Journal of Computational Physics 231 (2012) 7315–7334

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

An efficient preconditioner for monolithically-coupled large-displacement fluid–structure interaction problems with pseudo-solid mesh updates

Richard L. Muddle [a,b], Milan Mihajlović [b], Matthias Heil [a,*]

[a] School of Mathematics, University of Manchester, Manchester M13 9PL, UK
[b] School of Computer Science, University of Manchester, Manchester M13 9PL, UK

ARTICLE INFO

Article history:
Received 21 December 2010
Received in revised form 19 June 2012
Accepted 1 July 2012
Available online 11 July 2012

Keywords:
Fluid–structure interaction
Pseudo-solid mesh updates
Monolithic discretisation
Preconditioning
Multi-physics
Algebraic multigrid
Krylov methods

ABSTRACT

We present a block preconditioner for the efficient solution of the linear systems that arise when employing Newton's method to solve monolithically-coupled large-displacement fluid–structure interaction problems in which the update of the moving fluid mesh is performed by the equations of large-displacement elasticity. Following a theoretical analysis of the preconditioner, we propose an efficient implementation that yields a solver with near-optimal computational cost, in the sense that the time for the solution of the linear systems scales approximately linearly with the number of unknowns. We evaluate the performance of the preconditioner in selected two- and three-dimensional test problems.

1. Introduction

Large-displacement fluid–structure-interaction (FSI) problems are multi-physics problems in which elastic solids interact with finite-Reynolds-number flows. Applications exist in many areas such as physiological fluid mechanics [1,2], the design of parachutes [3], tent structures [4], and artificial heart valves [5]. The accurate, robust and efficient numerical simulation of such problems poses considerable challenges, not just from a mathematical but also from a software engineering perspective. A key objective in the design of solvers for all multi-physics problems is the ability to re-use existing, often highly-optimised solvers for the constituent single-physics (here fluid and solid mechanics) problems. In so-called partitioned (or segregated) approaches, this re-use is achieved at the level of the nonlinear solvers, typically by coupling the existing single-physics solvers in a fixed-point iteration: starting from an initial guess for the shape of the fluid-loaded solid, compute the fluid flow (keeping the geometry of the domain fixed); determine the traction that the fluid exerts onto the elastic solid; compute the resulting deformation of the solid (keeping the fluid traction constant); and iterate until convergence. Such solvers are relatively easy to implement but they often suffer from a serious lack of robustness and tend to require heavy under-relaxation (resulting in slow convergence rates) to avoid the divergence of the fixed-point iteration, even in cases when good initial guesses for the solution are available. A variety of methods have been developed to accelerate the convergence of the fixed-point iteration (e.g. vector-based [6,7] or component-wise [8] Aitken extrapolation, and steepest decent methods; see e.g. [9] for a recent comparison of some of these methods). In many cases such partitioned approaches work satisfactorily. However, there are also many applications (particularly problems in which the fluid interacts with a

* Corresponding author.
E-mail address: M.Heil@maths.man.ac.uk (M. Heil).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.07.001

thin-walled solid of comparable or smaller density – a situation that is encountered frequently in physiological FSI problems) where the lack of robustness due to the so-called "added mass effect" [10] presents a significant problem. The convergence problems in the fixed-point iteration may be bypassed by employing so-called loosely coupled schemes [11]. However, these are only applicable to unsteady problems and obtain a solution by solving the constituent single-physics problems sequentially; see [12–14] for a discussion of the accuracy and stability of such methods.

The main alternative to partitioned approaches are so-called monolithic solvers which operate directly on the large system of nonlinear algebraic equations that arises from the fully-coupled implicit discretisation of the fluid and solid equations. Starting from an initial guess for the vector of the discrete unknowns, $\mathbf{x}^{[0]}$, and the corresponding vector of discrete residuals $\mathbf{r}^{[0]} = \mathbf{r}(\mathbf{x}^{[0]})$, Newton's method can be used to solve the problem via the repeated solution of (large) linear systems of the form

$$J^{[m]}\,\Delta\mathbf{x}^{[m]} = -\mathbf{r}^{[m]}, \tag{1}$$

followed by the update $\mathbf{x}^{[m+1]} = \mathbf{x}^{[m]} + \Delta\mathbf{x}^{[m]}$. Here $J^{[m]}$ is the Jacobian matrix formed from the derivatives of the discrete residuals, $\mathbf{r}$, with respect to the discrete unknowns, $\mathbf{x}$. Provided a good initial guess for the solution can be supplied (by continuation methods or timestepping), Newton's method is extremely robust and converges quadratically.

While attractive from a theoretical perspective, the practical implementation of such solvers poses its own challenges. From a software engineering point-of-view, it is difficult to implement monolithic solvers starting from separate fluid and solid mechanics codes, particularly if these are "black-box" closed-source codes for which the residuals and their derivatives are difficult to obtain. Interface Newton–Krylov methods (e.g. [15]) bypass this problem by employing a Newton method that operates only on the variables that define the position of the FSI interface, but even these methods require (or benefit from) additional knowledge about the underlying fluid and solid equations and their specific discretisation. Monolithic solvers are easiest to implement within a software framework that facilitates the formulation of multi-physics interactions via small, hierarchical modifications to any already-existing single-physics capabilities. This requirement lends itself to an object-oriented design in which multi-physics interactions are easily implemented using techniques such as function overloading, templating and multiple inheritance. We refer to [16] for a discussion of how this approach is implemented in `oomph-lib`, the object-oriented multi-physics finite element library (available as open-source software at `http://www.oomph-lib.org`), which was used for the computations presented in this paper.

The key mathematical challenge in the solution of fluid–structure interaction problems by monolithic approaches is how to efficiently solve the large system of nonlinear algebraic equations arising from the fully-coupled implicit discretisation of the fluid and solid equations, a task has frequently been described as being prohibitively expensive (see, e.g., [17,18]). However, more recent studies [19,20] have shown that monolithic approaches are not only competitive with partitioned schemes, but often outperform them, even in problems in which partitioned solvers do not suffer from any convergence problems. Possible approaches to the solution of the nonlinear equations are the use of nonlinear block Gauss–Seidel or block-Newton methods [21,22], methods that operate on the FSI interface problem [23,24], or partitioned Newton methods [25]. In this paper we employ Newton's method in its exact form and focus on the development of an efficient preconditioner for the solution of the linear system (1) by Krylov subspace solvers such as GMRES. We will exploit the block structure of the Jacobian matrix to derive a preconditioner, $\mathcal{P}$, chosen such that GMRES, applied to the preconditioned linear system

$$\mathcal{P}^{-1}J\,\Delta\mathbf{x} = -\mathcal{P}^{-1}\mathbf{r} \tag{2}$$

(where we have dropped the superscript m) converges in a number of iterations that is much smaller than the iteration count for the original linear system (1). The development of the preconditioner is guided by two requirements:

1. $\mathcal{P}$ should be chosen such that the preconditioned matrix $\mathcal{P}^{-1}J$ has a spectrum that contains only a few small clusters of eigenvalues where each cluster is tightly bounded away from the origin of the complex plane. If these bounds on the spectrum are independent of the discretisation, GMRES will converge in a near constant number of iterations regardless of the number of unknowns in the problem [26, p. 54].
2. The setup and application of the preconditioner (i.e. the solution of linear systems of the form $\mathcal{P}\mathbf{u} = \mathbf{v}$) should have optimal computational cost, i.e. be linear in the number of unknowns.

A preconditioning strategy that satisfies these two requirements will result in a solver that itself has optimal computational cost.

We restrict ourselves to ALE-based discretisations of the FSI problem in which the fluid equations are solved on a body-fitted, moving mesh [27]. In earlier work [28,19] we developed an efficient block triangular preconditioner for the linear systems arising from the monolithic discretisation of such problems for the special case in which the update of the fluid mesh in response to the wall deformation is performed by a (user-specified) algebraic node update function. This approach allows fast, sparse node updates but requires significant user input and is restricted to relatively simple geometries and to problems in which the deformation of the fluid-loaded solid can be anticipated *a priori*. Furthermore, the block-triangular preconditioners were obtained by omitting certain coupling blocks in the Jacobian matrix in an *ad hoc* manner. This made it difficult to analyse the preconditioners theoretically, and forced us to evaluate their performance exclusively by means of numerical experiments. In the current paper we present a new preconditioning methodology for the alternative approach in which the

fluid mesh update is performed automatically by determining the deformation of the fluid mesh from the equations of large-displacement elasticity [29]. Where necessary we will distinguish the two solid mechanics problems by referring to the solid body that represents the fluid domain as the "pseudo solid" as opposed to the actual fluid-loaded solid body whose physical deformation we wish to compute. A key feature of our approach that is distinct from the methodology presented in recent work by Gee et al. [30] is the way in which we impose the boundary displacements of the pseudo-solid: We mimic the approach adopted in an algebraic node update method, and thus drive the deformation of the pseudo-solid via a one-way coupling, using the deformation of the actual solid to provide the displacement boundary conditions for the pseudo-solid. With this approach, the discretised fluid equations do not depend directly on the deformation of the actual solid, and the pseudo-solid does not depend directly on the fluid flow. The absence of these direct interactions creates a particular block structure in the Jacobian matrix J which makes it possible to develop a modular preconditioner whose application allows the re-use of solvers/preconditioners for the three constituent single-physics problems (involving the fluid, the solid and the pseudo-solid) without having to neglect any off-diagonal blocks in an *ad hoc* manner (as done in [28,19,30]). This allows us to analyse the spectrum of the preconditioned matrix $\mathcal{P}^{-1}J$ theoretically and enables us to prove that the spectrum remains bounded in a tight cluster in the complex plane under mesh refinement.

We start the development of the FSI preconditioner by considering the solution of the isolated pseudo-solid problem and develop an efficient augmented-Lagrangian type preconditioner for the solution of the saddle-point linear systems that arise from the discretisation of this problem when a prescribed boundary displacement is imposed by Lagrange multipliers. This preconditioner is then incorporated into a block preconditioner for the full FSI problem. We perform a theoretical analysis of the FSI preconditioner and evaluate its performance in representative 2D and 3D test problems. Finally, we present an efficient implementation that yields a solver with approximately mesh-independent GMRES convergence rates and near-optimal computational cost.

2. Problem description

2.1. The governing equations

We assume that the fluid flow is governed by the incompressible Navier–Stokes equations

$$Re\left(St\frac{\partial u_i}{\partial t}+u_j\frac{\partial u_i}{\partial x_j}\right)=-\frac{\partial p}{\partial x_i}+\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\quad\text{and}\quad\frac{\partial u_j}{\partial x_j}=0, \tag{3}$$

where we have non-dimensionalised time, t, the spatial coordinates, x_i, and the velocities, u_i, on problem-specific reference quantities, $\mathcal{T}$, $\mathcal{L}$ and $\mathcal{U}$, respectively, and have scaled the pressure, p, on the associated viscous scale, $\mu\mathcal{U}/\mathcal{L}$. The summation convention is used throughout this paper and Greek and italic Roman indices range from 1 to 2 and 1 to 3, respectively. The two non-dimensional parameters in (3) are the Reynolds number, $Re=\rho_f\mathcal{U}\mathcal{L}/\mu$, and the Strouhal number, $St=\mathcal{L}/(\mathcal{U}\mathcal{T})$, where ρ_f and μ are the fluid density and viscosity, respectively.

We employ the same non-dimensionalisation for all lengths and time when describing the deformation of the fluid-loaded solid body. Denoting the non-dimensional position vector to a material point with Lagrangian coordinates ξ^i by $\mathbf{R}_s(\xi^i,t)$, the deformation of the solid is governed by the principle of virtual displacements (see, e.g., [31]),

$$\delta\Pi_{\text{strain}}=\int\sigma^{ij}\,\delta\gamma_{ij}\,dv=\int\left(\mathbf{b}-\Lambda^2\frac{\partial^2\mathbf{R}_s}{\partial t^2}\right)\cdot\delta\mathbf{R}_s\,dv+\oint\mathbf{f}\cdot\delta\mathbf{R}_s\,dA, \tag{4}$$

where $\delta\Pi_{\text{strain}}$ is the variation of the strain energy, expressed in terms of the components of the (geometrically nonlinear) Green's strain tensor, γ_{ij}, and the second Piola–Kirchhoff stress, σ^{ij}, non-dimensionalised on some problem-specific reference stress, $\mathcal{S}$, such as Young's modulus. The solid body is loaded by a body force $\mathbf{b}$, and the surface traction $\mathbf{f}$, which are non-dimensionalised on $\mathcal{S}/\mathcal{L}$ and $\mathcal{S}$, respectively. $\Lambda=(\mathcal{L}/\mathcal{T})\sqrt{\rho_s/\mathcal{S}}$ represents the ratio of the body's intrinsic time-scale to the time $\mathcal{T}$ used in the non-dimensionalisation of the equations; Λ^2 may be interpreted as a non-dimensional version of the solid density, ρ_s.

We will also consider cases where the fluid-loaded solid can be described by Kirchhoff–Love thin-shell theory. In such cases, $\mathbf{R}_s$ denotes the position vector to the shell's midplane which we parametrise by two Lagrangian coordinates, ξ^α. Denoting the thickness of the shell by $\mathfrak{h}$, where $\mathfrak{h}\ll\mathcal{L}$, the principle of virtual displacements becomes [31],

$$\delta\Pi_{\text{strain}}=\iint E^{\alpha\beta\gamma\delta}\left[\gamma_{\alpha\beta}\,\delta\gamma_{\gamma\delta}+\frac{1}{12}\left(\frac{\mathfrak{h}}{\mathcal{L}}\right)^2\kappa_{\alpha\beta}\,\delta\kappa_{\gamma\delta}\right]\sqrt{a}\,d\xi^1\,d\xi^2=\iint\left[\left(\frac{\mathcal{L}}{\mathfrak{h}}\right)\sqrt{\frac{A}{a}}\mathbf{f}-\Lambda^2\frac{\partial^2\mathbf{R}_s}{\partial t^2}\right]\cdot\delta\mathbf{R}_s\sqrt{A}\,d\xi^1\,d\xi^2, \tag{5}$$

where $\gamma_{\alpha\beta}$ and $\kappa_{\alpha\beta}$ denote the non-dimensional mid-plane strain and bending tensors, and A and a are the determinants of the mid-plane metric tensors in the deformed and undeformed configuration, respectively. In (5) we have assumed linearly elastic behaviour, described by the tensor of elastic coefficients, $E^{\alpha\beta\gamma\delta}$, which we non-dimensionalised on the shell's effective Young's modulus, so that $\mathcal{S}=E/(1-\nu^2)$, where E and ν are the solid's 3D Young's modulus and Poisson ratio, respectively.

The fluid and solid equations interact at the fluid-loaded solid boundary, ∂D_{FSI}, along which the fluid exerts a traction onto the solid so that the components of the applied traction (non-dimensionalised on the solid stress scale, $\mathcal{S}$) are given by

$$f_i = Q\left(pN_i - \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)N_j\right) \quad \text{on } \partial D_{\text{FSI}}, \tag{6}$$

where the N_j are the components of the outer unit normal on the fluid. Here

$$Q = \frac{\mu \mathcal{U}}{\mathcal{L}\mathcal{S}} \tag{7}$$

is the final non-dimensional parameter in the problem. It represents the ratio of the fluid and solid stress scales, and characterises the strength of fluid–structure interaction in the sense that as $Q \to 0$ the deformation of the solid ceases to be affected by the fluid load, either because the stiffness of the solid becomes very large or because the fluid loading becomes very weak.

The solid affects the fluid via the no-slip condition

$$\mathbf{u} = St\frac{\partial \mathbf{R}_s}{\partial t} \quad \text{on } \partial D_{\text{FSI}}, \tag{8}$$

and by changing the shape of the domain occupied by the fluid.

In general, the problem is subject to further problem-specific single-physics boundary conditions, e.g., the in- and outflow conditions for the fluid, or displacement constraints and additional external loads on the fluid-loaded solid.

2.2. Discretisation and fluid mesh update

The finite-element discretisation of the principle of virtual displacements (4) [or (5)] is straightforward because the equations are formulated in Lagrangian coordinates and the integrals are therefore performed over a fixed domain. Conversely, the fluid equations are expressed in Eulerian form and have to be solved in the time-varying domain occupied by the fluid. We therefore discretise the fluid equations on a body-fitted moving mesh, taking the mesh motion into account when evaluating the (Eulerian) time-derivative in (3); see [32]. This approach is sometimes referred to as the three-field formulation of the fluid–structure interaction problem [33].

To update the fluid mesh in response to the deformation of the FSI boundary, we treat the fluid domain as a (pseudo-)solid body and regard the fluid nodes as material points within this deforming domain [29]. This is illustrated in the sketch in Fig. 1 which shows a 2D fluid domain, D, which is parametrised by Lagrangian coordinates (ξ^1, ξ^2), so that the position vector to a material point in the (pseudo-)solid is given by $\mathbf{R}_{\text{PS}}(\xi^1, \xi^2)$. We parametrise the FSI boundary, ∂D_{FSI}, by a boundary coordinate ζ such that $(\xi^1, \xi^2)|_{\partial D_{\text{FSI}}} = (\xi^1_{\partial D_{\text{FSI}}}(\zeta), \xi^2_{\partial D_{\text{FSI}}}(\zeta))$ and impose a *prescribed* displacement $\mathbf{R}_{\text{FSI}}(\zeta)$ which, in the fully-coupled FSI problem, will be determined by the deformation of the fluid-loaded solid. The constraint

$$\mathbf{R}_{\text{PS}}\left(\xi^1_{\partial D_{\text{FSI}}}(\zeta), \xi^2_{\partial D_{\text{FSI}}}(\zeta)\right) - \mathbf{R}_{\text{FSI}}(\zeta) = 0 \quad \text{on } \partial D_{\text{FSI}} \tag{9}$$

could, in principle, be imposed as a Dirichlet condition for the displacement field of the pseudo-solid. However, for reasons that will become apparent later (see Section 4.1 and the discussion in Section 5), we impose (9) by augmenting the principle of virtual displacements (4) by a Lagrange multiplier term,

$$\delta\Pi_{\text{strain}} + \delta\Pi_{\text{constraint}} = 0, \tag{10}$$

where

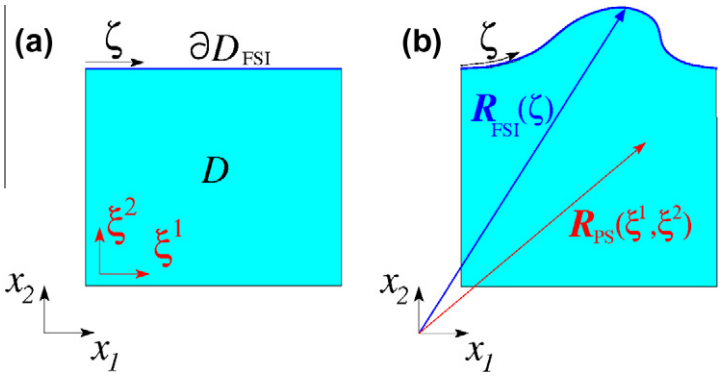

Fig. 1. Two-dimensional (pseudo-)solid problem in its (a) undeformed and (b) deformed configuration.

$$\Pi_{\text{constraint}} = \int_{\partial D_{\text{FSI}}} \left(\mathbf{R}_{\text{PS}}\left(\xi^1_{\partial D_{\text{FSI}}}(\zeta), \xi^2_{\partial D_{\text{FSI}}}(\zeta)\right) - \mathbf{R}_{\text{FSI}}(\zeta) \right) \cdot \boldsymbol{\Lambda} \left| \frac{d\mathbf{R}_{\text{PS}}\left(\xi^1_{\partial D_{\text{FSI}}}(\zeta), \xi^2_{\partial D_{\text{FSI}}}(\zeta)\right)}{d\zeta} \right| d\zeta. \tag{11}$$

The above is easily generalised to the 3D case. Body forces, surface tractions and inertial effects are ignored in (10) since they are not required to enforce the displacement constraint. The Lagrange multiplier $\boldsymbol{\Lambda}$ in (11) can be interpreted as the surface traction to be applied to ∂D_{FSI} in order to produce the required boundary deformation. We note that the use of the full, geometrically non-linear expression for Green's strain tensor, γ_{ij}, in the computation of the strain energy in (4) prevents the inversion of elements. This is essential if the fluid domain undergoes large deformations. However, since the pseudo-solid problem is only used to update the fluid mesh and does not have any physical significance it is possible and, as we shall see below, often beneficial to reset the pseudo-solid's stress-free reference configuration occasionally, e.g. after every load increment or timestep.

Having ensured the continuity of the displacements at the FSI boundary, we are free to express the right-hand-side of the no-slip condition (8) (which we interpret as a Dirichlet condition for the fluid velocity) in terms of the solid or the pseudo-solid. Choosing the latter will turn out to be beneficial for the development of the preconditioner (presented in Section 4.1 below) because it will allow us to theoretically prove the optimality of our preconditioning approach (in Section 4.2). This is a key advantage compared to the methodologies presented in references [28,19,30] where the performance of the preconditioners (obtained by the *ad hoc* neglect of certain off-diagonal blocks in the Jacobian matrix) could only be evaluated by numerical experiments, and, in some cases, showed a significant increase in GMRES iteration counts under mesh refinement, indicating the non-optimality of the respective preconditioner.

The problem is now fully specified by Eqs. (3), (4) [or (5)] and (10) which are subject to the coupling conditions (6), (8) and (9), and the additional problem-specific single-physics boundary conditions.

3. An efficient solver for the (pseudo-)solid equations subject to a prescribed boundary displacement

We start the development of a preconditioner for the linear systems arising in the solution of the fully-coupled FSI problem by considering the solution of the (pseudo-)solid Eqs. (10) for a given boundary displacement, $\mathbf{R}_{\text{FSI}}(\zeta)$. We discretise the vector field $\mathbf{R}_{\text{PS}}$ in terms of the nodal coordinates, so that

$$R_{\text{PS}\,i} = \sum_{j=1}^{N_{\text{node}}} X_{ij}\,\psi_j, \tag{12}$$

where N_{node} represents the number of nodes in the mesh, X_{ij} is the i-th coordinate of the j-th node, and ψ_j is the finite element basis function associated with node j. We use the same basis functions to expand the Lagrangian coordinates as

$$\xi^i = \sum_{j=1}^{N_{\text{node}}} \Xi_{ij}\,\psi_j, \tag{13}$$

where we set the nodal values of the Lagrangian coordinates to the node's initial Eulerian position, i.e. $\Xi_{ij} = X^{[0]}_{ij}$, ensuring that the undeformed configuration is stress free. The Lagrange multiplier field is only defined on the constrained boundary, ∂D_{FSI}, where we expand it as

$$\Lambda_i = \sum_{j=1}^{\overline{N_{\text{node}}}} L_{ij}\,\overline{\psi_j}. \tag{14}$$

Here the sum is taken over the $\overline{N_{\text{node}}}$ nodes on the constrained boundary and the basis functions $\overline{\psi_j}$ are obtained by restricting the interior basis functions ψ_j to ∂D_{FSI}.

We order the discrete unknowns so that the vector of unknowns, $\mathbf{x}_{\text{PS}} = [\mathbf{X}, \mathbf{L}]$, contains the n_{x} unknown nodal coordinates, followed by the n_{b} unknown discrete Lagrange multipliers. The linear systems to be solved in the course of the Newton-based solution of Eq. (10), subject to the displacement constraint (9), then have saddle-point structure,

$$J_{\text{PS}}\,\Delta\mathbf{x}_{\text{PS}} = \left[\begin{array}{c|c} E & C_{\text{xl}} \\ \hline C_{\text{lx}} & \end{array}\right] \left[\begin{array}{c} \Delta\mathbf{X} \\ \Delta\mathbf{L} \end{array}\right] = -\left[\begin{array}{c} \mathbf{r}_{\text{X}} \\ \mathbf{r}_{\text{L}} \end{array}\right], \tag{15}$$

where E is the tangent stiffness matrix of the unconstrained pseudo-solid problem, and the two off-diagonal blocks C_{xl} and $C_{\text{lx}} = C^T_{\text{xl}}$ arise through the imposition of the displacement constraint by the Lagrange multipliers. We refer to [34] for the proof of the LBB stability of this discretisation; see also [35,36] for a discussion of the LBB stability of the Lagrange-multiplier-based imposition of Dirichlet boundary conditions in related problems. We note that during the first step of the Newton iteration, E is symmetric positive definite since it represents the tangent stiffness matrix relative to the system's equilibrium configuration.

Subdividing the unknowns further according to their components in the Cartesian coordinate directions, and into unknown nodal positions that lie on the constrained boundary and those that do not, allows Eq. (15) to be rewritten as

$$\begin{array}{c} n_x \updownarrow \\ n_{\bar{x}} \updownarrow \\ n_y \updownarrow \\ n_{\bar{y}} \updownarrow \\ n_{\bar{x}} \updownarrow \\ n_{\bar{y}} \updownarrow \end{array} \left[\begin{array}{cccc|cc} E_{xx} & E_{x\bar{x}} & E_{xy} & E_{x\bar{y}} & & \\ E_{\bar{x}x} & E_{\bar{x}\bar{x}} & E_{\bar{x}y} & E_{\bar{x}\bar{y}} & M_x & \\ E_{yx} & E_{y\bar{x}} & E_{yy} & E_{y\bar{y}} & & \\ E_{\bar{y}x} & E_{\bar{y}\bar{x}} & E_{\bar{y}y} & E_{\bar{y}\bar{y}} & & M_y \\ \hline & M_x & & & & \\ & & & M_y & & \end{array}\right] \left[\begin{array}{c} \Delta\mathbf{X}_x \\ \Delta\overline{\mathbf{X}}_x \\ \Delta\mathbf{X}_y \\ \Delta\overline{\mathbf{X}}_y \\ \Delta\mathbf{L}_x \\ \Delta\mathbf{L}_y \end{array}\right] = - \left[\begin{array}{c} \mathbf{r}_x \\ \mathbf{r}_{\bar{x}} \\ \mathbf{r}_y \\ \mathbf{r}_{\bar{y}} \\ \mathbf{r}_{l_x} \\ \mathbf{r}_{l_y} \end{array}\right], \tag{16}$$

where we have restricted ourselves to the two-dimensional case. The vectors $\mathbf{X}_x$, $\mathbf{X}_y$, $\mathbf{L}_x$ and $\mathbf{L}_y$ contain the x_1- and x_2-components of the unknown nodal positions and Lagrange multipliers, respectively. (Note that we use subscripts x and y rather than x_1 and x_2 when labelling matrix and vector blocks to avoid double-subscripts.) The overbars denote the unknown nodal positions that are constrained by the Lagrange multiplier. The matrices M_x and M_y are mass matrices whose entries are formed from products of the basis functions, $[M_x]_{ij} = [M_y]_{ij} = \int_{\partial D_{FSI}} \overline{\psi_i}\,\overline{\psi_j}\, dS$.

Saddle point systems of the form (15) can be preconditioned by so-called augmentation preconditioners [37–39] of the form

$$\mathcal{P}_{aug} = \begin{bmatrix} E_{aug} & \\ & W \end{bmatrix} = \begin{bmatrix} E + C_{xl} W^{-1} C_{lx} & \\ & W \end{bmatrix}, \tag{17}$$

where $W \in \Re^{n_b \times n_b}$ (note that $n_b = n_{\bar{x}} + n_{\bar{y}}$) is an arbitrary matrix, chosen such that $\mathcal{P}_{aug}$ is an effective preconditioner, while ensuring that the augmented matrix $E_{aug} = E + C_{xl} W^{-1} C_{lx}$ remains sparse, and that linear systems involving E_{aug} and W are easy to solve (either exactly or approximately).

Given the block structure of (16), we develop a new optimal preconditioner by choosing

$$W = \begin{bmatrix} \frac{1}{\sigma} M_x^2 & \\ & \frac{1}{\sigma} M_y^2 \end{bmatrix}, \tag{18}$$

with $\sigma = \|E\|_\infty$. (The rationale behind this choice for σ will become clear in Section 3.1.) This produces a diagonal augmentation matrix and yields the preconditioner

$$\mathcal{P}_{PS} = \left[\begin{array}{cccc|cc} E_{xx} & E_{x\bar{x}} & E_{xy} & E_{x\bar{y}} & & \\ E_{\bar{x}x} & E_{\bar{x}\bar{x}} + \sigma I & E_{\bar{x}y} & E_{\bar{x}\bar{y}} & & \\ E_{yx} & E_{y\bar{x}} & E_{yy} & E_{y\bar{y}} & & \\ E_{\bar{y}x} & E_{\bar{y}\bar{x}} & E_{\bar{y}y} & E_{\bar{y}\bar{y}} + \sigma I & & \\ \hline & & & & \frac{1}{\sigma} M_x^2 & \\ & & & & & \frac{1}{\sigma} M_y^2 \end{array}\right]. \tag{19}$$

Our choice for W therefore not only retains the sparsity pattern of E, but also increases the diagonal dominance of the $E_{\bar{x}\bar{x}}$ and $E_{\bar{y}\bar{y}}$ blocks which will facilitate the approximate solution of the linear systems involving E_{aug} by multigrid methods during the application of the preconditioner. The lower-right block, W, is block diagonal by construction and the linear systems involving M_x^2 and M_y^2 can be solved by solving twice systems involving the mass matrices themselves. For example, the system $M_x^2 \mathbf{u} = \mathbf{v}$ can be solved by first solving $M_x \mathbf{w} = \mathbf{v}$ and then $M_x \mathbf{u} = \mathbf{w}$. This avoids having to perform matrix–matrix products to form M_x^2 and M_y^2, and allows the use existing optimal solvers for mass matrices.

3.1. Spectral properties of the preconditioned linear system

We assess the quality of the preconditioner by analysing the spectrum of the preconditioned matrix $\mathcal{P}_{PS}^{-1} J_{PS}$. For this purpose we perform a final re-ordering of the unknowns into three groups: the unknown nodal positions that are and are not subject to the displacement constraint (identified by subscripts b and i, respectively), and the unknown Lagrange multipliers (identified by the subscript l). The eigenvalues λ of $\mathcal{P}_{PS}^{-1} J_{PS}$ are then determined by the generalised eigenvalue problem

$$\begin{bmatrix} E_{ii} & E_{ib} & \\ E_{bi} & E_{bb} & M \\ & M & \end{bmatrix} \begin{bmatrix} \mathbf{w}_i \\ \mathbf{w}_b \\ \mathbf{w}_l \end{bmatrix} = \lambda \begin{bmatrix} E_{ii} & E_{ib} & \\ E_{bi} & E_{bb} + \sigma I & \\ & & \frac{1}{\sigma} M^2 \end{bmatrix} \begin{bmatrix} \mathbf{w}_i \\ \mathbf{w}_b \\ \mathbf{w}_l \end{bmatrix} \begin{array}{l} \updownarrow n_i \\ \updownarrow n_b \\ \updownarrow n_b \end{array}, \tag{20}$$

where $n_i = n_x + n_y$,

$$E_{ii} = \begin{bmatrix} E_{xx} & E_{xy} \\ E_{yx} & E_{yy} \end{bmatrix}, \quad E_{ib} = \begin{bmatrix} E_{x\bar{x}} & E_{x\bar{y}} \\ E_{y\bar{x}} & E_{y\bar{y}} \end{bmatrix}, \quad E_{bi} = \begin{bmatrix} E_{\bar{x}x} & E_{\bar{x}y} \\ E_{\bar{y}x} & E_{\bar{y}y} \end{bmatrix}, \quad E_{bb} = \begin{bmatrix} E_{\bar{x}\bar{x}} & E_{\bar{x}\bar{y}} \\ E_{\bar{y}\bar{x}} & E_{\bar{y}\bar{y}} \end{bmatrix}$$

and

$$M = \begin{bmatrix} M_x & \\ & M_y \end{bmatrix}. \tag{21}$$

Multiplying Eq. (20) from the left by $\mathcal{P}_{PS}^{-1}$, using the identity

$$\mathcal{P}_{PS}^{-1} = \begin{bmatrix} I & \sigma S_{ii}^{-1} E_{ib} E_{bb}^{-1} (I + \sigma S_{bb}^{-1})^{-1} & \\ & (I + \sigma S_{bb}^{-1})^{-1} & \\ & & I \end{bmatrix} \begin{bmatrix} S_{ii}^{-1} & -S_{ii}^{-1} E_{ib} E_{bb}^{-1} & \\ -S_{bb}^{-1} E_{bi} E_{ii}^{-1} & S_{bb}^{-1} & \\ & & \sigma M^{-2} \end{bmatrix}, \tag{22}$$

where $S_{bb} = E_{bb} - E_{bi} E_{ii}^{-1} E_{ib}$ and $S_{ii} = E_{ii} - E_{ib} E_{bb}^{-1} E_{bi}$ are Schur complements of E, yields an upper-block triangular eigenvalue problem whose top left block is an $n_i \times n_i$ identity matrix,

$$\left[\begin{array}{c|cc} I & \sigma S_{ii}^{-1} E_{ib} E_{bb}^{-1} (I + \sigma S_{bb}^{-1})^{-1} & S_{ii}^{-1} E_{ib} E_{bb}^{-1} (\sigma (I + \sigma S_{bb}^{-1})^{-1} S_{bb}^{-1} - I) M \\ \hline & (I + \sigma S_{bb}^{-1})^{-1} & (I + \sigma S_{bb}^{-1})^{-1} S_{bb}^{-1} M \\ & \sigma M^{-1} & \end{array}\right] \begin{bmatrix} \mathbf{w}_i \\ \mathbf{w}_b \\ \mathbf{w}_l \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{w}_i \\ \mathbf{w}_b \\ \mathbf{w}_l \end{bmatrix}. \tag{23}$$

This implies that $\mathcal{P}_{PS}^{-1} J_{PS}$ has n_i unit eigenvalues. The remaining $2n_b$ eigenvalues are determined by the $2n_b \times 2n_b$ eigenvalue problem

$$\begin{bmatrix} (I + \sigma S_{bb}^{-1})^{-1} & (I + \sigma S_{bb}^{-1})^{-1} S_{bb}^{-1} M \\ \sigma M^{-1} & \end{bmatrix} \begin{bmatrix} \mathbf{w}_b \\ \mathbf{w}_l \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{w}_b \\ \mathbf{w}_l \end{bmatrix}. \tag{24}$$

Substituting the second block row ($\mathbf{w}_l = (\sigma/\lambda) M^{-1} \mathbf{w}_b$) of Eq. (24) into the first one yields

$$\mathbf{w}_b + \frac{\sigma}{\lambda} S_{bb}^{-1} \mathbf{w}_b = \lambda \mathbf{w}_b + \lambda \sigma S_{bb}^{-1} \mathbf{w}_b, \tag{25}$$

which reveals the existence of a number of further unit eigenvalues. The remaining (non-unit) eigenvalues satisfy

$$S_{bb} \mathbf{w}_b = -\sigma \frac{1 + \lambda}{\lambda} \mathbf{w}_b \tag{26}$$

which shows that the n_b non-unit eigenvalues of $\mathcal{P}_{PS}^{-1} J_{PS}$ are related to the eigenvalues λ_{bb} of the Schur complement S_{bb} via

$$\lambda = -\frac{\sigma}{\sigma + \lambda_{bb}}. \tag{27}$$

Assuming that E is symmetric positive definite (a condition that is satisfied exactly only for the linear system to be solved during the first Newton step) we use the fact that the eigenvalues of any Schur complement of a Hermitian semi-definite matrix interlace the eigenvalues of that matrix [40]. This implies that the eigenvalues λ_{bb} of S_{bb} satisfy $0 < \lambda_{bb} \leqslant \|E\|_\infty$. Substituting this into Eq. (27) and recalling that $\sigma = \|E\|_\infty$ then shows that the non-unit eigenvalues of $\mathcal{P}_{PS}^{-1} J_{PS}$ are bounded by[1]

$$-1 < \lambda \leqslant -\frac{1}{2}. \tag{28}$$

Hence, the majority $(n_i + n_b)$ of the eigenvalues of $\mathcal{P}_{PS}^{-1} J_{PS}$ are equal to one while a small number ($n_b \ll n_i$) of additional eigenvalues are located in a small, bounded and mesh-independent interval between -1 and $-\frac{1}{2}$, indicating that the use of the preconditioner $\mathcal{P}_{PS}$ in a Krylov subspace solver such as GMRES will lead to mesh-independent iteration counts.

If E is not symmetric positive definite (as will be the case in subsequent Newton steps), we will retain the unit eigenvalues but the remaining non-unit eigenvalues will no longer satisfy the bounds in (28) exactly. However, given that during the simulation of an FSI problem the undeformed reference configuration is typically re-initialised after every timestep (by setting Ξ_{ij} to the current nodal positions), and that the fluid domain only tends to be subjected to modest deformations during each timestep, we do not expect the spectrum of E to change significantly. This suggests that the behaviour of the preconditioner will still be well characterised by the analysis presented above. This will be confirmed in the numerical experiments presented in the next section.

Finally we note that the above analysis is easily generalised to the three-dimensional case and yields the same bounds (28) on the non-unit eigenvalues of the preconditioned matrix.

3.2. Numerical evaluation of the preconditioner and development of an optimal implementation

We will now evaluate the performance of the preconditioner in a representative test problem and describe the development of an optimal implementation which retains the mesh-independent iteration counts predicted by the theoretical anal-

[1] This provides the motivation for the scaling $\sigma = \|E\|_\infty$ in (18). If we had omitted the scaling (i.e. set $\sigma = 1$) then the bounds on the non-unit eigenvalues would change to $-1 < \lambda < (1 + \max(\lambda_{bb}))^{-1} < 0$ which is not as tight as the bounds given in (28) and is likely to introduce a mesh dependence into the spectrum of the preconditioned matrix.

ysis, and allows the application of the preconditioner at a computational cost that is proportional to the number of unknowns in the problem.

We consider the deformation of a unit square subject to a boundary displacement

$$\mathbf{R}_{\mathrm{FSI}}(\zeta) = \begin{pmatrix} (1-\alpha)\zeta + \alpha(4(\zeta^3 - \zeta^2) + \zeta) \\ 1 + 2\alpha(\zeta^2 - \zeta^3) \end{pmatrix} \tag{29}$$

prescribed on the upper wall ($x_2 = 1$) which is parametrised by the boundary coordinate $\zeta \in [0, 1]$, while the other boundaries are held in a fixed position. The parameter α controls the magnitude of the imposed deflection, with $\alpha = 0$ corresponding to the undeformed state. We discretised the domain with quadrilateral bi-quadratic (Q2) elements and evaluated the performance of the preconditioner in a sequence of finite displacement problems. Starting from $\alpha = 0$, we increment α by 0.1, solved the resulting problem by Newton's method, reset the undeformed configuration, and repeated this until $\alpha = 1$. Fig. 2 illustrates three representative stages of the deformation and shows that in the final stages of the computation some of the elements are very distorted, providing a challenging test case for our solver.

Fig. 3 shows a plot of the Lagrange multiplier field (representing the surface traction required to deform the pseudo-solid into the required shape) for different levels of mesh refinement. The smoothness of the Lagrange multiplier field and the plot of the L2 norm of the error between the prescribed and actual surface displacement as a function of the mesh size demonstrate the accuracy and stability of the discretisation.

3.2.1. Performance of the exact preconditioner

We first consider the performance of GMRES (convergence tolerance $\tau = 10^{-8}$), preconditioned with the exact version of the preconditioner $\mathcal{P}_{\mathrm{PS}}$, obtained by solving the (subsidiary) linear systems involving E_{aug}, M_{x} and M_{y} with the sparse direct solver SuperLU [41].

Table 1 shows the average number of GMRES iterations, N_{GMRES}, (averaged over all linear systems solved in the course of the Newton iteration) as a function of the problem size (characterised by the number of unknowns, N_{dof}, and the mesh size, h, defined in Fig. 2(a)) and Poisson's ratio, ν, for the initial stage of the computation ($\alpha = 0 \rightarrow 0.1$). The iteration count is low and mesh-independent with a very weak dependence on Poisson's ratio.

Table 2 displays the dependence of N_{GMRES} on the problem size for different levels of the deformation at a fixed Poisson ratio of $\nu = 0.1$. The iteration count remains mesh-independent for all cases and displays a modest increase with the degree of deformation.

The iteration counts are similar for all linear solves performed in the course of the Newton iterations, confirming our assumption that the behaviour of the preconditioner is well characterised by the analysis presented above, even in cases when E is not exactly symmetric positive definite.

3.2.2. Development of an optimal preconditioner

GMRES, preconditioned by $\mathcal{P}_{\mathrm{PS}}$ in its exact form, does not constitute an optimal solver for the linear system (16) because the computational cost of solving the subsidiary linear systems involving E_{aug}, M_{x} and M_{y} with a sparse direct solver does not scale linearly with the number of unknowns. We therefore seek optimal-cost inexact solvers for these subsidiary linear systems.

The augmentation of E only affects a small subset of diagonal coefficients which suggests that effective (approximate) solvers for E should also constitute good solvers for linear system involving E_{aug}. We therefore employ the block-preconditioning approach developed in [42] for the Navier–Lamé equation of linear elasticity, and replace E_{aug} by its upper block-triangular counterpart (grouping together the unknowns according to their coordinate directions), to obtain

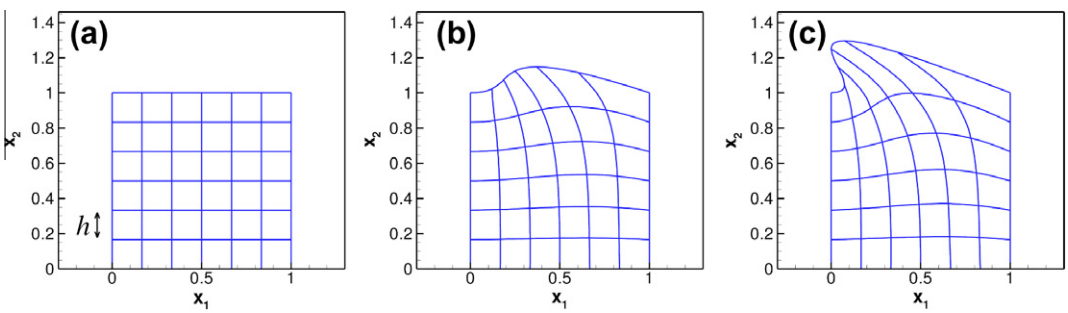

Fig. 2. Three domain shapes computed in the solution of the two-dimensional test problem for a Poisson's ratio of $\nu = 0.1$. (a) $\alpha = 0$, (b) $\alpha = 0.5$ and (c) $\alpha = 1.0$.

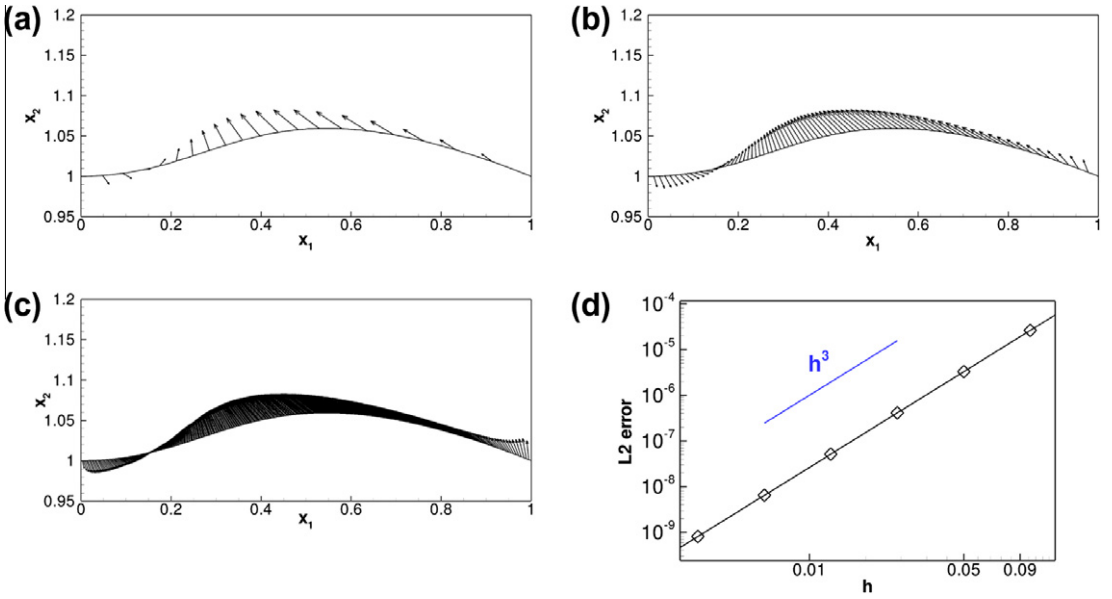

Fig. 3. (a–c) Plot of the Lagrange multiplier field (representing the surface traction required to deform the pseudo-solid into the required shape) for different levels of mesh refinement: (a) $h = 1/20$; (b) $h = 1/80$; (c) $h = 1/320$. (d) Plot of the L2 norm of the error between the prescribed and actual surface displacement as a function of the mesh size, h. All figures are for $\nu = 0.1$ and $\alpha = 0.3$.

Table 1
Average GMRES iteration count, N_{GMRES}, as a function of problem size (number of unknowns, N_{dof}, and mesh size, h) and Poisson's ratio (ν), using the exact version of the preconditioner, $\mathcal{P}_{\text{PS}}$, for the first displacement increment ($\alpha = 0 \to 0.1$).

N_{dof} / h	19,998 / $\frac{1}{25}$	79,998 / $\frac{1}{50}$	319,998 / $\frac{1}{100}$
$\nu = 0.1$	8.2	8.2	8.2
$\nu = 0.2$	8.0	8.0	8.0
$\nu = 0.3$	8.0	8.0	8.2
$\nu = 0.4$	8.2	8.2	8.2

Table 2
Average GMRES iteration count, N_{GMRES}, as a function of problem size (number of unknowns, N_{dof}, and mesh size, h) and degree of deformation (α), using the exact version of the preconditioner, $\mathcal{P}_{\text{PS}}$ ($\nu = 0.1$).

N_{dof} / h	19,998 / $\frac{1}{25}$	79,998 / $\frac{1}{50}$	319,998 / $\frac{1}{100}$
(a) $\alpha = 0.0 \to 0.1$	8.2	8.2	8.2
(b) $\alpha = 0.5 \to 0.6$	9.5	9.7	9.7
(c) $\alpha = 0.9 \to 1.0$	11.0	11.0	11.0

$$\widetilde{\mathcal{P}}_{\text{PS}} = \left[\begin{array}{cccc|cc} E_{xx} & E_{x\bar{x}} & E_{xy} & E_{x\bar{y}} & & \\ E_{\bar{x}x} & E_{\bar{x}\bar{x}} + \sigma I & E_{\bar{x}y} & E_{\bar{x}\bar{y}} & & \\ & & E_{yy} & E_{y\bar{y}} & & \\ & & E_{\bar{y}y} & E_{\bar{y}\bar{y}} + \sigma I & & \\ \hline & & & & \frac{1}{\sigma}M_x^2 & \\ & & & & & \frac{1}{\sigma}M_y^2 \end{array}\right]. \tag{30}$$

When applying this preconditioner we approximately solve the subsidiary linear systems involving the diagonal blocks

$$\begin{bmatrix} E_{xx} & E_{x\bar{x}} \\ E_{\bar{x}x} & E_{\bar{x}\bar{x}} + \sigma I \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} E_{yy} & E_{y\bar{y}} \\ E_{\bar{y}y} & E_{\bar{y}\bar{y}} + \sigma I \end{bmatrix} \tag{31}$$

Table 3
Average GMRES iteration count, N_{GMRES}, and (in brackets) the average GMRES solution times, T_{solve}, (in seconds), as a function of problem size (number of unknowns, N_{dof}, and mesh size, h) and Poisson's ratio (ν), using the inexact version of the preconditioner, $\widetilde{\mathcal{P}}_{\text{PS}}$, for the first displacement increment ($\alpha = 0 \rightarrow 0.1$).

N_{dof} / h	19,998 / $\frac{1}{25}$	79,998 / $\frac{1}{50}$	319,998 / $\frac{1}{100}$
$\nu = 0.1$	19.5 [0.91]	19.5 [4.49]	20.0 [22.26]
$\nu = 0.2$	21.2 [0.95]	21.2 [4.79]	21.2 [22.87]
$\nu = 0.3$	23.4 [1.21]	23.8 [6.55]	24.2 [29.50]
$\nu = 0.4$	30.4 [1.47]	31.6 [8.16]	33.2 [39.84]

Table 4
Average GMRES iteration count, N_{GMRES}, and (in brackets) the average GMRES solution times, T_{solve}, (in seconds), as a function of problem size (number of unknowns, N_{dof}, and mesh size, h) and degree of deformation (α), using the inexact version of the preconditioner, $\widetilde{\mathcal{P}}_{\text{PS}}$ ($\nu = 0.1$).

N_{dof} / h	19,998 / $\frac{1}{25}$	79,998 / $\frac{1}{50}$	319,998 / $\frac{1}{100}$
(a) $\alpha = 0.0 \rightarrow 0.1$	19.5 [0.91]	19.5 [4.49]	20.0 [22.26]
(b) $\alpha = 0.5 \rightarrow 0.6$	23.2 [1.05]	24.0 [5.54]	24.5 [26.98]
(c) $\alpha = 0.9 \rightarrow 1.0$	27.7 [1.25]	30.0 [6.89]	33.7 [37.36]

by performing a fixed number of algebraic multigrid (AMG) cycles. In the computations presented below the approximate solves were performed with two $V(1,1)$ multigrid cycles, using BoomerAMG, Hypre's implementation of AMG [43], with Ruge–Stüben coarsening (strength parameter 0.25) using Gauss–Seidel as the smoother. As discussed above, the subsidiary linear systems involving M_{x}^2 and M_{y}^2 can be solved by two consecutive linear solves involving the mass matrices themselves. Following Wathen [44] we employ CG, preconditioned by diagonal scaling, as the solver for these systems. This solver converges so rapidly that a fixed number of four CG iterations suffices to compute the solution to sufficient accuracy so that $\widetilde{\mathcal{P}}_{\text{PS}}$ can be treated as a constant preconditioner. With these approximations, the computational cost of the application of $\widetilde{\mathcal{P}}_{\text{PS}}$ is linear in the number of unknowns.

Tables 3 and 4 show that the use of the various approximations increases the number of GMRES iterations, but only introduces a weak mesh dependence into the iteration counts, indicating the solver is now close to optimal. (More detailed studies show that the slight increase in iteration counts for large Poisson ratios and for very strongly deformed configurations is caused by the use of AMG for the approximate solution of the linear systems involving the matrices in (31).) The near-optimality of the solver is confirmed by the average solution times, T_{solve}, shown in brackets in Tables 3 and 4, and in Fig. 4 which indicate that for the optimal implementation of the preconditioner, T_{solve} increases approximately linearly with the number of unknowns. The best behaviour (both in terms of absolute solve times and mesh (in)dependence) is obtained for small values of Poisson's ratio ν – an important observation for the application of the preconditioner in an FSI context where the constitutive parameters for the pseudo-solid can be chosen arbitrarily. In the FSI computations presented below we set Poisson's ratio for the pseudo-solid to $\nu = 0.1$. Furthermore, the deterioration of the iteration counts for large deformations only occurs when the mesh is already so strongly distorted (as in Fig. 2(c)) that the solution of the Navier–Stokes equations on that mesh is unlikely to be of particularly high quality. This problem arises in all ALE-based discretisations of moving boundary problems – remeshing is required if the domain is deformed too strongly. Finally, we note that the behaviour reported for the 2D test case analysed here is representative of the behaviour in equivalent 3D test cases [45].

4. Preconditioning for FSI problems with pseudo-solid mesh-updates

We will now utilise our new pseudo-solid preconditioner in the development of a preconditioner for the fully-coupled FSI problem. The linear systems to be solved in the course of the Newton iteration for such problems have the following block structure

$$J_{\text{FSI}}\,\Delta\mathbf{x} = \left[\begin{array}{ccc|c} F & & & C_{\text{fx}} \\ C_{\text{sf}} & S & & C_{\text{sx}} \\ \hline & & E & C_{\text{xl}} \\ & C_{\text{ls}} & C_{\text{lx}} & \end{array}\right] \left[\begin{array}{c} \Delta\mathbf{F} \\ \Delta\mathbf{S} \\ \Delta\mathbf{X} \\ \Delta\mathbf{L} \end{array}\right] = -\left[\begin{array}{c} \mathbf{r}_{\text{f}} \\ \mathbf{r}_{\text{s}} \\ \mathbf{r}_{\text{x}} \\ \mathbf{r}_{\text{l}} \end{array}\right] \begin{array}{l} \updownarrow n_{\text{f}} \\ \updownarrow n_{\text{s}} \\ \updownarrow n_{\text{x}} \\ \updownarrow n_{\text{b}} \end{array}, \tag{32}$$

where the vector $\mathbf{F}$ contains the Navier–Stokes (fluid) unknowns (velocities and pressure), $\mathbf{S}$ represents unknowns describing the deformation of the fluid-loaded solid, $\mathbf{X}$ represents the nodal positions in the fluid mesh, and $\mathbf{L}$ contains the discrete Lagrange multipliers which impose the deformation of the FSI boundary in the fluid mesh. The diagonal blocks in the Jacobian matrix, J_{FSI}, are the two "single-physics" Jacobian matrices (F, the Navier–Stokes Jacobian; S, the tangent stiffness matrix of the fluid-loaded solid) and the pseudo-solid Jacobian, J_{PS}, defined in (15). The non-zero off-diagonal blocks arise through the interactions between the various problems: C_{fx} represents the effect of the pseudo-solid unknowns (the nodal positions in

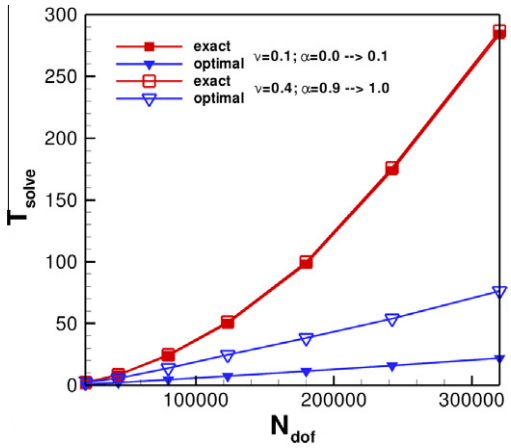

Fig. 4. Average GMRES solution times, T_{solve}, (in seconds) as a function of problem size (number of unknowns, N_{dof}), for the exact and optimal implementation of the pseudo-solid preconditioner, for two different Poisson ratios and degrees of deformation.

the fluid mesh) on the discretised Navier–Stokes equations – this incorporates the so-called shape-derivatives and the time-discretised version of the no-slip condition (8) which, as discussed in Section 2.2, expresses the fluid velocities in terms of the nodal velocities of the fluid mesh. C_{sf} captures the effect of the fluid traction (pressure and shear stresses) on the fluid-loaded solid. Since the shear stresses depend on gradients of the fluid velocity (see (6)), the traction is also affected by the nodal positions in the fluid mesh – this dependency gives rise to the matrix C_{sx}. Finally, C_{ls} arises because the prescribed boundary displacement of the fluid mesh, $\mathbf{R}_{\text{FSI}}$, in (9), now depends on the displacement field in the fluid-loaded solid. The zero (1,2) block reflects the fact that the discretised Navier–Stokes equations do not depend directly on the displacement field of the actual solid. The zero (3,1) and (3,2) blocks indicate that the pseudo-solid equations are not affected by the fluid unknowns. The (3,2) block is zero because the real solid affects the pseudo-solid only indirectly via the Lagrange multiplier constraint (which gives rise to C_{ls}). Finally, the non-zero blocks in the last block column reflect the fact that the Lagrange multiplier is only used to enforce the displacement constraint (9) in a one-way coupling in which the real solid affects the pseudo-solid but not vice versa.

4.1. Derivation of the FSI preconditioner

To construct a preconditioner for the linear system (32) we replace the bottom-right 2×2 block in the Jacobian with the pseudo-solid preconditioner, $\mathcal{P}_{\text{PS}}$, defined in (19). This yields the pseudo-solid FSI preconditioner

$$\mathcal{P}_{\text{FSI}} = \left[\begin{array}{ccc|c} F & & & C_{\text{fx}} \\ C_{\text{sf}} & S & & C_{\text{sx}} \\ \hline & & & E_{\text{aug}} \\ & C_{\text{ls}} & & W \end{array}\right]. \tag{33}$$

Permuting $\mathcal{P}_{\text{FSI}}$ with

$$Y = \begin{bmatrix} & & I_{n_{\text{x}}} & \\ I_{n_{\text{f}}} & & & \\ & I_{n_{\text{s}}} & & \\ & & & I_{n_{\text{b}}} \end{bmatrix}, \tag{34}$$

where I_m is an $m \times m$ identity matrix, yields

$$\mathcal{P}_{\text{FSI}}\, Y^T = \begin{bmatrix} E_{\text{aug}} & & & \\ C_{\text{fx}} & F & & \\ C_{\text{sx}} & C_{\text{sf}} & S & \\ & & C_{\text{ls}} & W \end{bmatrix}, \tag{35}$$

which shows that the preconditioner has a block triangular structure. The application of the preconditioner $\mathcal{P}_{\text{FSI}}$ therefore requires the solution of four subsidiary linear systems involving the matrices F, S, E_{aug} and W and four sparse matrix–vector products with the interaction matrices C_{fx}, C_{sx}, C_{sf} and C_{ls}. The ability to permute this system to lower block triangular form is a direct consequence of (i) the use of the Lagrange multiplier for the imposition of the displacement constraint (9), and (ii)

the use of the block diagonal pseudo-solid preconditioner $\mathcal{P}_{PS}$ as an approximation for the saddle-point pseudo-solid Jacobian J_{PS}.

4.2. Spectral properties of the FSI preconditioner

We assess the quality of the preconditioner by analysing the spectrum of the preconditioned matrix $\mathcal{P}_{FSI}^{-1}J_{FSI}$. For this purpose we split the FSI Jacobian defined in (32) into 2×2 sub-blocks,

$$J_{FSI}=\begin{bmatrix} J_{11} & J_{12}\\ J_{21} & J_{22}\end{bmatrix}=\left[\begin{array}{ccc|cc} F & & & C_{fx_i} & C_{fx_b}\\ C_{sf} & S & & C_{sx_i} & C_{sx_b}\\ & & & E_{ii} & E_{ib}\\ \hline & & & E_{bi} & E_{bb} \quad M\\ & C_{ls} & & M & \end{array}\right]\begin{array}{l}\updownarrow n_f\\ \updownarrow n_s\\ \updownarrow n_x-n_b\\ \updownarrow n_b\\ \updownarrow n_b\end{array}, \tag{36}$$

where we have sub-divided the pseudo-solid degrees of freedom (representing the nodal positions in the fluid mesh) into constrained and unconstrained values (identified by the subscripts b and i, respectively), and M is defined in (21). Using this sub-division, we rewrite the FSI preconditioner (33) as

$$\mathcal{P}_{FSI}=\begin{bmatrix} J_{11} & J_{12}\\ J_{21} & J_{22}+V\end{bmatrix}, \tag{37}$$

where

$$V=\begin{bmatrix} \sigma I & -M\\ -M & \frac{1}{\sigma}M^2\end{bmatrix}. \tag{38}$$

This allows the preconditioned matrix $\mathcal{P}_{FSI}^{-1}J_{FSI}$ to be written as

$$\mathcal{P}_{FSI}^{-1}J_{FSI}=\begin{bmatrix} I & S_{11}^{-1}J_{12}J_{22}^{-1}V(S_{22}+V)^{-1}S_{22}\\ & (S_{22}+V)^{-1}S_{22}\end{bmatrix}, \tag{39}$$

where $S_{22}=J_{22}-J_{21}J_{11}^{-1}J_{12}$ is the Schur complement associated with the block J_{22}. The identity matrix in the top left block in Eq. (39) shows that the preconditioned matrix $\mathcal{P}_{FSI}^{-1}J_{FSI}$ has at least $n_f+n_s+n_x-n_b$ unit eigenvalues, with the remaining $2n_b$ eigenvalues determined by the spectrum of $(S_{22}+V)^{-1}S_{22}$.

Using the identity

$$(S_{22}+V)^{-1}=\begin{bmatrix} (S_{bb}+\sigma I)^{-1} & \\ \sigma M^{-2}C_{ls}U(S_{bb}+\sigma I)^{-1} & \sigma M^{-2}\end{bmatrix}, \tag{40}$$

where $S_{bb}=E_{bb}-E_{bi}E_{ii}^{-1}E_{ib}$ is the Schur complement associated with the block E_{bb}, and

$$U=S^{-1}(C_{sx_b}-C_{sf}F^{-1}C_{fx_b}+(C_{sf}F^{-1}C_{fx_i}-C_{sx_i})E_{ii}^{-1}E_{ib}) \tag{41}$$

then shows that the remaining $2n_b$ eigenvalues are determined by the eigenvalue problem

$$\begin{bmatrix} R_{11} & R_{12}\\ R_{21} & R_{22}\end{bmatrix}\begin{bmatrix}\mathbf{w}_b\\ \mathbf{w}_l\end{bmatrix}=\lambda\begin{bmatrix}\mathbf{w}_b\\ \mathbf{w}_l\end{bmatrix}, \tag{42}$$

where

$$\begin{aligned} R_{11}&=(S_{bb}+\sigma I)^{-1}S_{bb},\\ R_{12}&=(S_{bb}+\sigma I)^{-1}M,\\ R_{21}&=\sigma M^{-2}C_{ls}U\Big((S_{bb}+\sigma I)^{-1}S_{bb}-I\Big)+\sigma M^{-1},\\ R_{22}&=\sigma M^{-2}C_{ls}U(S_{bb}+\sigma I)^{-1}M.\end{aligned}$$

Straightforward block elimination of (42) yields

$$(\lambda-\lambda^2)\frac{1}{\sigma}S_{bb}\mathbf{w}_b=((1-\lambda)M^{-1}C_{ls}U-(1-\lambda^2)I)\mathbf{w}_b, \tag{43}$$

which reveals the existence of further unit eigenvalues, with the remaining (at most n_b) non-unit eigenvalues determined by the generalised eigenvalue problem

$$S_{bb}\mathbf{w}_b=\frac{\sigma}{\lambda}(M^{-1}C_{ls}U-(1+\lambda)I)\mathbf{w}_b. \tag{44}$$

We were not able to derive rigorous bounds on these few remaining eigenvalues and will therefore employ numerical experiments to assess the quality of $\mathcal{P}_{\mathrm{FSI}}$. However, it interesting to note that for $Q = 0$ (which implies $U = 0$ because $C_{\mathrm{sx_i}} = 0$, $C_{\mathrm{sx_b}} = 0$ and $C_{\mathrm{sf}} = 0$), Eq. (44) is equivalent to Eq. (26) in the pseudo-solid case. Therefore, subject to the constraint that E is symmetric positive definite, the non-unit eigenvalues of $\mathcal{P}_{\mathrm{FSI}}^{-1}J_{\mathrm{FSI}}$ are also bounded by (27), suggesting that for sufficiently weak fluid–structure interaction (small values of Q), these bounds provide a good approximate characterisation of the spectrum of the preconditioned matrix.

4.3. Numerical evaluation of the FSI preconditioner and development of an optimal implementation

To evaluate the performance of the FSI preconditioner, we first consider the two-dimensional problem shown in Fig. 5: Viscous fluid is driven through a finite-length channel that is partially obstructed by an elastic leaflet which we model as a massless, thin-walled Kirchhoff–Love beam. We non-dimensionalise the problem, using the channel width as the lengthscale $\mathcal{L}$, while non-dimensionalising the velocity on the maximum inflow velocity so that the imposed steady inflow profile is given by $\mathbf{u}(x_1 = 0, x_2) = \mathbf{u}_{\mathrm{steady}}(x_2) = 4x_2(1 - x_2)\mathbf{e}_1$. When considering time-dependent problems we subject the inflow profile to a time-periodic variation, such that $\mathbf{u}(x_1 = 0, x_2, t) = \mathbf{u}_{\mathrm{steady}}(x_2)(1 + \frac{1}{2}\sin(2\pi t))$ where time has been non-dimensionalised

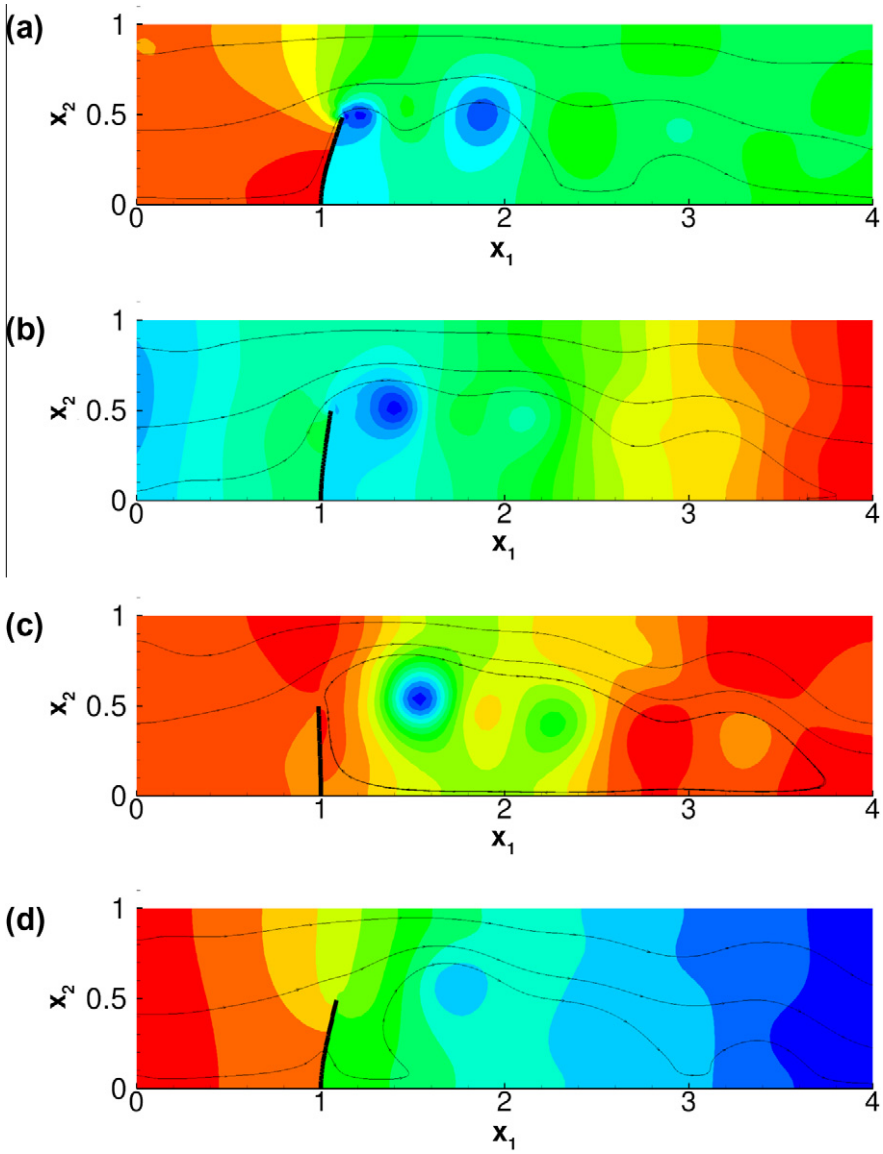

Fig. 5. Illustration of the flow field (pressure contours and instantaneous streamlines) for unsteady flow past a thin-walled elastic leaflet ($\mathfrak{h}/\mathcal{L} = 0.05$) at a Reynolds number of $Re = 200$ for $Q = 2 \times 10^{-7}$ and $\Lambda = 0$. (a) $t = 1.00$, (b) $t = 1.25$, (c) $t = 1.50$, and (d) $t = 2.00$.

Table 5
Iteration counts for the GMRES solver preconditioned with the exact pseudo-solid FSI preconditioner applied to the steady and unsteady simulation of the two-dimensional FSI channel with leaflet problem.

N_{dof} / h	38,724 / $\frac{1}{24}$	155,784 / $\frac{1}{48}$	351,180 / $\frac{1}{72}$	624,912 / $\frac{1}{96}$	
$Re = 50$	6.7	7.5	8.2	8.3	
$Re = 100$	6.5	6.7	7.5	8.7	(steady)
$Re = 200$	7.3	7.0	8.5	8.5	
$Re = 50$	7.3	7.6	7.8	8.0	
$Re = 100$	7.2	7.5	7.7	8.2	(unsteady)
$Re = 200$	7.4	7.9	8.2	8.5	

on the flow's intrinsic timescale, $\mathcal{T} = \mathcal{L}/\mathcal{U}$, corresponding to $St = 1$. We impose parallel, axially traction-free outflow and clamp the leaflet at its base, $x_2 = 0$.

We discretised the Navier–Stokes equations with quadrilateral Taylor–Hood (Q2Q1) elements and used one-dimensional Hermite elements to discretise the one-dimensional version of the variational principle (5). The simulations were then carried out in two stages. First we performed a steady simulation in which we increased the Reynolds number from zero to its target value in increments of 25. The steady solution at the target Reynolds number was then used as the initial condition for an unsteady simulation in which the inflow profile was subjected to the time-periodic variation specified above. We computed the flow field over three periods of the time-periodic inflow (from $t = 0$ to $t = 3$), evaluating the time-derivatives in the fluid and solid equations with BDF2 and Newmark schemes, respectively, using a fixed timestep of $\Delta t = 1/40$.

4.3.1. Performance of the exact preconditioner

Table 5 shows the average GMRES iteration counts (averaged over all linear solves performed in the course of all Newton iterations, for a GMRES convergence tolerance of $\tau = 10^{-8}$) for the exact version of the preconditioner in which the subsidiary linear systems involving F, S, E_{aug} and W are solved using SuperLU. The iteration counts in the steady and unsteady

Table 6
Spectral bounds of the n_{nu} non-unit eigenvalues of the preconditioned matrix $\mathcal{P}_{FSI}^{-1}J_{FSI}$ for the steady solution of the FSI channel with leaflet problem at $Re = 50$.

N_{dof} / h	1014 / $\frac{1}{2}$	4204 / $\frac{1}{4}$	9570 / $\frac{1}{6}$	17,112 / $\frac{1}{8}$
n_{nu}	14	30	46	62
$\min(\Re(\lambda))$	−0.9722	−0.9861	−0.9907	−0.9932
$\max(\Re(\lambda))$	−0.8345	−0.8356	−0.8356	−0.8354
$\min(\Im(\lambda))$	−0.0015	−0.0035	−0.0032	−0.0025
$\max(\Im(\lambda))$	0.0015	0.0035	0.0032	0.0025

Table 7
Iteration counts for the GMRES solver, preconditioned with the intermediate inexact pseudo-solid FSI preconditioner applied to the steady and unsteady solution of the two-dimensional FSI channel with leaflet problem.

N_{dof} / h	38,724 / $\frac{1}{24}$	155,784 / $\frac{1}{48}$	351,180 / $\frac{1}{72}$	624,912 / $\frac{1}{96}$	
$Re = 50$	15.0	17.5	18.0	18.0	
$Re = 100$	14.2	16.2	16.5	19.2	(steady)
$Re = 200$	16.6	17.3	21.5	20.7	
$Re = 50$	15.9	16.9	17.4	17.9	
$Re = 100$	15.3	16.9	17.4	18.2	(unsteady)
$Re = 200$	15.6	17.0	17.7	18.5	

Table 8
Iteration counts for the GMRES solver preconditioned with the inexact pseudo-solid FSI preconditioner applied to the steady and unsteady solution of the two-dimensional FSI channel with leaflet problem.

N_{dof} / h	38,724 / $\frac{1}{24}$	155,784 / $\frac{1}{48}$	351,180 / $\frac{1}{72}$	624,912 / $\frac{1}{96}$	
$Re = 50$	51.5	68.0	78.5	86.0	
$Re = 100$	60.0	72.2	87.7	101.5	(steady)
$Re = 200$	149.0	98.2	108.5	118.6	
$Re = 50$	27.5	30.6	31.7	32.8	
$Re = 100$	27.2	30.4	31.8	33.3	(unsteady)
$Re = 200$	30.6	32.2	33.0	33.6	

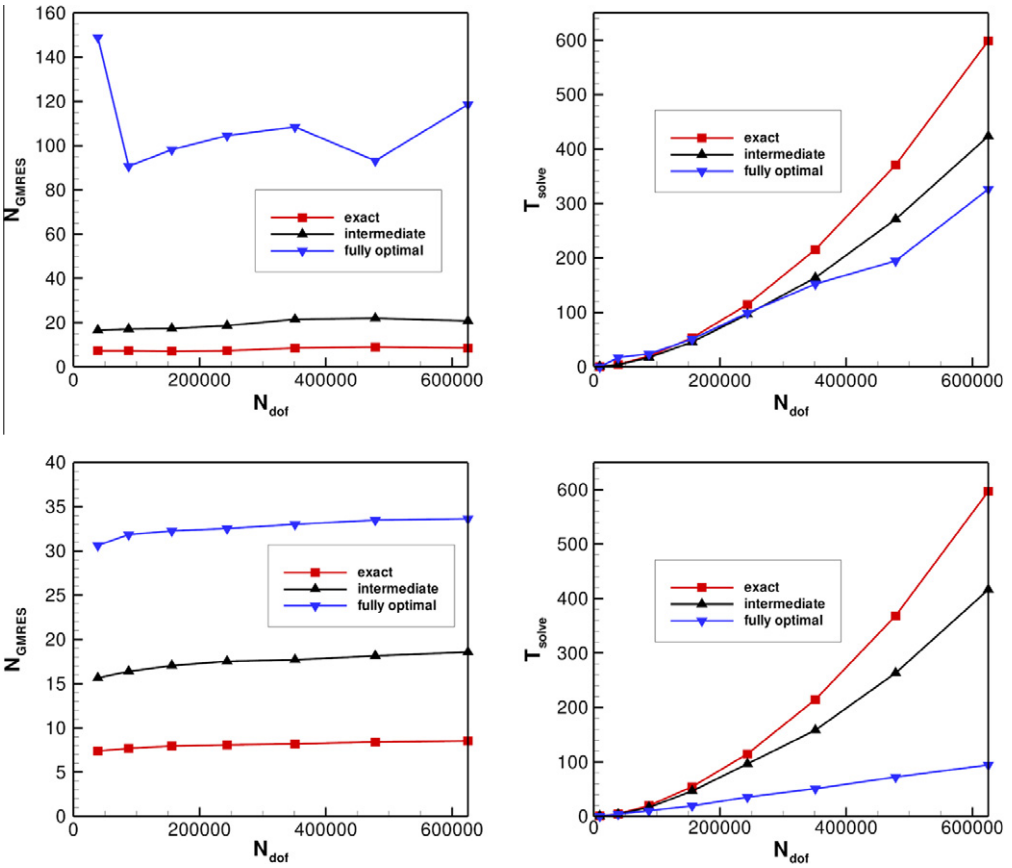

Fig. 6. GMRES iteration counts (left) and solve times (in seconds, right) for the three different implementations of the FSI preconditioner when used in the steady (top) and unsteady (bottom) simulations of the two-dimensional test problem at $Re = 200$.

simulations are remarkably low and display only a very weak dependence on the Reynolds number and mesh size. This suggests that, even though the analysis presented in Section 4.1 is incomplete, the largest and smallest eigenvalues, λ, of the preconditioned matrix $\mathcal{P}_{FSI}^{-1}J_{FSI}$ remain bounded under mesh refinement. We confirmed this by computing the spectrum of $\mathcal{P}_{FSI}^{-1}J_{FSI}$ for a number of modest-sized problems. Table 6 shows that the maximum and minimum real and imaginary parts of the n_{nu} non-unit eigenvalues of $\mathcal{P}_{FSI}^{-1}J_{FSI}$ remain in a small bounded region of the complex plane. The bounds have a very weak dependence on the problem size and are far from the origin, consistent with the observation that the GMRES iteration counts are virtually mesh independent.

4.3.2. Development of an optimal preconditioner

We start the development of an optimal implementation of the preconditioner by replacing the exact solution of the linear systems involving E_{aug} and W by the application of the approximate pseudo-solid preconditioner $\widetilde{\mathcal{P}}_{PS}$ discussed in Section 3.2. Table 7 shows that the use of this intermediate version of the preconditioner causes a small increase in the absolute iteration counts while retaining the weak dependence on both mesh spacing h and Reynolds number Re. With this implementation the most computationally expensive phase of the application of the preconditioner is the solution of the linear

Table 9
Average GMRES iteration count, N_{GMRES}, as a function of the timestep, Δt, and a range of values of the wall inertia parameter Λ^2 for the GMRES solver preconditioned with the inexact pseudo-solid FSI preconditioner, applied to the unsteady solution of the two-dimensional FSI channel with leaflet problem with $Re = 100$ and a spatial discretisation involving 38,724 degrees of freedom.

Δt	0.0125	0.025	0.05	0.1
$\Lambda^2 = 0.0$	25.3	27.6	30.6	34.1
$\Lambda^2 = 0.1$	22.1	24.5	27.1	29.6
$\Lambda^2 = 1.0$	21.7	23.2	25.5	29.9
$\Lambda^2 = 10.0$	20.7	22.3	24.3	28.4

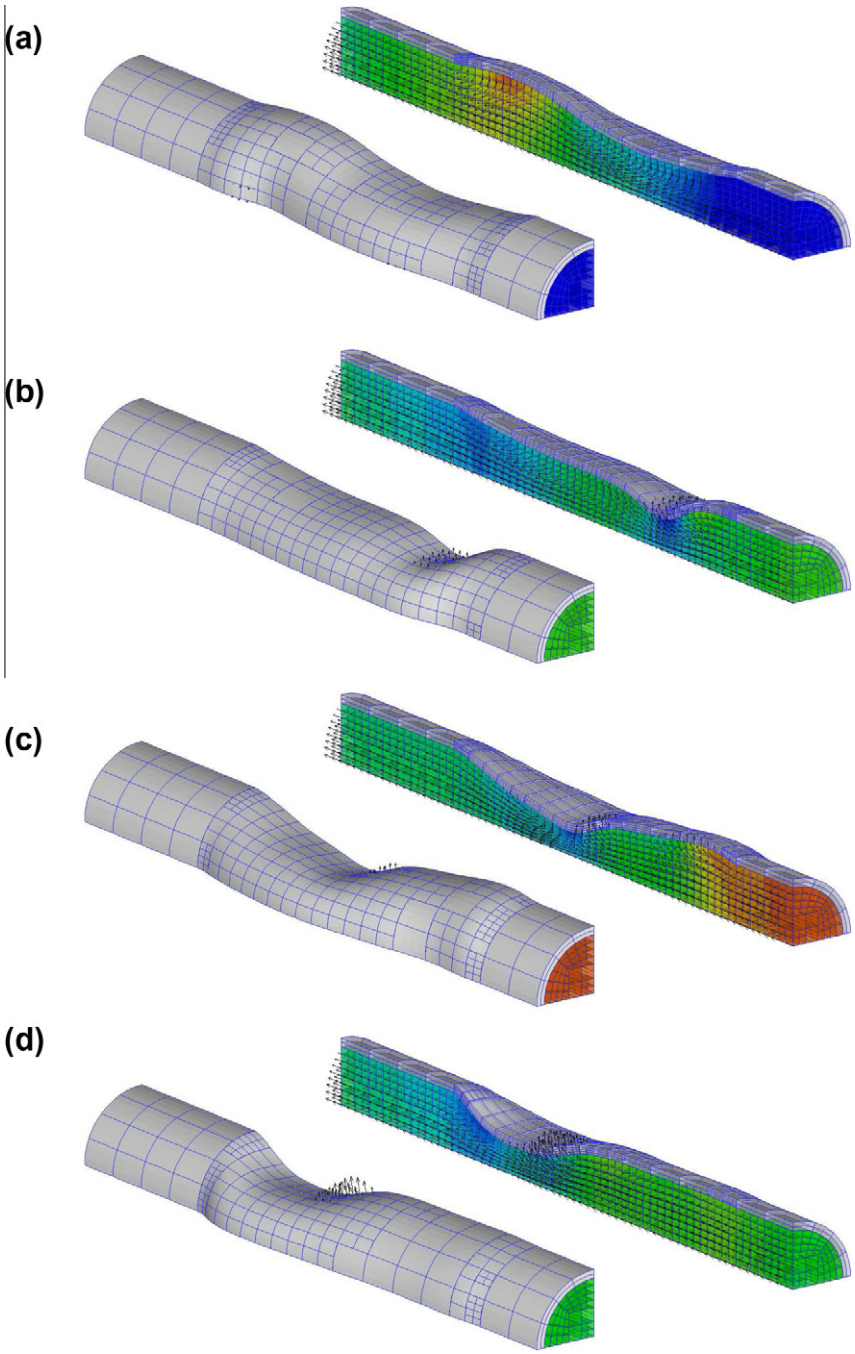

Fig. 7. Snapshots of the flow field (velocity vectors and pressure contours) and wall shapes during the forced oscillation of a collapsible tube, subject to time-periodic inflow, and an external pressure of $p_{ext} = 2 \times 10^{-4}$. Length of the elastic and rigid upstream and downstream tube sections, $L_{elastic}/\mathfrak{a} = 12$, $L_{up}/\mathfrak{a} = 1.5$, $L_{down}/\mathfrak{a} = 3$, respectively; wall thickness $\mathfrak{h}/a = 0.15$; Poisson's ratio $\nu = 0.3$; wall density $\Lambda = 0$; Reynolds number $Re = 100$. (a) $t = 3.375$, (b) $t = 3.875$, (c) $t = 4.375$, and (d) $t = 4.875$.

system involving the Navier–Stokes Jacobian, F.[2] Following the approach we adopted in the development of our preconditioner for FSI problems with algebraic node updates [28,19], we replace the action of F^{-1} by the application of Elman, Silvester and

[2] The use of a sparse direct solver for the linear system involving the tangent stiffness matrix S of the Kirchhoff Love beam does not affect the optimality of the preconditioner. This is because S arises from the discretisation of a spatially one-dimensional sub-problem. Therefore under uniform mesh refinement the size of S scales as $\mathcal{O}(N_{dof}^{1/2})$. When combined with a sparse direct solver (with a computational cost that scales as $\mathcal{O}(m^2)$ for a linear system of size m) this still yields a solution at a cost proportional to N_{dof}.

Wathen's Least-Squares Commutator (LSC) Navier–Stokes preconditioner (see [46]; details of the specific implementation are given in Appendix A).

Table 8 shows the GMRES iteration counts obtained with this final version of the FSI preconditioner. For the steady simulations, the introduction of the LSC preconditioner leads to a significant increase in N_{GMRES}, with a slightly erratic dependence on the mesh size and a noticeable dependence on the Reynolds number. Conversely, in the unsteady simulations the introduction of the LSC preconditioner merely causes a modest increase in the iteration counts while retaining its weak dependence on Re and h. This is entirely consistent with the behaviour of the LSC preconditioner in equivalent single-physics Navier–Stokes problems; see e.g. [46].

The plots of the GMRES iteration counts and solution times (for the simulations at $Re = 200$) shown in Fig. 6 illustrate that the low and virtually mesh independent iteration counts obtained for the exact and intermediate implementations of the preconditioner do not result in optimal solution times because the application of these preconditioners still requires the solution of subsidiary linear systems by sparse direct solvers. Conversely, in the fully-optimal implementation of the preconditioner the solution time exhibits nearly optimal asymptotic scaling in the sense that for sufficiently large problem sizes, the solve times increase approximately linearly with the number of unknowns. The steady computations are significantly more costly than their unsteady counterparts due the much higher iteration counts.

Finally, Table 9 demonstrates that the behaviour reported above is robust with respect to variations in the timestep, Δt, and the presence or absence of solid inertia (characterised by the parameter Λ^2).

4.4. A three-dimensional test problem

To assess how well the preconditioner performs in 3D problems we performed a simulation of the collapsible tube problem shown in Fig. 7. An elastic circular tube of inner radius $\mathfrak{a}$ and thickness $\mathfrak{h}$ is mounted between two rigid sections and conveys viscous fluid that is driven by an imposed, pulsatile inflow at the far upstream end. We non-dimensionalised all lengths on the inner radius of the tube, so that $\mathcal{L} = \mathfrak{a}$, and scaled the velocity on the (time-)average inflow velocity on the tube's centreline so that the imposed inflow profile is given by $\mathbf{u}(z = 0, r, t) = 2r(1 - r)(1 + \frac{1}{2}\sin(2\pi t))\mathbf{e}_z$. At the far downstream end, we imposed parallel, axially traction-free outflow. We modelled the tube as a massless 3D solid, loaded by the fluid traction and an external pressure p_{ext} (non-dimensionalised on the wall's Young's modulus, so that $\mathcal{S} = E$) which we modulated by a small azimuthal imperfection to induce the tube's non-axisymmetric buckling. Fig. 7 shows snapshots of the large-amplitude oscillation that develops from an initial condition of Poiseuille flow through the undeformed tube for a Reynolds number of $Re = 100$. We discretised a quarter of the tube, exploiting its twofold symmetry.

To systematically assess the mesh-dependence of the GMRES iteration counts we initially performed the entire simulation with a relatively coarse, uniform discretisation. We then restarted the computation at selected timesteps (corresponding to the configurations shown in Fig. 7) and performed a number of uniform mesh refinements before continuing the simulation for a small number of timesteps. All computations were performed using the optimal implementation of the preconditioner, in which we performed the solution of the subsidiary linear systems (apart from those involving the solid Jacobian S) as described in the previous section. In the present problem the fluid-loaded solid is three-dimensional, therefore the use of a direct solver for the linear systems involving S is no longer optimal. We therefore employed the same block triangular preconditioner with AMG block solves that we used for the pseudo-solid mesh update system (30). The average GMRES iteration counts (again for a GMRES convergence tolerance of $\tau = 10^{-8}$) observed in these simulations are reported in Table 10 which shows a very modest increase in iteration counts despite the dramatic increase in the overall number of degrees of freedom under uniform mesh refinement.

In practical applications it is rarely necessary to refine the mesh uniformly to fully resolve the flow field, especially in cases where sharp spatial gradients arise only in shear or boundary layers, say. In such cases, the use of spatial adaptivity becomes essential. We assessed the performance of the preconditioner in spatially adaptive computations by performing the entire simulation yet again, controlling the adaptive mesh refinement in the fluid and solid meshes with error estimates obtained from Zhu and Zienkiewiz's flux recovery procedure [47], using the components of the solid strain and the fluid rate of strain tensors as the generalised fluxes. The dynamic mesh adaptation resulted in an average of 138,631 unknowns (varying between 117,666 and 164,169 unknowns) over the course of the simulation. (For comparison, an equivalent, spatially uniform mesh refinement would have resulted in 358,114 unknowns). GMRES, preconditioned by the optimal implementa-

Table 10
Average GMRES iteration count, N_{GMRES}, for the GMRES solver preconditioned with the inexact pseudo-solid FSI preconditioner, applied to the unsteady solution of the three-dimensional collapsible tube for different levels $N_{\mathrm{unif.ref.}}$ of uniform mesh refinement, resulting in the specified number of unknowns. All simulations were restarted from the equivalent of the configurations shown in Fig. 7 and N_{GMRES} represents the average of all linear solves when computing the subsequent three timesteps.

N_{dof} / $N_{\mathrm{unif.\ ref.}}$	6178 / 0	46,228 / 1	358,114 / 2	2,820,262 / 3
$t_{\mathrm{restart}} = 3.375$	65.00	66.75	74.50	82.50
$t_{\mathrm{restart}} = 3.875$	61.00	65.75	77.12	84.88
$t_{\mathrm{restart}} = 4.375$	61.75	68.75	80.00	90.50
$t_{\mathrm{restart}} = 4.875$	62.25	71.50	81.50	88.25

tion of the FSI preconditioner, converged within an average of $N_{\mathrm{GMRES}} = 78.3$ iterations, consistent with the behaviour observed in computations on uniformly refined meshes.

5. Summary and discussion

We have developed and analysed a preconditioner for the efficient solution of the linear systems that arise when Newton's method is employed to solve monolithically-coupled large-displacement fluid–structure problems in which the moving, body-fitted fluid mesh is updated by the equations of (pseudo-)elasticity. The preconditioner is applicable to problems in which the fluid-loaded solid is described by the equations of elasticity or thin-shell/beam theory. We demonstrated that in its exact or intermediate implementation (in which the subsidiary linear systems involving the fluid and solid Jacobians are solved exactly), use of the preconditioner results in mesh-independent GMRES iteration counts. An optimal implementation of the preconditioner requires the replacement of these costly exact linear solves by approximate solvers/preconditioners. An important feature of our FSI preconditioner is that it lends itself to a modular implementation which allows these tasks to be performed by existing single-physics preconditioners. The approach of reusing solvers for single-physics problems at the linear algebra (specifically, preconditioning) level is in direct contrast to the approach adopted in partitioned methods where the re-use of the solvers takes place at the level of the (black-box) non-linear sub-problems.

We presented a particular implementation of our approach, using the Least Squares Commutator preconditioner to approximately solve the linear systems involving the Navier–Stokes Jacobian, and a block triangular preconditioner for the linear systems involving the solid Jacobian. In all single-physics preconditioners the approximate solution of the linear systems involving the principal diagonal blocks and the associated approximate Schur complements was performed by a small, fixed number of AMG *V*-cycles with standard point smoothers. This implementation allowed the application of the preconditioner at a cost that is linear in the number of unknowns, while retaining an approximately constant GMRES iteration count under uniform mesh refinement, resulting in a near-optimal overall solver. The preconditioning strategy was found to work equally well in simulations in which spatial adaptivity was employed to refine the meshes non-uniformly in 3D.

The results obtained in the test cases reported in this paper are fairly typical for the behaviour of the preconditioner in many other FSI problems [45]. A common feature observed in all these tests is that the behaviour of the FSI preconditioner (e.g. the dependence of the GMRES iteration counts on the mesh size, and on physical problem parameters such as the Reynolds number) tends to mirror the behaviour of the constituent subsidiary (fluid and solid) preconditioners when used in comparable single-physics problems, implying that improvements to the single-physics preconditioners will directly benefit the performance of the FSI preconditioner.

The use of Lagrange multipliers for the imposition of the displacement constraint (9) for the pseudo-solid, and the use of an augmented Lagrangian preconditioning strategy for the solution of the linear system (15) are crucial for the development and the theoretical analysis of the preconditioner for the linear systems that arise from the monolithic discretisation of FSI problems. However, we note that the preconditioning strategy for the solution of the isolated pseudo-solid problem discussed in Section 3 can also be used to update the fluid mesh when solving FSI problems by partitioned approaches. In fact, numerical experiments show that the imposition of the boundary displacements by Lagrange multipliers is far more robust than the direct imposition of the displacement boundary conditions as Dirichlet conditions. This is because in the latter approach the imposition of relatively large boundary deflections (compared to the size the of the fluid elements next the FSI boundary) can create inverted elements before a solution is even attempted; the Lagrange-multiplier-based imposition of the displacement boundary conditions applies the displacement constraint more gently.

Finally we mention that all the test cases presented in this paper are scheduled for inclusion into the next release of `oomph-lib`, scheduled for 2012.

Acknowledgements

The authors wish to acknowledge many helpful discussions with David Silvester and Andrew Hazel. We are also grateful for many comments from an anonymous referee which substantially improved the paper.

Appendix A. The Least Squares Commutator preconditioner

Here we provide brief details of the Least Squares Commutator (LSC) Navier–Stokes preconditioner [46] and its approximate implementation used in the computations presented in Section 4.1. The preconditioner is applicable to the solution of the linear systems

$$\begin{bmatrix} A & B^T \\ B & \end{bmatrix} \begin{pmatrix} \Delta\mathbf{u} \\ \Delta\mathbf{p} \end{pmatrix} = \begin{pmatrix} -\mathbf{r}_{\mathrm{u}} \\ -\mathbf{r}_{\mathrm{p}} \end{pmatrix} \tag{A.1}$$

that arise in the Newton-based solution of Navier–Stokes problems, discretised with LBB-stable finite elements (such as the Taylor–Hood Q2Q1 elements used throughout this paper). In equation (A.1) we have grouped the unknowns into the vectors of discrete velocities, $\mathbf{u}$, and pressures, $\mathbf{p}$. In its exact form, the LSC preconditioner is given by

$$\mathcal{P}_{\mathrm{LSC}} = \begin{bmatrix} A & B^T \\ & -W_s \end{bmatrix}, \tag{A.2}$$

where

$$W_s = (BQ_{\mathrm{diag}}^{-1}B^T)(BQ_{\mathrm{diag}}^{-1}AQ_{\mathrm{diag}}^{-1}B^T)^{-1}(BQ_{\mathrm{diag}}^{-1}B^T) \tag{A.3}$$

is an approximation to the Schur complement, $BA^{-1}B^T$. The approximate Schur complement W_s is formed from the block matrices B, A and B^T that occur naturally in the Jacobian matrix, and the cheap-to-compute diagonal of the velocity mass matrix, Q_{diag}. The application of the preconditioner requires the solution of two linear systems involving the pressure-Poisson-like matrix $BQ_{\mathrm{diag}}^{-1}B^T$, one matrix vector product with $BQ_{\mathrm{diag}}^{-1}AQ_{\mathrm{diag}}^{-1}B^T$ (performed via three separate matrix–vector products and two scaling operations), and the solution of a linear system involving the momentum block, A.

An optimal approximate implementation of the preconditioner (in which the cost of each application is proportional to the number of unknowns) is obtained by approximating the two pressure-Poisson solves involving $BQ_{\mathrm{diag}}^{-1}B^T$ by two $V(1,1)$ AMG cycles. In the computations presented in this paper we employed BoomerAMG, Hypre's implementation of AMG [43], using Ruge–Stüben coarsening (with a strength parameter of 0.25) and Jacobi smoothing (with a damping factor of $\frac{2}{3}$). The linear systems involving the momentum block A were solved by replacing A by an upper block triangular approximation (sub-dividing the discrete velocities according to their coordinate directions), and approximately solving the linear systems involving the diagonal blocks with AMG (as above, but without damping).

References

[1] J.B. Grotberg, O.E. Jensen, Biofluid mechanics in flexible tubes, Annual Review of Fluid Mechanics 36 (2004) 121–147.
[2] M. Heil, A.L. Hazel, Fluid-structure interaction in internal physiological flows, Annual Review of Fluid Mechanics 43 (2011) 141–162.
[3] V. Kalro, T. Tezduyar, A parallel 3D computational method for fluid–structure interactions in parachute systems, Computer Methods in Applied Mechanics and Engineering 190 (2000) 321–332.
[4] K.-U. Bletzinger, R. Wüchner, A. Kupzok, Algorithmic treatment of shells and free form-membranes in FSI, in: M. Schäfer, H.-J. Bungartz (Eds.), Fluid-Structure Interaction, Springer, 2006, pp. 336–355.
[5] L.P. Dasi, H.A. Simon, P. Sucosky, A.P. Yoganathan, Fluid mechanics of artificial heart valves, Clinical and Experimental Pharmacology and Physiology 36 (2009) 225–237.
[6] B.M. Irons, R.C. Tuck, A version of the Aitken accelerator for computer iteration, International Journal of Numerical Methods in Engineering 1 (1969) 275–277.
[7] D.P. Mok, W.A. Wall, Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures, in: W.A. Wall, K.-U. Bletzinger, K. Schweizerhof (Eds.), Trends in Computational Structural Mechanics, CIMNE, Barcelona, Spain, 2001.
[8] M. Heil, Stokes flow in an elastic tube – a large-displacement fluid–structure interaction problem, International Journal for Numerical Methods in Fluids 28 (1998) 243–265.
[9] U. Kuettler, W.A. Wall, Fixed-point fluid–structure interaction solvers with dynamic relaxation, Computational Mechanics 43 (2008) 61–72.
[10] P. Causin, J. Gerbeau, F. Nobile, Added-mass effect in the design of partitioned algorithms for fluid–structure problems, Computer Methods in Applied Mechanics and Engineering 194 (2005) 4506–4527.
[11] C. Farhat, P. Geuzaine, Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids, Computer Methods in Applied Mechanics and Engineering 193 (2004) 4073–4095.
[12] C. Farhat, K. van der Zee, P. Geuzaine, Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity, Computer Methods in Applied Mechanics and Engineering 195 (2006) 1973–2001.
[13] E. Burman, M.A. Fernndez, Stabilization of explicit coupling in fluid–structure interaction involving fluid incompressibility, Computer Methods in Applied Mechanics and Engineering 198 (2009) 766–784.
[14] G. Guidoboni, R. Glowinski, N. Cavallini, S. Canic, Stable loosely-coupled-type algorithm for fluid–structure interaction in blood flow, Journal of Computational Physics 228 (2009) 6916–6937.
[15] M. Fernandez, M. Moubachir, A Newton method using exact jacobians for solving fluid–structure coupling, Computers and Structures 83 (2005) 127–142.
[16] M. Heil, A.L. Hazel, `oomph-lib` – an object-oriented multi-physics finite-element library, in: M. Schäfer, H.-J. Bungartz (Eds.), Fluid–Structure Interaction, Springer, 2006, pp. 19–49. `oomph-lib` is available as open-source software at `http://www.oomph-lib.org`.
[17] C. Förster, W.A. Wall, E. Ramm, Artificial mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows, Computer Methods in Applied Mechanics and Engineering 196 (2007) 1278–1293.
[18] M.A. Fernandez, J.-F. Gerbeau, C. Grandmont, A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid, International Journal of Numerical Methods in Engineering 69 (2007) 794–821.
[19] M. Heil, A.L. Hazel, J. Boyle, Solvers for large-displacement fluid-structure interaction problems: segregated vs. monolithic approaches, Computational Mechanics 43 (2008) 91–101.
[20] U. Kuettler, M. Gee, C. Foerster, A. Comerford, W.A. Wall, Coupling strategies for biomedical fluid–structure interaction problems, International Journal for Numerical Methods in Biomedical Engineering 26 (2010) 305–321.
[21] H.G. Matthies, J. Steindorf, Partitioned but strongly coupled iteration schemes for nonlinear fluid–structure interaction, Computers and Structures 80 (2002) 1991–1999.
[22] H.G. Matthies, J. Steindorf, Partitioned strong coupling algorithms for fluid–structure interaction, Computers and Structures 81 (2003) 805–812.
[23] S. Badia, A. Quaini, A. Quarteroni, Modular vs. non-modular preconditioners for fluid–structure systems with large added-mass effect, Computer Methods in Applied Mechanics and Engineering 197 (2008) 4216–4232.
[24] S. Badia, F. Nobile, C. Vergara, Robin–Robin preconditioned Krylov methods for fluid–structure interaction problems, Computer Methods in Applied Mechanics and Engineering 198 (2009) 2768–2784.
[25] W. Dettmer, D. Perić, A computational framework for fluid–structure interaction: finite element formulation and applications, Computer Methods in Applied Mechanics and Engineering 195 (2006) 5754–5779.
[26] A. Greenbaum, Iterative Methods for Solving Linear Systems, SIAM, 1997.
[27] T.J.R. Hughes, W.K. Liu, T. Zimmerman, An arbitrary Lagrangian–Eulerian finite element formulation for incompressible viscous flow, Computer Methods in Applied Mechanics and Engineering 29 (1981) 329–349.
[28] M. Heil, An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems, Computer Methods in Applied Mechanics and Engineering 193 (2004) 1–23.

[29] P. Sackinger, P. Schunk, R. Rao, A Newton–Raphson pseudo-solid domain mapping technique for free and moving boundary problems: a finite element implementation, Journal of Computational Physics 125 (1996) 83–103.
[30] M.W. Gee, U. Küttler, W.A. Wall, Truly monolithic algebraic multigrid for fluid–structure interaction, International Journal for Numerical Methods in Engineering 85 (2011) 987–1016.
[31] G.A. Wempner, Mechanics of Solids with Applications to Thin Bodies, Sijthoff & Noordhoff, Alphen aan den Rijn, 1981.
[32] J. Donea, S. Guiliani, J.P. Halleux, An arbitrary Lagrangian–Eulerian finite-element method for transient dynamic fluid structure interactions, Computer Methods in Applied Mechanics and Engineering 33 (1982) 689–723.
[33] C. Farhat, M. Lesoinne, N. Naman, Mixed explicit/implicit time integration of coupled aeroelastic problems: three-field formulation, geometric conservation and distributed solution, International Journal for Numerical Methods in Fluids 21 (1995) 807–835.
[34] E. Bechet, N. Moes, B. Wohlmuth, A stable Lagrange multiplier space for stiff interface conditions within the extended finite element method, International Journal for Numerical Methods in Engineering 78 (2009) 931–954.
[35] K. Bathe, F. Brezzi, Stability of finite element mixed interpolations for contact problems, Atti della Accademia Nazionale dei Lincei s. 9 12 (2001) 159–166.
[36] J. Pitkäranta, Boundary subspaces for the finite-element method with Lagrange multipliers, Numerische Mathematik 33 (1979) 273–289.
[37] C. Greif, D. Schötzau, Preconditioners for saddle point linear systems with highly singular (1,1) blocks, Electronic Transactions on Numerical Analysis 22 (2006) 114–121.
[38] C. Greif, D. Schötzau, Preconditioners for the discretized time-harmonic Maxwell equations in mixed form, Numerical Linear Algebra with Applications 14 (2007) 281–297.
[39] T. Rees, C. Greif, A preconditioner for linear systems arising from interior point optimization methods, SIAM Journal on Scientific Computing 29 (2007) 1992–2007.
[40] R. Smith, Some interlacing properties of the Schur complement of a Hermitian matrix, Linear Algebra and its Applications (1992) 137–144.
[41] J.W. Demmel, S.C. Eisenstat, J.R. Gilbert, X.S. Li, J.W.H.R. Liu, A supernodal approach to sparse partial pivoting, SIAM Journal on Matrix Analysis and Applications 20 (1999) 720–755.
[42] S. Mijalković, M. Mihajlović, Component-wise algebraic multigrid preconditioning for the iterative solution of stress analysis problems from microfabrication technology, Communications in Numererical Methods in Engineering 17 (2001) 737–747.
[43] V.E. Henson, U.M. Yang, BoomerAMG: a parallel algebraic multigrid solver and preconditioner, Applied Numerical Mathematics (2002) 155–177.
[44] A. Wathen, Realistic eigenvalue bounds for the Galerkin mass matrix, IMA Journal of Numerical Analysis 7 (1987) 449–457.
[45] R.L. Muddle, Parallel block preconditioning for multi-physics problems, Ph.D. thesis, University of Manchester, Manchester, 2010.
[46] H.C. Elman, D.J. Silvester, A.J. Wathen, Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics, Oxford University Press, 2005.
[47] O.C. Zienkiewicz, J.Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. Part 2: error estimates and adaptivity, International Journal for Numerical Methods in Engineering 33 (1992) 1365–1382.