Comput Mech (2008) 43:91–101
DOI 10.1007/s00466-008-0270-6

ORIGINAL PAPER

Solvers for large-displacement fluid–structure interaction problems: segregated versus monolithic approaches

Matthias Heil · Andrew L. Hazel · Jonathan Boyle

Received: 31 October 2007 / Accepted: 2 March 2008 / Published online: 26 March 2008

Abstract We compare the relative performance of monolithic and segregated (partitioned) solvers for large-displacement fluid–structure interaction (FSI) problems within the framework of OOMPH- LIB, the object-oriented multi-physics finite-element library, available as open-source software at http://www.oomph-lib.org. Monolithic solvers are widely acknowledged to be more robust than their segregated counterparts, but are believed to be too expensive for use in large-scale problems. We demonstrate that monolithic solvers are competitive even for problems in which the fluid–solid coupling is weak and, hence, the segregated solvers converge within a moderate number of iterations. The efficient monolithic solution of large-scale FSI problems requires the development of preconditioners for the iterative solution of the linear systems that arise during the solution of the monolithically coupled fluid and solid equations by Newton's method. We demonstrate that recent improvements to OOMPH- LIB's FSI preconditioner result in mesh-independent convergence rates under uniform and non-uniform (adaptive) mesh refinement, and explore its performance in a number of two- and three-dimensional test problems involving the interaction of finite-Reynolds-number flows with shell and beam structures, as well as finite-thickness solids.

Keywords Fluid-structure interaction · Monolithic solvers · Preconditioning

1 Introduction

Numerical methods for the solution of large-displacement fluid–structure interaction (FSI) problems can be classified as either segregated (partitioned) or monolithic. In a monolithic approach, the complete system of nonlinear algebraic equations that arises from the coupled discretisation of the equations of motion in the fluid and solid domains is solved as a whole, typically using a variant of Newton's method. This approach is generally acknowledged to be more robust than a segregated approach in which separate fluid and solid problems are coupled via a Picard (fixed point) iteration. It is widely believed, however, that monolithic solvers (i) are too computationally expensive and (ii) cannot take advantage of software modularity to the same extent as segregated solvers. The alleged superiority of the latter approach is generally attributed to the fact that in a segregated scheme "smaller and better conditioned subsystems are solved instead of one overall problem" [1]. It is also believed to be "difficult to devise efficient global preconditioners and to maintain state-of-the-art schemes in each solver" [2] when a monolithic solver is used. As a result, the monolithic approach is often regarded as unsuitable for large-scale problems.

OOMPH- LIB, the object-oriented multi-physics finite-element library, available as open-source software at http://www.oomph-lib.org, was developed to address these concerns. A main design goal is to provide an environment that facilitates the monolithic discretisation and solution of multi-physics problems. The overall design of the library has been discussed in Ref. [3] and further details can be found in the online documentation. Here, we provide a brief overview of the library's overall structure and an outline of the implementation for FSI problems (Sect. 2). The main objective of the paper is to assess the performance of different solution strategies for large-displacement FSI problems. In Sect. 3 we compare the performance of monolithic and segregated solvers and demonstrate that the former are competitive even in cases where the fluid–solid coupling is relatively weak and, hence, segregated solvers converge in a moderate num-

M. Heil (✉) · A. L. Hazel · J. Boyle
School of Mathematics, University of Manchester,
Oxford Road, Manchester M13 9PL, UK
e-mail: mheil@maths.manchester.ac.uk

ber of iterations. In Sect. 4 we discuss recent improvements to OOMPH- LIB's FSI preconditioner and demonstrate its excellent performance in a number of test problems solved using the library's monolithic Newton solver.

2 Problem formulation and solution

2.1 OOMPH- LIB's overall design

OOMPH- LIB's design is based on a (finite-)element-like framework in which the system of nonlinear algebraic equations arising from the fully coupled discretisation of multi-physics problems is generated using an element-by-element assembly procedure. Typically, time-derivatives are treated implicitly and for the unsteady simulations described in this paper a second-order, backward-difference method (BDF2) is used for the unsteady terms in the fluid equations, and a Newmark method is used for those in the solid equations. The library provides a large number of single-physics elements that are easily (re-)used in multi-physics problems, via a combination of inheritance and template (generic) programming. In addition, the library provides a variety of node-update strategies to accommodate moving boundaries. In all the test problems below, an algebraic node-update strategy was used to adjust the nodal positions within the fluid domain in response to changes in its (solid) boundaries. The key feature of this strategy is that the position of each node in the fluid domain depends only on the positions of a small number of solid nodes, resulting in fast mesh updates during the assembly of the fully coupled system. Furthermore, the approach leads to sparse shape-derivative matrices whose entries are computed automatically by finite-differencing. Further details can be found in Ref. [3] and on the OOMPH-LIB webpages. Also present in the library are numerous high-level helper functions that facilitate the specification of multi-physics interactions; for instance, functions that automatically determine the (fluid mechanics) degrees of freedom that affect the fluid load on the solid. It is, therefore, straightforward to combine two (or more) single-physics problems to create a monolithically coupled multi-physics problem.

Once a monolithic discretisation has been specified, the coupled problem can be solved by OOMPH- LIB's Newton solver, acting on the fully coupled system of nonlinear algebraic equations. A variety of direct and iterative linear solvers, together with appropriate preconditioners, are provided for the solution of the linear systems that arise during the Newton iteration.

Alternatively, a segregated solution strategy may be employed: OOMPH- LIB's segregated FSI solver starts by "pinning" the degrees of freedom associated with the solid mechanics problem and modifies the assembly procedure to include only those elements associated with the fluids problem. Thus, the Newton solver will solve the equations governing the fluid motion for the current, "frozen", wall shape. Next, the original boundary conditions for the solid problem are re-assigned, the fluid degrees of freedom are "pinned" and the assembly procedure is restricted to the solid elements. After these changes, the Newton solver computes a new wall shape for the given flow field. If desired, different linear solvers/preconditioners can be employed for the fluid and solid solves, allowing the (re-)use of optimal solution methodologies for the solution of the sub-problems. The basic fixed-point iteration can be augmented by constant or adaptive under-relaxation (the latter based on Irons & Tuck's convergence acceleration technique for vector sequences [4]), and predictors for the wall displacement at the next timestep can be employed in time-dependent simulations. Numerical experiments showed that for problems with strong FSI the convergence of the solid sub-solves within the segregated solution strategy were dramatically improved by updating the fluid mesh (and hence the applied viscous stresses) after each linear solve. The algebraic node-update procedure allows mesh updates to be performed very quickly and so this step was performed by default as it had very little effect on the timings but significantly improved the robustness of the fixed-point iteration.

Because OOMPH- LIB's monolithic and segregated solvers are implemented within the same overall framework, it is possible to perform a direct comparison between the two approaches.

2.2 OOMPH- LIB's fluid and solid elements and their interaction

Within OOMPH- LIB it is generally assumed that all lengths and coordinates have been non-dimensionalised on a problem-specific lengthscale $\mathcal{L}$, while time is non-dimensionalised on some reference timescale $\mathcal{T}$. Assuming that the fluid velocities are non-dimensionalised on a representative velocity $\mathcal{U}$, the dimensionless Navier–Stokes equations, which govern the flow of an incompressible Newtonian fluid with density ρ_f and viscosity μ, are then given by[1]

$$Re\left(St\frac{\partial u_i}{\partial t}+u_j\frac{\partial u_i}{\partial x_j}\right)=-\frac{\partial p}{\partial x_i}+\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)$$

$$\text{and}\quad \frac{\partial u_j}{\partial x_j}=0, \tag{1}$$

where $Re=\rho_f\mathcal{U}\mathcal{L}/\mu$ is the Reynolds number, $St=\mathcal{L}/(\mathcal{U}\mathcal{T})$ is the Strouhal number, and the pressure is non-dimensionalised on the viscous scale $\mu\mathcal{U}/\mathcal{L}$. Equation (1), implemented in the Arbitrary–Lagrangian–Eulerian (ALE)

[1] Throughout this paper, Latin indices take the values $i = 1, 2, 3$, Greek indices take the values $\alpha = 1, 2$, and we use the summation convention that repeated indices are summed over all possible values of the index.

form, are the basis of all Navier–Stokes elements in the library.

OOMPH- LIB's geometrically nonlinear Kirchhoff–Love beam and shell elements, used in the test cases in Sects. 3.1, 4.2.2 and 4.2.4, are based on the variational principle

$$\begin{aligned}&\iint \Bigg[(\sigma_0^{\gamma\delta} + E^{\alpha\beta\gamma\delta}\gamma_{\alpha\beta})\,\delta\gamma_{\gamma\delta} \\ &\quad + \frac{1}{12}\left(\frac{h}{\mathcal{L}}\right)^2 E^{\alpha\beta\gamma\delta}\kappa_{\alpha\beta}\,\delta\kappa_{\gamma\delta}\Bigg]\sqrt{a}\,d\xi^1\,d\xi^2 \\ &= \iint \left[\left(\frac{\mathcal{L}}{h}\right)\sqrt{\frac{A}{a}}\,\mathbf{f} - \Lambda^2\frac{\partial^2\mathbf{R}_s}{\partial t^2}\right] \\ &\quad \times\delta\mathbf{R}_s\sqrt{a}\,d\xi^1\,d\xi^2, \end{aligned} \tag{2}$$

which describes the large-displacements of an incrementally linearly elastic shell of thickness h and density ρ_s, Young's modulus E and Poisson ratio ν. Here, $\mathbf{R}_s$ is the position vector to the shell's deformed midplane, parameterised by the Lagrangian coordinates ξ^1 and ξ^2; a beam is understood to be the (obvious) restriction to the case when the midplane is parameterised by a single Lagrangian coordinate. The stresses, the fourth-order elasticity tensor $E^{\alpha\beta\gamma\delta}$, and the load vector $\mathbf{f}$ are non-dimensionalised by the structure's effective Young's modulus, $E_{eff} = E/(1-\nu^2)$. The (second Piola–Kirchhoff) pre-stress is represented by $\sigma_0^{\alpha\beta}$, and $\gamma_{\alpha\beta}$ and $\kappa_{\alpha\beta}$ are the midplane strain and bending tensors. $\sqrt{A}\,d\xi_1 d\xi_2$ and $\sqrt{a}\,d\xi_1 d\xi_2$ represent infinitesimal area elements of the shell's deformed and undeformed midplanes, respectively. The parameter $\Lambda = (\mathcal{L}/\mathcal{T})\sqrt{\rho_s/E_{eff}}$ is the ratio of the structure's natural timescale (for free in-plane vibrations) to the timescale $\mathcal{T}$ used in the non-dimensionalisation of the equations.

OOMPH- LIB's general large-displacement elasticity elements (available in displacement and pressure-displacement forms for compressible and (near-)incompressible behaviour) are based on the variational principle

$$\int \sigma^{ij}\,\delta\gamma_{ij}\,dv = \int\left(\mathbf{b} - \Lambda^2\frac{\partial^2\mathbf{R}_s}{\partial t^2}\right)\cdot\delta\mathbf{R}_s\,dv + \oint\mathbf{f}\cdot\delta\mathbf{R}_s\,dA, \tag{3}$$

where the two integrals are performed over the undeformed reference volume and over the deformed surface of the body, respectively. The second Piola–Kirchhoff stress tensor, σ^{ij}, is determined as a function of the Green strain tensor γ_{ij} via a user-specified constitutive equation. σ^{ij} is assumed to be non-dimensionalised on some characteristic stiffness parameter, $\mathcal{S}$, such as Young's modulus E, which is also used for the consistent non-dimensionalisations of the body force $\mathbf{b}$ and the surface traction $\mathbf{f}$; and the timescale ratio is now given by $\Lambda = (\mathcal{L}/\mathcal{T})\sqrt{\rho_s/\mathcal{S}}$.

The solid displacements affect the fluid via the induced changes in the domain geometry and via the no-slip condition

$$\mathbf{u} = St\,\frac{\partial\mathbf{R}_s}{\partial t} \quad \text{on fluid–solid interfaces.} \tag{4}$$

The Cartesian components of the traction that the Newtonian fluid exerts onto the solid (on the solid stress scale) is given by

$$f_i^{[\mathrm{FSI}]} = Q\left(-p\delta_{ij} + \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right)N_j, \tag{5}$$

where the N_j are the Cartesian components of the outer unit normal on the deformed solid (pointing into the fluid) and Q is the ratio of the stress scales used in the non-dimensionalisation of the solid and fluid equations. The parameter Q indicates the strength of the FSI: as $Q \to 0$, the fluid stresses acting on the structure become negligible, effectively decoupling the fluid and solid problems. We stress, however, that this statement is to be understood in an asymptotic sense. A "small" but finite value of Q does not necessarily imply that FSI can be neglected, particularly if the stiffness parameter used to non-dimensionalise the solid stresses does not provide a good indication of the structure's stiffness. For instance, a small value of Q in a thin-shell problem indicates that the fluid stresses are small relative to the shell's extensional stiffness. If the shell deforms in a bending mode (in which it is much more flexible) the fluid stresses may still induce large deformations at small Q.

3 Comparing monolithic and segregated solvers

3.1 The test problem: flow in a collapsible channel

We shall explore the relative performance of segregated and monolithic solvers using the well-studied FSI problem of flow in a collapsible channel. Incompressible, Newtonian fluid is driven through a 2D channel whose width is used as the lengthscale $\mathcal{L}$. A Poiseuille (parabolic) velocity profile of mean velocity $\mathcal{U}$ is imposed at the inflow boundary and the outflow velocity is set to be parallel and axially traction free. A section of the upper channel wall is elastic and modelled as a pre-stressed Kirchhoff–Love beam loaded by an external pressure, p_{ext}, and the fluid traction (5). The system is known to develop large-displacement, self-excited oscillations (see, e.g., Ref. [5] and Fig. 2), provided that Re and Q are sufficiently large.

We employed OOMPH- LIB's `QTaylorHood` (Q_2Q_1) Navier–Stokes elements to discretise the ALE form of the unsteady Navier–Stokes equations and used the Kirchhoff–Love `HermiteBeamElements` to discretise the flexible part of the channel wall. Displacement control (prescribing

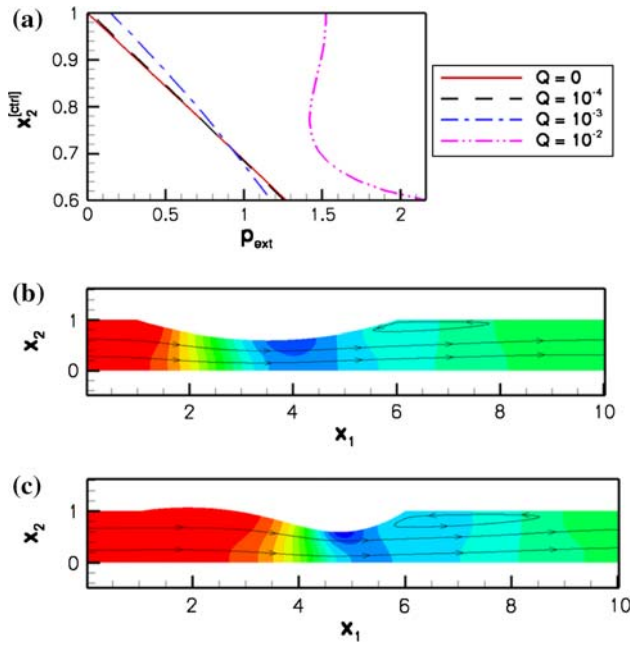

Fig. 1 **a** Load–displacement diagram: the vertical position, $x_2^{[\mathrm{ctrl}]}$, of a control point on the elastic wall (located at 50, 50, 60 and 70% of its length for $Q = 0,\ 10^{-4}, 10^{-3}$ and 10^{-2}, respectively) as a function of the external pressure, p_{ext}. **b**, **c** Steady flow fields (streamlines and pressure contours in the vicinity of the elastic segment) at $Re = 500$, for **b** $Q = 10^{-4}$ and **c** $Q = 10^{-2}$

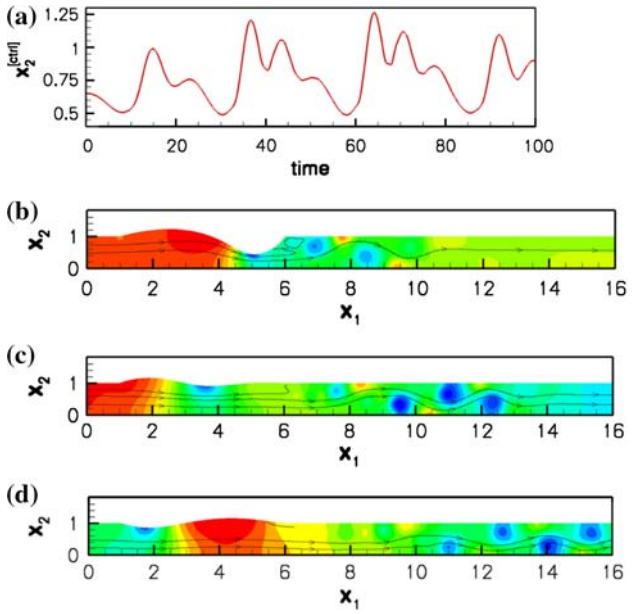

Fig. 2 **a** Time-trace of the vertical position of the control point on the wall (located at 70% of its length) during a self-excited oscillation at $Re = 500$, $St = 1$ and $Q = 10^{-2}$. **b–d** Snapshots of the unsteady flow fields (instantaneous streamlines and pressure contours) at **b** $t = 32.4$, **b** $t = 34.9$ and **c** $t = 37.4$

the vertical displacement of a control point near the anticipated point of strongest collapse and solving for the unknown external pressure required to achieve that deformation) was employed to handle the system's possible snap-through behaviour in steady simulations; see Fig. 1a.

3.2 Results of the simulations

Simulations were performed for a channel in which a massless ($\Lambda = 0$) flexible wall segment of thickness $h/\mathcal{L} = 1/20$ and length $L = 5$ was subjected to an axial prestress of $\sigma_0 = 10^3$ and mounted on two rigid channels of length $L_{\mathrm{up}} = 1$ and $L_{\mathrm{down}} = 10$.

In Fig. 1a we illustrate the system's load–displacement characteristics by plotting the vertical position of a control point on the elastic section of the wall, $x_2^{[\mathrm{ctrl}]}$, as a function of the non-dimensional external pressure, p_{ext}. For small values of the FSI parameter Q, the fluid traction is a small fraction of the load on the wall and an increase in p_{ext} leads to an approximately proportional increase in the wall deflection; the fluid reacts passively to the changes in the wall geometry. For instance, in Fig. 1b the wall deformation is virtually symmetric, with the point of strongest collapse located at the centre of the elastic segment, indicating that the load on the wall is dominated by the spatially constant external pressure. As Q increases, the fluid traction noticeably affects the load–displacement curve. At larger values of Q, a larger external pressure is required to keep the wall in its (approximately) undeformed position because p_{ext} must balance the increasingly large fluid pressure in the elastic section, and so the load–displacement curves shift to the right. The increase in fluid pressure is a consequence of the fact that the outflow boundary conditions impose a zero fluid pressure at the end of the rigid downstream section, but the viscous pressure drop through that section increases with Q. Another consequence of the increased viscous pressure drop is that the point of strongest collapse moves downstream and so the control point is varied accordingly. In addition, the Bernoulli effect causes a reduction in fluid pressure in the most strongly collapsed part of the channel, locally increasing the compressive load on the wall. Hence, at larger values of Q a smaller increase in external pressure is required to collapse the channel to a given degree. At $Q = 10^{-2}$ this destabilising effect is so strong that limit points develop in the load-displacement curve, indicating that, at sufficiently large p_{ext}, the wall "snaps through" into a strongly deformed equilibrium configuration. The corresponding plot of the wall shape in Fig. 1c illustrates that the wall deformation is now strongly affected by fluid pressure distribution.

Figure 2a shows the time-trace of the wall displacement in an unsteady simulation at $Re = 500,\ St = 1$ and $Q = 10^{-2}$. For this simulation the steady solution for $p_{\mathrm{ext}} = 1.68$ was used as the initial condition for a time-dependent simulation in which p_{ext} was set to 2.51 at $t = 0$. The time-trace and

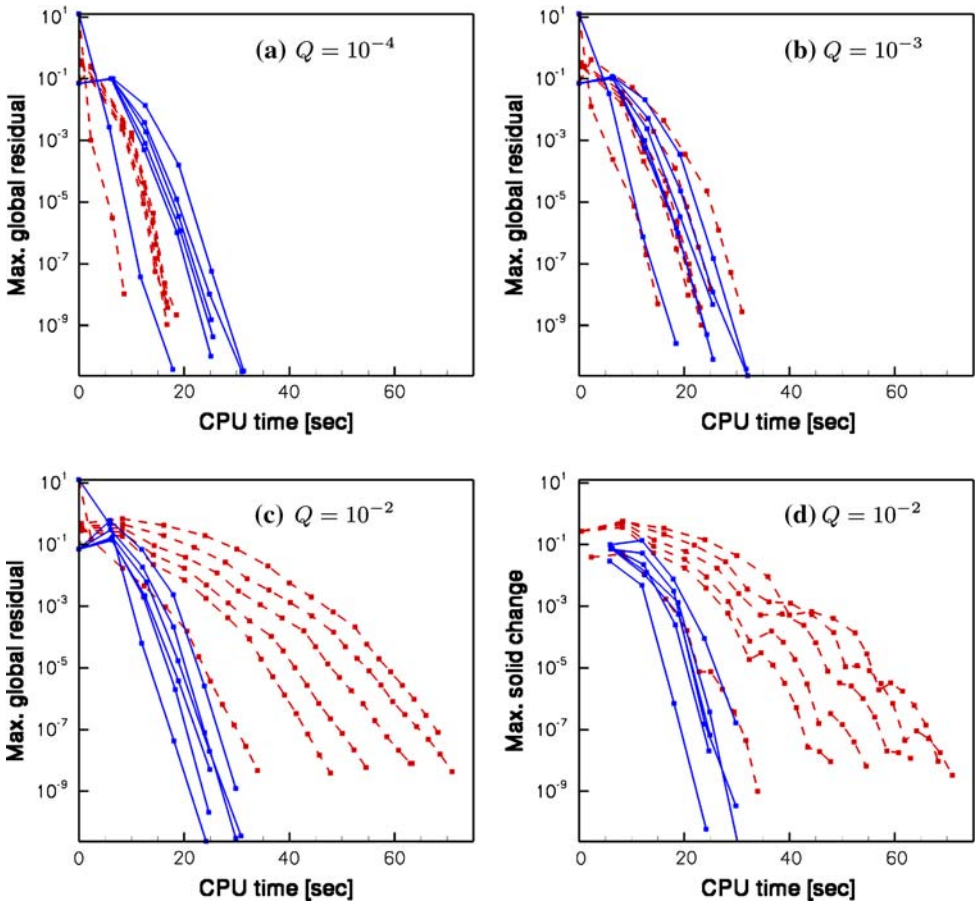

Fig. 3 Convergence histories of the segregated (*dashed lines*) and monolithic (*solid lines*) solvers for a steady computation at $Re = 500$ and various values of the interaction parameter Q. In **a-c** the convergence is characterised by the maximum residual of the coupled equations; in **d** convergence is assessed in terms of the maximum change in the solid variables between two successive iterations

the snapshots of the flow field show that, following the initial perturbation, the system performs sustained, large-amplitude self-excited oscillations during which complex vortical flow structures develop downstream of the oscillating wall segment.

3.3 The relative performance of the segregated and monolithic solvers

Figures 3a–c show the convergence histories (maximum residual of the coupled equations vs. CPU time on a 3.60 GHz Intel Xeon processor with 2 GB of memory) of the monolithic and segregated solvers for a steady computation at $Re = 500$, and for three different values of the interaction parameter Q. The six curves in each figure represent the convergence histories during six nonlinear solves performed in the course of a parameter study during which the channel's maximum collapse was increased from 0 to 35% of its width. For simplicity, all linear systems were solved with OOMPH- LIB's default linear solver, SUPERLU [6], a sparse direct solver which is efficient for this moderately sized 2D problem (20,390 unknowns); see Tables 1 and 3.

For all cases considered, both nonlinear solvers converge in a moderate number of iterations, with roughly comparable CPU times, irrespective of the convergence criterion (maximum global residual or maximum change in the position of the solid nodes between two successive nonlinear iterations; see Figs. 3c, d). The variable CPU time increments in the convergence histories for the segregated solver indicate that the number of Newton iterations required by the two sub-solvers tends to decrease as the Picard iteration proceeds.

Table 1 Average CPU times (in seconds) for the monolithic Newton solver applied to the steady collapsible channel problem with $Re = 500$ and $Q = 10^{-2}$, for different linear solvers/preconditioners

N_{DOF}	SuperLU	GMRES & $\mathcal{P}_1$	GMRES & $\mathcal{P}_2$
8987	9.7	12.2(12.4)	11.8(11.0)
20391	27.7	27.1(31.2)	25.4(29.2)
36403	65.8	52.4(61.5)	49.7(60.2)
57023	130.5	87.7(116.2)	83.6(110.8)
82251	230.7	153.5(213.7)	147.3(205.7)
112087	394.4	227.6(324.6)	218.8(329.4)
146531	–	332.5(470.5)	321.2(499.7)
185583	–	459.8(653.2)	432.7(729.5)

Numbers in brackets are for the original version of the preconditioner used in Ref. [7], i.e. without the scaling factor $\widehat{\mathbf{Q}}$

Table 2 Average GMRES iteration counts for the steady collapsible channel problem with $Re = 500$ and $Q = 10^{-2}$, for different preconditioners

N_{DOF}	GMRES & $\mathcal{P}_1$	GMRES & $\mathcal{P}_2$
8987	71.5 (70.0)	56.5 (58.9)
20391	65.5 (66.1)	56.5 (59.0)
36403	59.8 (72.6)	51.8 (64.4)
57023	52.6 (70.0)	45.6 (67.1)
82251	47.3 (76.1)	41.1 (71.4)
112087	43.4 (80.7)	38.0 (77.1)
146531	40.2 (86.2)	35.2 (84.0)
185583	38.4 (95.0)	33.9 (88.5)

Numbers in brackets as in Table 1

Table 3 Average CPU times (in seconds) for the monolithic Newton solver applied to the unsteady collapsible channel problem for $Re = 500$, $St = 1$ and $Q = 10^{-2}$, with different linear solvers/preconditioners

N_{DOF}	SuperLU	GMRES & $\mathcal{P}_1$	GMRES & $\mathcal{P}_2$
8986	5.3	5.7(5.7)	5.6(5.5)
20390	17.8	14.6(14.8)	14.4(14.2)
36402	38.2	28.9(29.3)	28.4(28.7)
57022	72.0	51.7(52.1)	49.9(50.5)
82250	120.3	91.0(92.8)	88.9(91.1)
112086	197.7	137.3(140.5)	133.9(138.1)
146530	–	197.3(202.8)	193.6(199.2)
185582	–	277.8(286.4)	273.4(282.7)

Numbers in brackets as in Table 1

As expected, the monolithic solver performs better when the coupling is strong (large Q) and the segregated solver performs best for weak coupling (small Q). Nonetheless, for the steady solutions computed here, both methods are competitive. In all cases with non-negligible FSI the monolithic solver tends to perform better than the segregated approach—recall that Fig. 1a shows that for $Q = 10^{-4}$ the fluid load has a minimal effect on the system's load–displacement characteristics. Qualitatively similar behaviour was found at different Reynolds numbers and at different spatial resolutions.

During the steady simulations under-relaxation was neither required nor beneficial and the most rapid convergence was achieved in a simple, unmodified Picard iteration. Conversely, in unsteady problems the segregated solver was found to diverge rapidly unless very strong under-relaxation was applied. Under-relaxation parameters of less than 10^{-3} were required to stabilise the segregated solution procedure.

Figure 4b shows the CPU times for the different nonlinear solvers during the course of an unsteady simulation at $Re = 500$, $St = 1$ and $Q = 10^{-2}$. Since the impulsive start

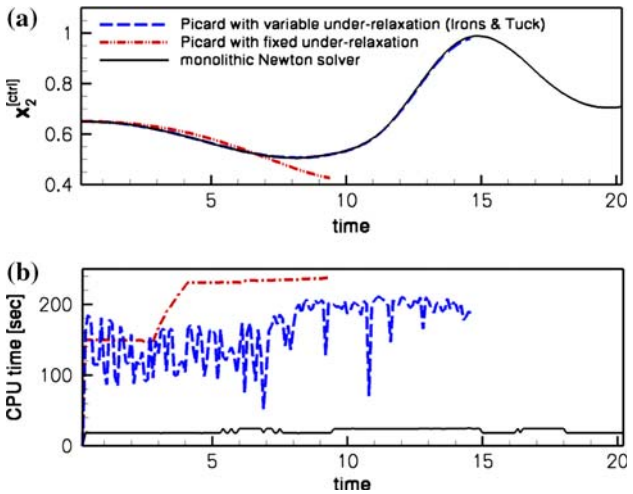

Fig. 4 **a** Time-trace of the vertical position of the control point on the wall (located at 70% of its length) during a time-dependent simulation with $Re = 500$, $St = 1$ and $Q = 10^{-2}$, computed with the monolithic and segregated solvers. **b** CPU times required by the various nonlinear solvers

at the beginning of the simulation is likely to cause particular problems for a segregated solution procedure, we performed the first few timesteps with the monolithic Newton solver and then restarted the simulation, either with the monolithic or the segregated solver. If a fixed under-relaxation parameter of $\omega = 10^{-4}$ is used throughout the Picard iteration, the convergence rates are so slow that the segregated solver never converges within (the maximum number of) 50 iterations. The effect is seen is Fig. 4a in which the solution drifts away from the "exact" solution (computed by the monolithic Newton solver) until, at $t \approx 9.5$, the Newton iteration fails to converge during the solid solve. At larger values of the under-relaxation parameter the segregated solver fails even earlier. The adaptive adjustment of the under-relaxation parameter by Irons & Tuck's convergence acceleration procedure [4] leads to significant improvements in the convergence rates and hence a much more accurate solution. However, even in this case the Picard iteration ultimately fails near $t \approx 15$, again because the Newton solver diverges during the solid solve. Furthermore, the CPU times required by the segregated solver are now significantly larger than those required by the monolithic solver which continues to converge in approximately the same CPU times observed in the steady computations.

These results confirm the widely acknowledged fact that monolithic solvers tend to be more robust than their segregated counterparts. More importantly, we demonstrated that even in problems with weak FSI in which segregated solvers are expected to perform well, the monolithic and segregated solvers converge in comparable CPU times, dispelling the myth that the former are necessarily more expensive.

4 Preconditioners for monolithic FSI solvers

4.1 OOMPH- LIB's FSI preconditioning strategy

We still have to address the fact that the use of direct solvers for the solution of the linear systems that arise during the solution of the monolithically coupled nonlinear algebraic equations by Newton's method ceases to be feasible for larger problems, particularly in 3D. We therefore turn our attention to the solution of the linear systems by iterative, Krylov subspace, methods, such as GMRES, which require the provision of efficient preconditioners. As discussed in the introduction, the ability to re-use highly optimised solvers and preconditioners for the solution of the individual sub-problems is often regarded as one of the key advantages of the segregated approach. We shall now demonstrate that a block-triangular FSI preconditioner allows the re-use of existing single-physics solver/preconditioning strategies within the framework of a monolithic solver.

For this purpose we write the discretised fluid and solid equations as $\mathcal{F}(\mathbf{u}, \mathbf{p}; \mathbf{s}) = \mathbf{0}$, and $\mathcal{S}(\mathbf{s}; \mathbf{u}, \mathbf{p}) = \mathbf{0}$, where $\mathbf{u}$, $\mathbf{p}$ and $\mathbf{s}$ are the vectors containing discrete fluid velocities, fluid pressures and solid displacements, respectively. The assumed existence of "optimal" solvers for the uncoupled fluid and solid sub-problems implies that we can efficiently solve the linear systems

$$\begin{pmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta\mathbf{u} \\ \delta\mathbf{p} \end{pmatrix} = - \begin{pmatrix} \mathbf{r}_u \\ \mathbf{r}_p \end{pmatrix}, \tag{6}$$

which arise during the Newton iteration for the fluid equations, while keeping the solid unknowns $\mathbf{s}$ fixed. Here the matrix $\mathbf{F}$ arises from the discretisation of the unsteady advection-diffusion parts of the fluid momentum equations while $\mathbf{G}$ and $\mathbf{D}$ represent discrete gradient and divergence operators.

Similarly, we assume that there exist efficient solvers for the linear systems that arise during the Newton iteration for the solid equations while the fluid velocities and pressures are held fixed,

$$\mathbf{S}\,\delta\mathbf{s} = -\mathbf{r}_s, \tag{7}$$

where $\mathbf{S}$ is the solid's tangent stiffness matrix.

The challenge is to re-use the existing optimal solvers in the context of the linear systems

$$\left(\begin{array}{cc|c} \mathbf{F} & \mathbf{G} & \mathbf{C}_{us} \\ \mathbf{D} & \mathbf{0} & \mathbf{C}_{ps} \\ \hline \mathbf{C}_{su} & \mathbf{C}_{sp} & \mathbf{S} \end{array}\right) \left(\begin{array}{c} \delta\mathbf{u} \\ \delta\mathbf{p} \\ \hline \delta\mathbf{s} \end{array}\right) = - \left(\begin{array}{c} \mathbf{r}_u \\ \mathbf{r}_p \\ \hline \mathbf{r}_s \end{array}\right) \tag{8}$$

that must be solved during the Newton iteration of the monolithically coupled equations. Here the off-diagonal matrix blocks $\mathbf{C}_{**}$ arise from the interaction between fluid and solid equations: $\mathbf{C}_{us}$ and $\mathbf{C}_{ps}$ contain the so-called "shape derivatives"—the derivatives of the Navier–Stokes residuals with respect to the solid displacements that affect the nodal positions in the fluid mesh. Similarly, $\mathbf{C}_{su}$ and $\mathbf{C}_{sp}$ contain the derivatives of the solid residuals with respect to the fluid variables; this interaction arises through the fluid loading on the wall via Eq. (5). As discussed in Sect. 2.1, OOMPH- LIB's algebraic node-update strategy ensures that the interaction matrices are very sparse. The maximum fill level for the examples presented in this paper is about 3% and such (relatively) large values only arose in computations with very coarse meshes; the much finer meshes used in typical production runs resulted in much sparser matrices.

We showed in Ref. [7] that the use of block-triangular approximations to the global Jacobian matrix, obtained by neglecting the fluid–solid or solid–fluid interaction blocks,

$$\mathcal{P}_1 = \left(\begin{array}{cc|c} \mathbf{F} & \mathbf{G} & \mathbf{0} \\ \mathbf{D} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{C}_{su} & \mathbf{C}_{sp} & \mathbf{S} \end{array}\right) \quad \text{and} \quad \mathcal{P}_2 = \left(\begin{array}{cc|c} \mathbf{F} & \mathbf{G} & \mathbf{C}_{us} \\ \mathbf{D} & \mathbf{0} & \mathbf{C}_{ps} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{S} \end{array}\right) \tag{9}$$

in the Newton method seriously degrades its performance, resulting in the loss of its quadratic convergence and thus one of its the most attractive features. However, the block-triangular approximations were shown to be excellent preconditioners for the solution of the linear system (8) by Krylov subspace methods. Because of their block-triangular structure each application of the preconditioners involves linear solves with each of the two single-physics systems (6) and (7), and matrix–vector products with the retained interaction matrices.

The current implementation of this preconditioner within OOMPH- LIB uses Elman, Silvester & Wathen's "least squares commutator" (LSC) preconditioner [9] to approximately solve the fluid system (6). Compared to the original implementation of the FSI preconditioner, discussed in Ref. [7], this version of the Navier–Stokes preconditioner incorporates additional scaling operations which result in a significant improvement in its performance.

Each application of the LSC preconditioner to a vector $(\mathbf{y}_u, \mathbf{y}_p)$ requires the solution of the linear system

$$\begin{pmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{0} & -\widetilde{\mathbf{M}}_s \end{pmatrix} \begin{pmatrix} \mathbf{x}_u \\ \mathbf{x}_p \end{pmatrix} = \begin{pmatrix} \mathbf{y}_u \\ \mathbf{y}_p \end{pmatrix}, \tag{10}$$

for $(\mathbf{x}_u, \mathbf{x}_p)$. Here $\widetilde{\mathbf{M}}_s$ is an approximation to the pressure Schur complement $\mathbf{M}_s = \mathbf{D}\mathbf{F}^{-1}\mathbf{G}$. The solution is performed in two stages. First we (formally) solve the second row of (10) for $\mathbf{x}_p$ via

$$\mathbf{x}_p = -\widetilde{\mathbf{M}}_s^{-1}\mathbf{y}_p. \tag{11}$$

Given $\mathbf{x}_p$, we then obtain $\mathbf{x}_u$ from the linear system

$$\mathbf{F}\mathbf{x}_u = \mathbf{y}_u - \mathbf{G}\mathbf{x}_p. \tag{12}$$

The action of the inverse pressure Schur complement $\widetilde{\mathbf{M}}_s^{-1}$ in (11) is approximated by

$$\mathbf{x}_p = -\left(\mathbf{D}\widehat{\mathbf{Q}}^{-1}\mathbf{G}\right)^{-1}\left(\mathbf{D}\widehat{\mathbf{Q}}^{-1}\mathbf{F}\widehat{\mathbf{Q}}^{-1}\mathbf{G}\right)\left(\mathbf{D}\widehat{\mathbf{Q}}^{-1}\mathbf{G}\right)^{-1}\mathbf{y}_p, \tag{13}$$

where $\widehat{\mathbf{Q}}$ is the diagonal of the velocity mass matrix which is easy to compute and cheap to invert. The evaluation of this expression involves two linear solves with the matrix

$$\mathbf{P} = \left(\mathbf{D}\widehat{\mathbf{Q}}^{-1}\mathbf{G}\right) \tag{14}$$

which has the character of a discrete pressure Poisson operator. We also have to evaluate matrix–vector products with the matrix

$$\mathbf{E} = \mathbf{D}\widehat{\mathbf{Q}}^{-1}\mathbf{F}\widehat{\mathbf{Q}}^{-1}\mathbf{G}. \tag{15}$$

OOMPH- LIB allows the use of different (approximate) linear solvers for the solution of the various linear systems. Linear systems involving the scaled pressure Poisson matrix (14) can be solved efficiently with algebraic (or geometric) multigrid. Within OOMPH- LIB we typically employ the algebraic multigrid solvers from the HYPRE [11] or TRILINOS [12] libraries and perform a single multigrid cycle to obtain an approximate solution. We demonstrated in Ref. [7] that the system (12) can be solved efficiently by geometric multigrid with grid-dependent stabilisation. Geometric multigrid is currently under development in OOMPH- LIB, and in the studies presented below the linear system (12) was solved with SUPERLU.

4.2 Performance of the FSI preconditioner

4.2.1 The collapsible channel problem

Tables 1–4 illustrate the performance of the FSI preconditioner for the collapsible channel problem of Sect. 3. Tables 1 and 3 compare the average CPU times for the monolithic Newton solver when the linear system (8) is solved by SUPERLU or by GMRES (with a convergence tolerance of 10^{-6}), preconditioned with the two block-triangular FSI preconditioners $\mathcal{P}_1$ and $\mathcal{P}_2$. While the direct solution of the linear systems with SUPERLU is competitive at moderate spatial resolutions, the CPU times (and memory requirements) increase rapidly with the number of degrees of freedom; for the two finest discretisation considered, the total memory required by SUPERLU exceeded the 2 GB memory available on our computer. Conversely, the CPU (and memory) requirements for the iterative linear solvers increase much more slowly with the number of degrees of freedom, allowing the solution of much larger problems.

Table 4 Average GMRES iteration counts for the unsteady collapsible channel problem with $Re = 500$, $St = 1$ and $Q = 10^{-2}$, for different preconditioners

N_{DOF}	GMRES & $\mathcal{P}_1$	GMRES & $\mathcal{P}_2$
8986	22.8 (23.5)	19.9 (20.3)
20390	25.5 (26.7)	22.2 (22.9)
36402	25.7 (28.2)	22.4 (24.0)
57022	25.6 (29.0)	22.4 (25.1)
82250	25.4 (29.0)	22.1 (24.8)
112086	25.2 (28.9)	21.9 (25.1)
146530	24.1 (28.6)	21.1 (25.4)
185582	23.3 (27.7)	20.5 (25.8)

Numbers in brackets as in Table 1

A comparison between the CPU times and iteration counts for the old and new versions of the preconditioner shows that the inclusion of the scaling factor $\widehat{\mathbf{Q}}$ into Eqs. (13), (14) and (15) leads to a dramatic improvement in its performance, particularly for steady problems where an increase in the number of degrees of freedom now leads to a rapid reduction in the average iteration counts. For unsteady problems the old version of the preconditioner only suffered from a modest deterioration of the GMRES convergence rates with an increase in the number of degrees of freedom. The inclusion of the scaling factor makes the convergence rates essentially mesh independent.

A detailed analysis of the timings confirms that the present implementation is not completely optimal, i.e. the CPU time does not scale linearly with the number of degrees of freedom. This is because SUPERLU is still used for the solution of the linear systems (7) and (12). We expect to gain further improvements and near-optimal scalings by replacing the direct solver for (12) by a stabilised geometric multigrid solver, as in Ref. [7].

4.2.2 An immersed leaflet in pulsatile flow

The next test problem involves a fully-immersed beam structure and explores the performance of the FSI preconditioner in simulations with spatial adaptivity. The problem is illustrated in Fig. 5. A rigid-walled 2D channel whose dimensional width $\mathcal{L}$ is used to non-dimensionalise all lengths, is partially occluded by a thin-walled elastic leaflet of undeformed length $\frac{1}{2}\mathcal{L}$, represented by a massless Kirchhoff–Love beam of thickness $h/\mathcal{L} = 1/20$. A pulsatile Poiseuille flow whose mean velocity fluctuates between $\mathcal{U}$ and $2\mathcal{U}$ is imposed at the upstream and of the channel. We scale the

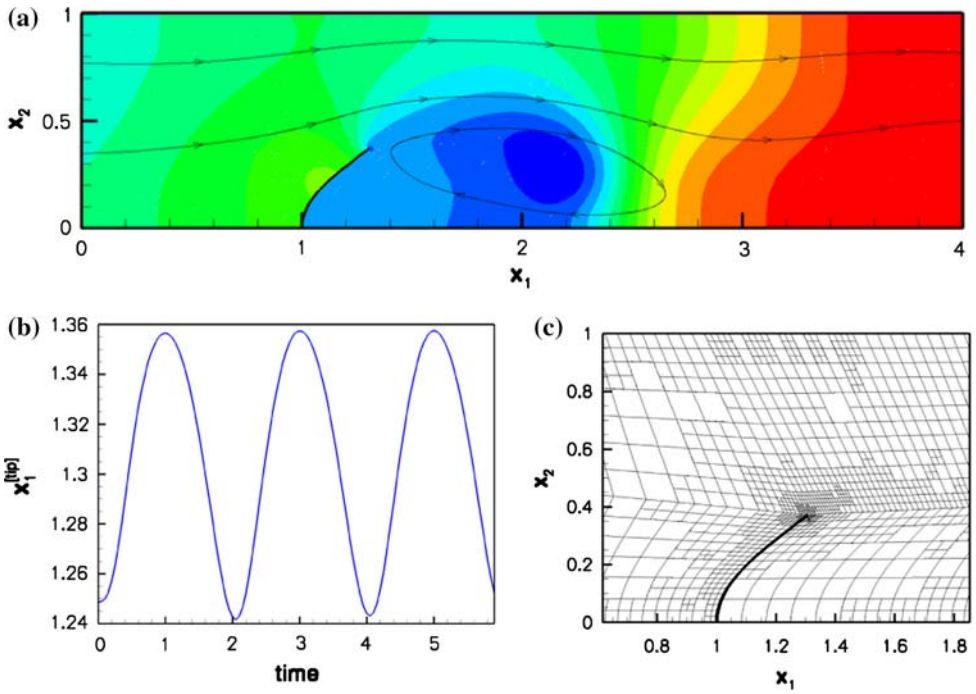

Fig. 5 Pulsatile flow in a 2D channel that is partially occluded by a thin-walled elastic leaflet. The Reynolds number associated with the minimum mean velocity of the prescribed pulsatile inflow is $Re = 200$. **a** Snapshot of the flow field (instantaneous streamlines and pressure contours) at $t = 1.55$. **b** Time-trace of the horizontal position of the leaflet tip. **c** Detail of the adaptively refined mesh in the vicinity of the leaflet

velocity on $\mathcal{U}$ and impose

$$\mathbf{u} = 6\, x_2(1 - x_2)\left(1 + \frac{1}{2}\left(1 - \cos\left(\pi t\right)\right)\right)\mathbf{e}_1 \tag{16}$$

at $x_1 = 0$, while parallel, axially traction-free outflow is imposed at the outflow. Time is non-dimensionalised on the flow's intrinsic timescale $\mathcal{T} = \mathcal{L}/\mathcal{U}$ implying that $St = 1$. The flow is started from the steady solution for the inflow profile (16) evaluated at $t = 0$.

Figure 5a,b show a snapshot of the flow field and the time-trace of the horizontal position of the leaflet's tip for a Reynolds number of $Re = 200$ and an interaction parameter of $Q = 10^{-6}$. Following the decay of initial transients, the leaflet performs periodic large-amplitude oscillations with the period of the pulsating inflow. Large velocity gradients develop at the front of the leaflet and in the shear layer that emanates from its tip and separates the recirculating flow region behind the leaflet from the main flow. Figure 5c illustrates the non-uniform mesh refinement and shows the improved resolution in the high-shear regions, particularly near the leaflet's tip where the pressure is singular. The mesh was continuously adapted throughout the simulation and contained an average of about 32,000 degrees of freedom. This is a fraction of the 1,324,343 degrees of freedom that would be required to achieve the same local resolution via uniform mesh refinement. On average the monolithic Newton solver computed a fully converged solution in 47.8 s, with an average GMRES iteration count (over all linear solves) of 46.5 when $\mathcal{P}_1$ was used as the preconditioner.

4.2.3 Flow around a cylinder with a "flag"

Next we consider Turek & Hron's FSI benchmark problem of the flow around an elastic "flag" attached to a circular cylinder that is mounted slightly asymmetrically in a 2D channel. The full problem specification may be found in Ref. [10]. We non-dimensionalise all lengths on the cylinder's diameter and use a timescale of $\mathcal{T}$=1 sec to facilitate comparisons against the (dimensional) results presented in Ref. [10]. This test involves FSI with genuine solid (rather than beam) elements, and includes non-zero wall mass.

Figure 6 shows a snapshot of the flow field and the time-trace of the vertical position of the flag's tip for parameter values that correspond to those of Turek & Hron's test case "FSI2": $Re = 100$, $St = 0.1$ and $Q = 7.143 \times 10^{-6}$. The ratio of solid and fluid densities is $\rho_s/\rho_f = 10$, resulting in a timescale ratio of $\Lambda^2 = 7.143 \times 10^{-5}$.

Following an initial transient period, the system settles into a large-amplitude self-excited oscillation during which the oscillating "flag" generates a regular vortex pattern that is advected along the channel. The computation was performed with spatial adaptivity for the fluid and solid meshes, resulting in an average of approximately 65,000 degrees of freedom. A relatively large timestep of $\Delta t = 0.01$—corresponding to about 50 timesteps per period of the oscillation—was used. With this discretisation the system settled into oscillations with a period of ≈ 0.52 and an amplitude of the vertical tip displacement of 0.01 ± 0.83. GMRES was preconditioned with $\mathcal{P}_1$ and required an average of 44.2 iterations to converge, resulting in an average

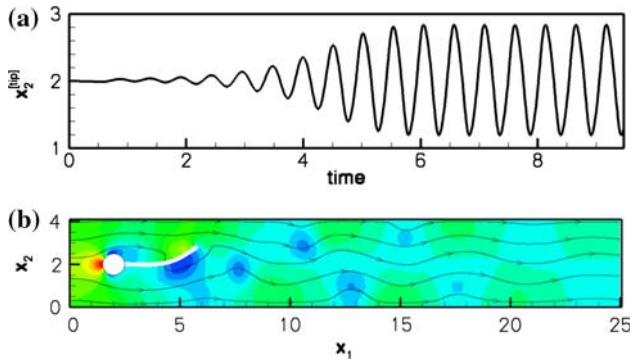

Fig. 6 Flow past a cylinder with an attached flag for the parameter values of test case "FSI2" in Ref. [10]. **a** Time-trace of the x_2-coordinate of the flag's tip. **b** Snapshot of the flow field (instantaneous streamlines and pressure contours) at $t = 6.04$

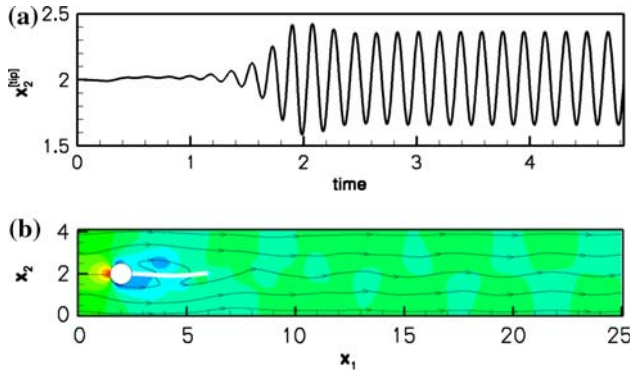

Fig. 7 Flow past a cylinder with an attached flag for the parameter values of test case "FSI3" in Ref. [10]. **a** Time-trace of the x_2-coordinate of the flag's tip. **b** Snapshot of the flow field (instantaneous streamlines and pressure contours) at $t = 3.615$

time of 75.3 s for the monolithic Newton solver at each timestep.

Figure 7 shows the corresponding results for the test case "FSI3" where fluid and wall densities are equal and $Re = 200$, $St = 0.05$, $Q = 3.571 \times 10^{-6}$ and $\Lambda^2 = 1.7855 \times 10^{-6}$. Compared to the conditions in test case "FSI2", the system performs oscillations of much higher frequency and smaller amplitude. Our computation was performed with a fixed timestep of $\Delta t = 0.005$ and resulted in oscillations with a period of ≈ 0.19 and an amplitude of the vertical tip displacement of $\approx 0.01 \pm 0.36$. The increase in frequency and Reynolds number leads to the development of thinner boundary and shear layers which require a finer spatial resolution, involving an average of 84,000 degrees of freedom. This had very little effect on the GMRES convergence rates. With $\mathcal{P}_1$ used as preconditioner an average of 45.8 iterations were required to obtain a converged solution of the linear systems, while the solution of the nonlinear systems by Newton's method now required an average of 104.4 s.

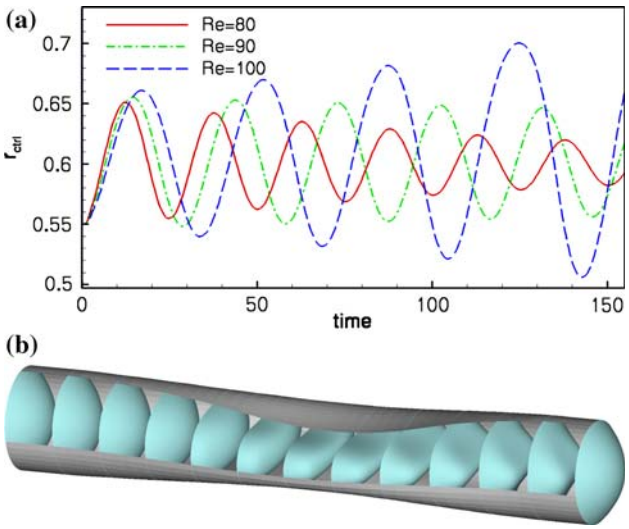

Fig. 8 Unsteady flow through a collapsible tube for $Q = 10^{-7}$, $St = 1$. **a** Time-trace of the radial position of a control point located at 70% of the tube length, at the most strongly collapsed point in the cross section, for various Reynolds numbers. **b** Snapshot of the axial velocity profiles at $t = 181.26$ for the $Re = 100$ case

4.2.4 Self-excited oscillations of a 3D collapsible tube

In the final example we evaluate the performance of the FSI preconditioner in preliminary 3D simulations of unsteady finite-Reynolds number flows in a collapsible tube, modelled as a circular cylindrical shell of radius $\mathcal{L}$, length $10\mathcal{L}$ and wall thickness $h/\mathcal{L} = 1/20$, and a Poisson ratio of $\nu = 0.49$, mounted on two short rigid tubes of length $\mathcal{L}$. We prescribe Poiseuille flow with average velocity $\mathcal{U}$ at the far downstream end and impose parallel, axially traction-free inflow upstream. Figure 8 shows the time-trace, $r_{\text{ctrl}}(t)$, of the radial position of a control point on the tube wall, located towards the downstream end of the tube at 70% of its length, for $Q = 10^{-7}$, $St = 1$ and $\Lambda = 0$ at various Reynolds numbers. In all cases the simulation was started from the steady solution with $r_{\text{ctrl}} = 0.55$, which required different external pressures for each different Reynolds number. At $t = 0$ we reduced p_{ext} to a value corresponding to an equilibrium configuration in which $r_{\text{ctrl}} = 0.6$ and followed the system's evolution. The time-trace shows that the system oscillates about the new equilibrium position. At small Reynolds numbers the oscillations decay but for sufficiently large Reynolds number they grow in amplitude, as observed in physical experiments; see e.g. [8].

The results presented here are preliminary in the sense that the assessment of their mesh and timestep independence has not yet been completed. However, the plot of the velocity field in Fig. 8 suggests that even for a relatively coarse discretisation with 57,486 degrees of freedom, the velocity field is well resolved. The Jacobian matrix arising from the monolithic discretisation of 3D problems is significantly more dense

than in the 2D problems considered earlier. Hence, despite the relatively modest number of degrees of freedom there was not sufficient memory available for SUPERLU to solve the linear systems during the Newton iteration. The FSI preconditioner performs as well as in the 2D examples considered earlier, however, and GMRES, preconditioned by $\mathcal{P}_1$, converged with an average of 27.4 iterations, requiring an average of 90.2 s to solve each linear system.

5 Summary

Monolithic solvers for large-displacement FSI problems are known to be more robust than their segregated counterparts, but they are often believed to be too expensive both in terms of memory and in terms of CPU time requirements. We have compared both solution strategies within the same overall framework, implemented in OOMPH- LIB, which allows a direct comparison of their performance and have found these beliefs to be unjustified. We demonstrated that monolithic solvers are competitive even in test cases with very weak FSI, for which segregated solvers converge reliably and in a small number of iterations. For strongly coupled problems segregated solvers tend to suffer from severe convergence problems and the use of monolithic solvers becomes essential. However, for large problems, particularly in 3D, the effective use of monolithic solvers requires the provision of efficient preconditioners for the iterative solution of the large linear systems that arise during the solution of the fully coupled system of nonlinear algebraic equations by Newton's method. Block-triangular preconditioners, obtained by neglecting selected interaction blocks from the full Jacobian matrix, performed extremely well in a number of challenging test problems involving the interaction of finite-Reynolds-number flows with beam and shell structures, as well as finite-thickness solids. The implementation of the preconditioner within OOMPH- LIB allows the re-use of existing optimal solvers/preconditioning strategies for the constituent sub-problems. The key component of our implementation is the use of Elman, Silvester & Wathen's LSC Navier–Stokes preconditioner [9] which produced mesh-independent convergence rates for unsteady problems, and convergence rates that improve under uniform mesh refinement for steady problems. The preconditioner was shown to work equally well when used with adaptive mesh refinement—a key requirement for the efficient solution of problems in which the flow field contains thin shear or boundary layers.

The test cases presented in this paper are scheduled for inclusion into the next release of `oomph-lib`, expected in early 2008.

Acknowledgments The work presented in this paper was supported by the UK's Engineering and Physical Sciences Research Council (EPSRC). Stefan Kollmannsberger (TU Munich) and Floraine Cordier (ENSTA, Paris) made significant contributions to the development of the test cases discussed in Sects. 4.2.2 and 4.2.3. MH wishes to acknowledge many stimulating discussions with Ulrich Küttler and Axel Gerstenberger during his HPC-EUROPA-funded visit to Wolfgang Wall's group at the TU Munich (RII3-CT-2003-506079).

References

1. Förster C, Wall W, Ramm E (2007) Artificial mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. Comput Methods Appl Mech Eng 196:1278–1293
2. Fernandez MA, Gerbeau J-F, Grandmont C (2007) A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. Int J Numer Methods Eng 69:794–821
3. Heil M, Hazel AL (2006) OOMPH- LIB—An object-oriented multi-physics finite-element library. In: Schafer M, Bungartz H-J (eds) Fluid–structure interaction. Springer (Lecture Notes on Computational Science and Engineering), pp 19–49
4. Irons BM, Tuck RC (1969) A version of the Aitken accelerator for computer iteration. Int J Num Methods Eng 1:275–277
5. Jensen OE, Heil M (2003) High-frequency self-excited oscillations in a collapsible-channel flow. J Fluid Mech 481:235–268
6. Demmel JW, Eisenstat SC, Gilbert JR, Li XS, Liu JWH (1999) A supernodal approach to sparse partial pivoting. SIAM J. Matrix analysis and applications 20:720-755. http://crd.lbl.gov/~xiaoye/SuperLU/
7. Heil M (2004) An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems. Comput Methods Appl Mech Eng 193:1–23
8. Bertram CD, Tscherry J (2006) The onset of flow-rate limitation and flow-induced oscillations in collapsible tubes. J Fluids Struct 22:1029–1045
9. Elman HC, Silvester DJ, Wathen AJ (2006) Finite elements and fast iterative solvers with applications in incompressible fluid dynamics. Oxford University Press, New York
10. Turek S, Hron J, (2007) Proposal for numerical benchmarking of fluid–structure interaction between an elastic object and laminar incompressible flow. In: Schafer M, Bungartz H-J (eds) Fluid–structure interaction. Lecture Notes on Computational Science and Engineering. Springer, Heidelberg. pp 371–385
11. HYPRE—High performance preconditioning library. Center for Applied Scientific Computing at Lawrence Livermore National Laboratory. http://www.llnl.gov/CASC/hypre/software.html
12. TRILINOS. Sandia National Laboratories. http://trilinos.sandia.gov/