{\hat{\Omega}_f} \\ + \langle \hat{\rho}_f \nu_f \hat{J}(\hat{F}^{-T}\hat{\nabla}\hat{v}^T \hat{n}_f)\hat{F}^{-T}, \hat{\psi}^v \rangle_{\hat{\Gamma}_{\text{out}}} + (\hat{\rho}_s \partial_t \hat{v}, \hat{\psi}^v)_{\hat{\Omega}_s} + (\hat{F}\hat{\Sigma}, \hat{\nabla}\hat{\psi}^v)_{\hat{\Omega}_s} \Big)\, dt \\ + (\hat{J}(\hat{v}(0) - \hat{v}_0), \hat{\psi}^v(0))_{\hat{\Omega}_f} + (\hat{v}(0) - \hat{v}_0, \hat{\psi}^v(0))_{\hat{\Omega}_s} = 0,\end{aligned}$$

the 2nd solid eq.

$$\int_I \Big(\hat{\rho}_s (\partial_t \hat{u}_s - \hat{v}|_{\hat{\Omega}_s}, \hat{\psi}^u_s)_{\hat{\Omega}_s} \Big)\, dt + (\hat{u}_s(0) - \hat{u}_{s,0}, \hat{\psi}^u_s(0)) = 0,$$

the mass conservation

$$\int_I \Big((\widehat{div}\, (\hat{J}\hat{F}^{-1}\hat{v}), \hat{\psi}^p_f)_{\hat{\Omega}_f} \Big)\, dt = 0,$$

and the mesh motion

$$\int_I (\hat{\sigma}_{\text{mesh}}, \hat{\nabla}\hat{\psi}^u_f)_{\hat{\Omega}_f}\, dt = 0,$$

hold for all $\hat{\Psi} = (\hat{\psi}^v, \hat{\psi}^u, \hat{\psi}^p_f) \in \hat{X}^0$, with $\hat{\psi}^u = \hat{\psi}^u_f + \hat{\psi}^u_s$. All notations and definitions of scalars, vectors, tensors, and transformations can be found in our prior work [8]. In compact form, the above problem reads: Find $\hat{U} \in \hat{X}$ such that

$$\hat{A}(\hat{U})(\hat{\Psi}) = 0 \quad \forall\, \hat{\Psi} \in \hat{X}^0,$$

where the FSI equations are combined in the semi-linear form $\hat{A}(\hat{U})(\hat{\Psi})$.

3 Numerical solution

3.1 Newton linearization

The previous FSI model is discretized in time by an A stable implicit finite difference scheme and in space by finite elements on quadrilaterals. The temporal and spatial discretization parameters are denoted by k and h, respectively. At time step t_n, we need to solve for U_h^{n+1} at t_{n+1} for which we utilize Newton's method. At each Newton step (index j), we have to solve a linear variational problem of the form

$$\begin{aligned}\underbrace{A'(U_h^{n,j})(\delta U_h, \Psi_h)}_{=A\delta U} &= \underbrace{-A(U_h^{n,j})(\Psi_h)}_{=B} \;\forall\, \Psi_h \in X_h^0 \subset X^0, \\ U_h^{n,j+1} &= U_h^{n,j} + \lambda \delta U_h, \quad \lambda \in (0,1].\end{aligned}$$

Thus, we finally obtain the linear system of finite element equations

$$A\delta U = B$$

for determining the Newton correction δU. We note that the finite element functions and operators are identified with the corresponding matrix and vector representations via the finite element isomorphism. Up to 10^5 unknowns in $2d$, respectively, 10^4 unknowns in 3D, sparse direct solvers work still fine in the context of FSI problems. However, for large-scale problems with considerable more unknowns, we should use preconditioned iterative solvers in order to reduce the memory demand and the computational costs in terms of arithmetical operations required.

3.2 Block structure of linear systems

Since the FSI problem is non-symmetric, a GMRES scheme (generalized minimal residual) is a classical choice for the overall solution of the linear system arising at each Newton iteration. In order to reduce the number of GMRES iterations, one needs a suitable preconditioner P for the system matrix A. In [8], we have constructed a (left) preconditioner P such that

$$P^{-1}A\delta U = P^{-1}B$$

with $P^{-1} \approx A^{-1}$ in the sense that $P^{-1}A$ is close to the identity matrix I. We refer the reader to [15] for GMRES convergence results.

Observing the previous FSI model, we have three unknowns when global continuity of the displacements $\hat{u}_f$ and $\hat{u}_s$ and $\hat{v}_f$ and $\hat{v}_s$ is realized, which is due to the variational-monolithic coupling scheme. Consequently, $\hat{u}, \hat{v}, \hat{p}$ are obtained from three principal problems: (M) mesh motion, (F) fluid, (S) solid. This results into the following 3×3 block system:

$$A := \begin{bmatrix} \mathcal{M} & \mathcal{C}_{ms} & 0 \\ \mathcal{C}_{sm} & \mathcal{S} & \mathcal{C}_{sf} \\ \mathcal{C}_{fm} & \mathcal{C}_{fs} & \mathcal{F} \end{bmatrix}.$$

A brief analysis yields that the principal problems appear on the diagonal. The coupling terms $\mathcal{C}_{**}$ are on the off-diagonals. In [1], details on the influence of the coupling terms $\mathcal{C}_{**}$ were studied on the overall solver behavior. Aiming for cost-optimal parallel schemes, the interface coupling terms play however a crucial rule [8].

3.3 Physics-based preconditioner

We now concentrate on the construction of the preconditioner P^{-1}, which is based on a simplified LDU block factorization

$$A \approx \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \mathcal{C}_{fm}\mathcal{M}^{-1} & \tilde{\mathcal{C}}_{fs}\mathcal{S}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathcal{M} & 0 & 0 \\ 0 & \mathcal{S} & 0 \\ 0 & 0 & \mathcal{F} \end{bmatrix} \begin{bmatrix} I & \mathcal{M}^{-1}\mathcal{C}_{ms} & 0 \\ 0 & I & \mathcal{S}^{-1}\mathcal{C}_{sf} \\ 0 & 0 & I \end{bmatrix} = LDU = P,$$

where we neglect the coupling term $\mathcal{C}_{sm}$. We have (see [6][Section 6.4.3]) $\tilde{\mathcal{C}}_{fs} = \mathcal{C}_{fs} - \mathcal{C}_{fm}\mathcal{M}^{-1}\mathcal{C}_{ms}$. Having such a decomposition, it is easy to compute the action of the inverse. We note that, in Krylov subspace methods, we only need the action of P^{-1} on the residual r.

From linear algebra we know that $P^{-1}r = U^{-1}D^{-1}L^{-1}r$ with $P = LDU$ from above. Consecutively solving with L, D and U yields the following result:

Algorithm 1 *Evaluation of $P^{-1}r$ (matrix-vector multiplications):*

```
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 $x_m = x_m - \mathcal{M}^{-1}\mathcal{C}_{ms}x_s$
```

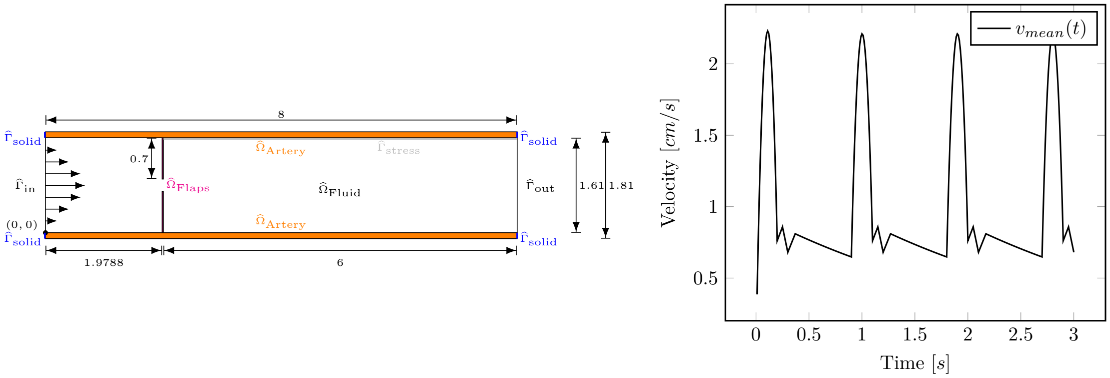

Figure 1: Geometry with inflow profile (left) and mean inflow velocity (right).

It remains to discuss the solutions of the subproblems with the system matrices $\mathcal{M}$, $\mathcal{S}$ and $\mathcal{F}$. In our $2d$ numerical example presented in Sect. 4, we use the sparse direct solver MUMPS[1] that solves these smaller subproblems very efficiently. However, if the subproblems are larger, we should replace the direct solvers for $\mathcal{M}^{-1}$, $\mathcal{S}^{-1}$ and $\mathcal{F}^{-1}$ by preconditioned iterative solvers $\widetilde{\mathcal{M}}^{-1}$, $\widetilde{\mathcal{S}}^{-1}$ and $\widetilde{\mathcal{F}}^{-1}$; see [6, 8], where we used AMG-based solvers for the subproblems. The implementation is based on the open-source finite element package deal.II [16].

4 Flapping membranes with elastic solid walls

This example was originally inspired from [11], later extended by ourselves, and the current configuration was recently used in [17] for optimal control with fluid-structure interaction.

The geometry is shown in Figure 1 (left). It consists of the fluid domain $\widehat{\Omega}_{\text{Fluid}} := (0, 8) \times (0.0, 1.61) \setminus \widehat{\Omega}_{\text{Flaps}}$ with inscribed flaps $\widehat{\Omega}_{\text{Flaps}} := (1.9788, 2.0) \times ((0, 0.7) \cup (0.91, 1.61))$. It is further surrounded by elastic arteries $\widehat{\Omega}_{\text{Artery}} := (0, 8) \times ((-0.1, 0.0) \cup (1.61, 1.71))$ on the top and bottom of the computational domain.

On the inflow boundary, $\widehat{\Gamma}_{\text{in}} := \{0\} \times (0, 1.61)$, we prescribe a parabolic inflow profile $\hat{v}(0, y, t) := 6(1.61)^{-2} y(1.61 - y) v_{\text{mean}}(t)$ for $t \in I := [0, 3.6]$, where $v_{\text{mean}}(t)$ is given by the profile in Figure 1 (right). At the outflow boundary the do-nothing outflow condition $\widehat{\Gamma}_{\text{out}}$ is prescribed for $\hat{v}$ and $\hat{p}$. The elastic walls are fixed at the left and right, i.e., on $\widehat{\Gamma}_{\text{solid, left}} := \{0\} \times ((-0.1, 0.0) \cup (1.61, 1.71))$ and $\widehat{\Gamma}_{\text{solid, right}} := \{8\} \times ((-0.1, 0.0) \cup (1.61, 1.71))$, we prescribe $\hat{u} = 0$ and $\hat{v} = 0$.

The computations are performed on the time interval $I = (0, 3.6\text{s})$. The fluid parameters are given by the kinematic viscosity $\nu_f = 10^{-1}\text{cm}^2\,\text{s}^{-1}$, and density $\hat{\rho}_f = 10^2\text{g}\,\text{cm}^{-3}$. In the solid domains $\widehat{\Omega}_{\text{Flaps}}$ and $\widehat{\Omega}_{\text{Artery}}$, we use a Poisson ratio $\nu = 0.4$, and density $\rho_s = 10^2\text{g}\,\text{cm}^{-3}$. The Lamé parameters are given by $\mu_s^{\text{flaps}} = 2.0 \cdot 10^7\text{g}\,\text{cm}^{-1}\,\text{s}^{-2}$ in $\widehat{\Omega}_{\text{Flaps}}$, and $\mu_s^{\text{walls}} = 1.0 \cdot 10^9\text{g}\,\text{cm}^{-1}\,\text{s}^{-2}$ in $\widehat{\Omega}_{\text{Artery}}$.

We are interested in evaluating the number of GMRES iterations per linear solve in each Newton step. Moreover, we monitor the number of nonlinear iterations, the position of the tip $(2, 0.91)$ of the upper elastic flap, and the drag and lift acting on the wall behind it. The latter quantities are defined by $(\text{drag}, \text{lift}) = \int_{\widehat{\Gamma}_{\text{stress}}} \hat{J} \left(-\hat{p}I + \hat{\rho}_f \hat{\nu}_f (\hat{\nabla}\hat{v}\hat{F}^{-1} + \hat{F}^{-T}\nabla v^T) \right) \hat{F}^{-T} \hat{n}\, d\hat{s}$ with $\widehat{\Gamma}_{\text{stress}} := (2, 8) \times \{1.61\}$.

Figure 2 shows that, during the whole simulation, we require an almost constant number of 4 to 6 Newton iterations. Similarly, the average number of linear GMRES iterations stays between 8 and 11 during refinement, although a slight increase can be observed on the finer grids. The

[1] http://mumps.enseeiht.fr/

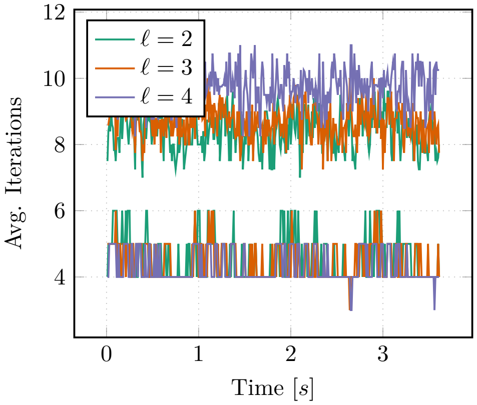

Figure 2: Number of GMRES (higher values) and Newton iterations (lower values).

ℓ	DoFs	Assemble[s]	Build[s]	Solve[s]	Total[s]
2	$8.7 \cdot 10^3$	$1.9 \cdot 10^2$	$3.6 \cdot 10^2$	$7.0 \cdot 10^0$	$5.9 \cdot 10^2$
3	$3.4 \cdot 10^4$	$7.8 \cdot 10^2$	$2.0 \cdot 10^3$	$3.3 \cdot 10^1$	$2.9 \cdot 10^3$
4	$1.3 \cdot 10^5$	$3.1 \cdot 10^3$	$1.2 \cdot 10^4$	$1.5 \cdot 10^2$	$1.6 \cdot 10^4$

ℓ	DoFs	Assemble[s]	Build[s]	Solve[s]	Total[s]
2	$8.7 \cdot 10^3$	$2.6 \cdot 10^2$	$1.7 \cdot 10^2$	$2.9 \cdot 10^2$	$1.0 \cdot 10^3$
3	$3.4 \cdot 10^4$	$1.0 \cdot 10^3$	$1.0 \cdot 10^3$	$8.7 \cdot 10^2$	$4.2 \cdot 10^3$
4	$1.3 \cdot 10^5$	$3.9 \cdot 10^3$	$5.4 \cdot 10^3$	$3.6 \cdot 10^3$	$1.8 \cdot 10^4$

Table 1: Timings of different components of the finite element simulation that include the time to assemble the sparse matrix, building the preconditioner/direct solvers, plain solving time and total simulation run-time. Minor contributions like assembly of residuals and functional evaluations are omitted. Top: sparse direct solver used for the solution of the complete linear systems. Bottom: GMRES with the proposed preconditioner using sparse direct solvers for fluid, solid and mesh.

computational aspects of certain parts of our simulation are summarized in Table 1. In comparison to a sparse direct solver for the full FSI problem, we need slightly more time on the coarser levels. However, for $\ell = 4$, the GMRES with sparse direct solvers for the smaller subproblems becomes competitive in terms of performance. Furthermore, the memory footprint of the iterative variant is roughly halved compared to the sparse direct solver; see Table 2. We note that for $2d$ problems, sparse direct solvers are hard to beat in terms of performance. For larger problems, we can split the application of the direct solver to the respective subproblems. This reduces the amount of memory and flops required to compute the factorization.

The resulting drag and lift values are visualized in Figure 3, the elongation of the tip is plotted in Figure 4. All these functional evaluations show surprisingly good agreement throughout the various levels of refinement. Only small differences are visible at the tips. As expected due to the symmetry of the configuration, evaluating the displacement, drag, or lift in the lower or upper part does not make a difference (apart from the sign).

ℓ	DoFs	Matrix[B]	Fluid[B]	Mesh[B]	Solid[B]	Sum[B]	Full[B]
2	$8.7 \cdot 10^3$	$6.3 \cdot 10^6$	$4.0 \cdot 10^6$	$4.0 \cdot 10^6$	$3.0 \cdot 10^6$	$1.1 \cdot 10^7$	$2.1 \cdot 10^7$
3	$3.4 \cdot 10^4$	$2.5 \cdot 10^7$	$1.8 \cdot 10^7$	$1.6 \cdot 10^7$	$1.4 \cdot 10^7$	$4.8 \cdot 10^7$	$9.3 \cdot 10^7$
4	$1.3 \cdot 10^5$	$1.0 \cdot 10^8$	$8.2 \cdot 10^7$	$7.6 \cdot 10^7$	$6.5 \cdot 10^7$	$2.2 \cdot 10^8$	$4.3 \cdot 10^8$

Table 2: Memory requirements in bytes for the full direct solver and our preconditioner (sum of fluid, solid, and mesh solver).

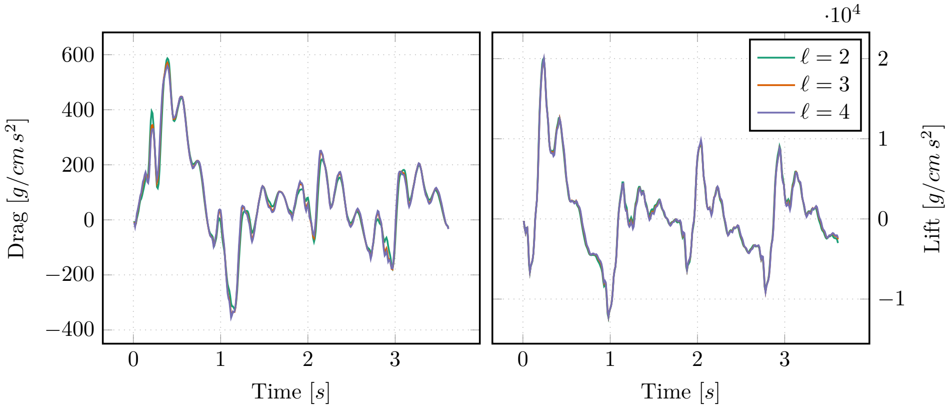

Figure 3: Drag (left) and lift (right) evaluated at the artery behind the top flap, i.e., $(2.0, 8.0) \times \{1.61\}$.

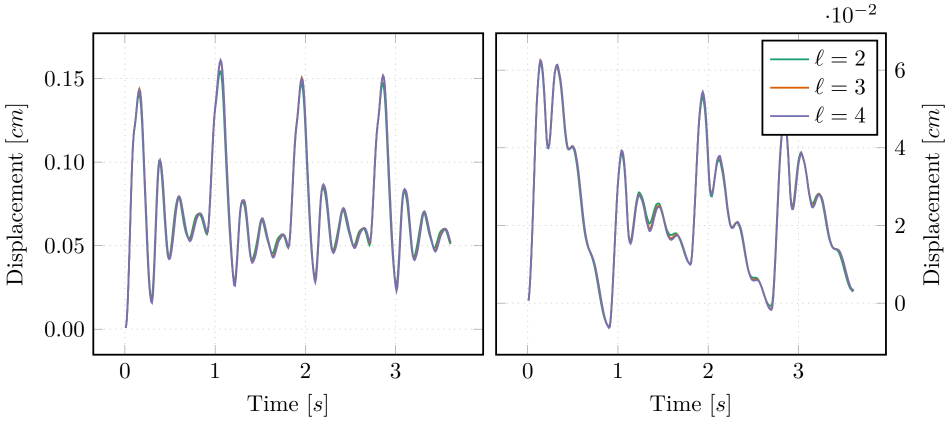

Figure 4: Displacement of the top flap at $(2.0, 0.91)$ in x-direction (left) and y-direction (right).

5 Conclusions and Outlook

We presented a preconditioner based on a block-LDU-decomposition of the linear systems for a challenging $2d$ FSI problem. For a small number of degrees of freedoms, a sparse direct solver for the full problem is hard to beat. Nonetheless, the reduction of the sparse direct solver to the separate subproblems already leads to an improvement of a factor 2 in terms of memory requirements. For large systems, the storage cost and computational complexity of sparse direct solvers becomes a prohibitive barrier. Replacing the solvers for the fluid, solid and mesh problems by iterative or matrix-free techniques may solve this issue. Implementing matrix-free solvers for FSI is a very challenging task, mainly caused by the difficulties to treat the fluid subproblem. In [18], we have applied the matrix-free technique successfully to fracture propagation.

References

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

[2] A. T. Barker and X.-C. Cai, "Scalable parallel methods for monolithic coupling in fluid-structure interaction with application to blood flow modeling," *J. Comp. Phys.*, vol. 229, no. 3, pp. 642–659, 2010.

[3] M. W. Gee, U. Küttler, and W. A. Wall, "Truly monolithic algebraic multigrid for fluid–structure interaction," *Comput. Methods Appl. Mech. Engrg.*, vol. 85, no. 8, pp. 987–1016, 2011.

[4] T. Richter, "A monolithic geometric multigrid solver for fluid-structure interactions in ALE formulation," *Int. J. Numer. Methods Eng.*, vol. 104, no. 5, pp. 372–390, 2015.

[5] U. Langer and H. Yang, "Robust and efficient monolithic fluid-structure-interaction solvers," *Int. J. Numer. Methods Eng.*, vol. 108, no. 4, pp. 303–325, 2016.

[6] D. Jodlbauer and T. Wick, "A monolithic FSI solver applied to the FSI 1,2,3 benchmarks," in *Fluid-Structure Interaction: Modeling, Adaptive Discretization and Solvers*, ser. Radon Series on Computational and Applied Mathematics, S. Frei, B. Holm, T. Richter, T. Wick, and H. Yang, Eds., vol. 20, de Gruyter, Berlin, 2017, pp. 193–234.

[7] U. Langer and H. Yang, "Recent development of robust monolithic fluid-structure interaction solvers," in *Fluid-Structure Interaction: Modeling, Adaptive Discretization and Solvers*, ser. Radon Series on Computational and Applied Mathematics, S. Frei, B. Holm, T. Richter, T. Wick, and H. Yang, Eds., vol. 20, de Gruyter, Berlin, 2017, pp. 169–191.

[8] D. Jodlbauer, U. Langer, and T. Wick, "Parallel block-preconditioned monolithic solvers for fluid-structure interaction problems," *Int. J. Numer. Methods Eng.*, vol. 117, no. 6, pp. 623–643, 2019.

[9] 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. Eng.*, vol. 32, no. 10, e02756, 2016.

[10] P. Crosetto, S. Deparis, G. Fourestey, and A. Quarteroni, “Parallel algorithms for fluid-structure interaction problems in haemodynamics,” *SIAM J. Sci. Comput.*, vol. 33, no. 4, pp. 1598–1622, 2011.

[11] A. J. Gil, A. A. Carreno, J. Bonet, and O. Hassan, “The immersed structural potential method for haemodynamic applications,” *J. Comput. Phys.*, vol. 229, pp. 8613–8641, 2010.

[12] C. S. Peskin, “Flow patterns around heart valves: A numerical method,” *J. Comp. Phys.*, vol. 10, no. 2, pp. 252–271, 1972.

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

[14] T. Wick, “Fluid-Structure Interactions using Different Mesh Motion Techniques,” *Comput. Struct.*, vol. 89, no. 13-14, pp. 1456–1467, 2011.

[15] Y. Saad, *Iterative Methods for Sparse Linear Systems.* Philadelphia: SIAM, 2003.

[16] D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells, “The deal.II finite element library: Design, features, and insights,” *Comput. Math. with Appl.*, 2020.

[17] T. Wick and W. Wollner, “Optimization with nonstationary, nonlinear monolithic fluid-structure interaction,” *Int. J. Numer. Methods Eng.*, pp. 1–20, 2020, https://doi.org/10.1002/nme.6372.

[18] D. Jodlbauer, U. Langer, and T. Wick, “Matrix-free multigrid solvers for phase-field fracture problems,” *Comput. Methods Appl. Mech. Engrg.*, vol. 372, p. 113 431, 2020.

Technical Reports of the Doctoral Program "Computational Mathematics"

2021

2021-01 J. Qi: *A tree-based algorithm on monomials in the Chow group of zero cycles in the moduli space of stable pointed curves of genus zero* Jan 2021. Eds.: S. Radu, J. Schicho

2021-02 A. Jiménez Pastor, P. Nuspl, V. Pillwein: *On C^2-finite sequences* Feb 2021. Eds.: M. Kauers, P. Paule

2021-03 A. Jiménez Pastor: *Simple differentially definable functions* Feb 2021. Eds.: M. Kauers, V. Pillwein

2021-04 U. Langer, A. Schafelner: *Simultaneous space-time finite element methods for parabolic optimal control problems* March 2021. Eds.: V. Pillwein, R. Ramlau

2021-05 U. Langer, A. Schafelner: *Space-time hexahedral finite element methods for parabolic evolution problems* March 2021. Eds.: B. Jüttler, V. Pillwein

2021-06 D. Jodlbauer, U. Langer, T. Wick: *Efficient monolithic solvers for fluid-structure interaction applied to flapping membranes* April 2021. Eds.: B. Jüttler, V. Pillwein

2020

2020-01 N. Smoot: *A Single-Variable Proof of the Omega SPT Congruence Family Over Powers of 5* Feb 2020. Eds.: P. Paule, S. Radu

2020-02 A. Schafelner, P.S. Vassilevski: *Numerical Results for Adaptive (Negative Norm) Constrained First Order System Least Squares Formulations* March 2020. Eds.: U. Langer, V. Pillwein

2020-03 U. Langer, A. Schafelner: *Adaptive space-time finite element methods for non-autonomous parabolic problems with distributional sources* March 2020. Eds.: B. Jüttler, V. Pillwein

2020-04 A. Giust, B. Jüttler, A. Mantzaflaris: *Local (T)HB-spline projectors via restricted hierarchical spline fitting* March 2020. Eds.: U. Langer, V. Pillwein

2020-05 K. Banerjee, M. Ghosh Dastidar: *Hook Type Tableaux and Partition Identities* June 2020. Eds.: P. Paule, S. Radu

2020-06 A. Bostan, F. Chyzak, A. Jiménez-Pastor, P. Lairez: *The Sage Package* `comb_walks` *for Walks in the Quarter Plane* June 2020. Eds.: M. Kauers, V. Pillwein

2020-07 A. Meddah: *A stochastic multiscale mathematical model for low grade Glioma spread* June 2020. Eds.: E. Buckwar, V. Pillwein

2020-08 M. Ouafoudi: *A Mathematical Description for Taste Perception Using Stochastic Leaky Integrate-and-Fire Model* June 2020. Eds.: E. Buckwar, V. Pillwein

2020-09 A. Bostan, A. Jiménez-Pastor: *On the exponential generating function of labelled trees* July 2020. Eds.: M. Kauers, V. Pillwein

2020-10 J, Forcan, J. Qi: *How fast can Dominator win in the Maker-Breaker domination game?* July 2020. Eds.: V. Pillwein, J. Schicho

2020-11 J. Qi: *A calculus for monomials in Chow group of zero cycles in the moduli space of stable curves* Sept 2020. Eds.: P. Paule, J. Schicho

2020-12 J. Qi, J. Schicho: *Five Equivalent Ways to Describe a Phylogenetic Tree* November 2020. Eds.: M. Kauers, S. Radu

The complete list since 2009 can be found at
`https://www.dk-compmath.jku.at/publications/`

Doctoral Program
"Computational Mathematics"

Director:

Assoc. Prof. Dr. Veronika Pillwein
Research Institute for Symbolic Computation

Deputy Director:

Prof. Dr. Bert Jüttler
Institute of Applied Geometry

Address:

Johannes Kepler University Linz
Doctoral Program "Computational Mathematics"
Altenbergerstr. 69
A-4040 Linz
Austria
Tel.: ++43 732-2468-6840

E-Mail:

office@dk-compmath.jku.at

Homepage:

http://www.dk-compmath.jku.at