A domain decomposition framework for fluid-structure interaction problems

Simone Deparis[1], Marco Discacciati[1], and Alfio Quarteroni[1,2]

[1] Institute of Analysis and Scientific Computing, CMCS-IACS, EPF Lausanne
simone.deparis@epfl.ch, marco.discacciati@epfl.ch
[2] MOX, Politecnico di Milano alfio.quarteroni@epfl.ch

Summary. In this note we review some classical algorithms for fluid-structure interaction problems and we propose an alternative viewpoint mutuated from the domain decomposition theory. This approach yields preconditioned Richardson iterations on the Steklov-Poincaré nonlinear equation at the fluid-structure interface.

Key words: Fluid-structure interaction, domain decomposition, finite element approximation, blood flow

1 Introduction

In this work, we address the numerical solution of fluid-structure interaction problems, in the case where implicit time advancing schemes are used. This yields at each time step a coupled system which is highly nonlinear, since the fluid domain depends on the unknown displacement of the structure. Standard strategies for solving this nonlinear problems are fixed point based methods such as Block-Gauss-Seidel (BGS) iterations (cf. [11, 15, 16]) or block Newton methods with exact or inexact jacobian (cf. [4, 7, 8, 9, 12]).

In this paper we revise these approaches, then we reformulate the fluid-structure interaction problem in a domain decomposition framework. On the associated interface problem we propose several splitting algorithms which are mutuated from subdomain iterative procedures. In particular we introduce several preconditioners that are obtained from the Steklov-Poincaré nonlinear operators associated with the fluid and the structure problems.

We consider a numerical example originated from the modeling of the interaction between blood-flow and vessel wall in a cylindric tract of an artery.

2 Fluid-structure interaction

In order to address each problem in its natural setting, we choose to consider the fluid in an ALE (Arbitrary Lagrangian Eulerian) formulation (cf. [4, 16]) and the structure in a pure Lagrangian framework.

The system under investigation occupies a moving domain Ω_t in its actual configuration. It is made of a deformable structure Ω_t^{s} (such as an arterial

wall, a pipe-line, ...) surrounding a fluid under motion (blood, oil, ...) in the complement $\Omega_t^{\rm f}$ of $\Omega_t^{\rm s}$ in Ω_t (see Fig. 1).

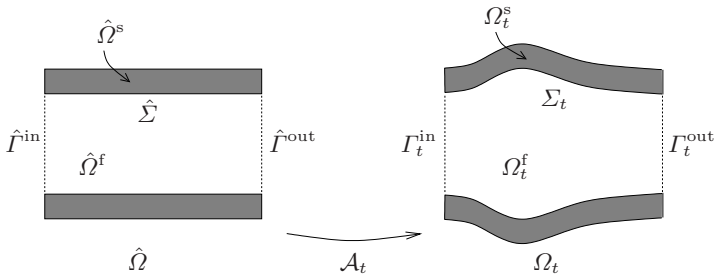

Fig. 1. ALE mapping between the initial configuration and the configuration at time t.

We assume the fluid to be Newtonian, viscous, homogeneous and incompressible. Its behavior is described by its velocity and pressure. The elastic solid under large displacements is described by its velocity and its stress tensor. The classical conservation laws of the continuum mechanics govern the evolution of these unknowns.

We denote by $\Gamma_t^{\rm in}$ and $\Gamma_t^{\rm out}$ the inflow and outflow sections of the fluid domain, by $\boldsymbol{n}_{\rm f}$ the fluid domain's outward normal on $\partial\Omega_t^{\rm f}$ and by $\boldsymbol{n}_{\rm s}$ the one of the structure on the reference boundary $\partial\hat{\Omega}^{\rm s}$. The boundary conditions on the fluid inlet and outlet can be either natural or essential (i.e., of Neumann or Dirichlet type, respectively), while on the interface we impose that the fluid and structure velocities match and so do the normal stresses. For simplicity, we assume zero body forces on both the structure and the fluid and that the boundary conditions on the remaining part of the structure boundary are of Dirichlet or of Neumann type.

The problem consists in finding the time evolution of the configuration $\Omega_t^{\rm f}$, as well as the velocity $\boldsymbol{u}$ and pressure p for the fluid and the displacement $\boldsymbol{d}$ of the structure. We define the ALE mapping

$$\forall t\,,\ \mathcal{A}_t : \hat{\Omega}^{\rm f} \to \Omega_t^{\rm f},$$

i.e. a map that retrieves at each time the current configuration of the computational domain $\Omega_t^{\rm f}$. Note in particular that on the reference interface $\hat{\Sigma}$, $\boldsymbol{n}_{\rm f}\circ\mathcal{A}_t = -\boldsymbol{n}_{\rm s}$. We denote by $\hat{\boldsymbol{x}}$ the coordinates on the reference configuration $\hat{\Omega}^{\rm f}$ and by $\boldsymbol{w} = \frac{d\mathcal{A}_t}{dt}$ the domain velocity.

For simplicity, we denote in short by Fluid(...) and Str(...) the fluid and structure problems, respectively. Precisely, for given vector functions $\boldsymbol{u}_{\rm in}$, $\boldsymbol{g}_{\rm f}$ and $\boldsymbol{f}_{\rm f}$, Fluid$(\boldsymbol{u}, p, \mathcal{A}_t; \boldsymbol{u}_{\rm in}, \boldsymbol{g}_{\rm f}, \boldsymbol{f}_{\rm f})$ means that we consider the following problem whose solution is $\boldsymbol{u}$, p, $\mathcal{A}_t$:

$$\text{Fluid}(\boldsymbol{u}, p, \mathcal{A}_t; \boldsymbol{u}_{\text{in}}, \boldsymbol{g}_{\text{f}}, \boldsymbol{f}_{\text{f}}) : \begin{cases} \boldsymbol{\Delta}\mathcal{A}_t = 0 \text{ in } \hat{\Omega}^{\text{f}}, \\ \mathcal{A}_t = 0 \text{ on } \partial\hat{\Omega}^{\text{f}} \setminus \hat{\Sigma}, \\ \Omega_t^{\text{f}} = \mathcal{A}_t(\hat{\Omega}^{\text{f}}), \\ \rho_{\text{f}} \left(\left.\dfrac{\partial \boldsymbol{u}}{\partial t}\right|_{\hat{\boldsymbol{x}}} + (\boldsymbol{u} - \boldsymbol{w}) \cdot \boldsymbol{\nabla}\boldsymbol{u} \right) \\ \qquad = \text{div}(2\mu\boldsymbol{\epsilon}(\boldsymbol{u})) - \boldsymbol{\nabla} p + \boldsymbol{f}_{\text{f}} \text{ in } \Omega_t^{\text{f}}, \\ \text{div}\, \boldsymbol{u} = 0 \text{ in } \Omega_t^{\text{f}}, \\ \boldsymbol{u} = \boldsymbol{u}_{\text{in}} \text{ on } \Gamma_t^{\text{in}}, \\ \boldsymbol{\sigma}_{\text{f}}(\boldsymbol{u}, p) \cdot \boldsymbol{n}_{\text{f}} = \boldsymbol{g}_{\text{f}} \text{ on } \Gamma_t^{\text{out}}, \end{cases} \tag{1}$$

where ρ_{f} is the fluid density, μ its viscosity, $\boldsymbol{\epsilon}(\boldsymbol{u}) = \frac{\left(\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^T\right)}{2}$ is the strain rate tensor and $\boldsymbol{\sigma}_{\text{f}}(\boldsymbol{u}, p) = -pId + 2\mu\boldsymbol{\epsilon}(\boldsymbol{u})$ the Cauchy stress tensor (Id is the identity matrix). Note that (1) does not univocally define a solution $(\boldsymbol{u}, p, \mathcal{A}_t)$ as no boundary data are prescribed on the interface Σ_t.

Similarly, for given vector functions $\boldsymbol{g}_{\text{s}}$, $\boldsymbol{f}_{\text{s}}$, $\text{Str}(\boldsymbol{d}; \boldsymbol{g}_{\text{s}}, \boldsymbol{f}_{\text{s}})$ means that we consider the following problem whose solution is $\boldsymbol{d}$:

$$\text{Str}(\boldsymbol{d}; \boldsymbol{g}_{\text{s}}, \boldsymbol{f}_{\text{s}}) : \begin{cases} \rho_{\text{s}} \dfrac{\partial^2 \boldsymbol{d}}{\partial t^2} = \text{div}(\boldsymbol{\sigma}_{\text{s}}(\boldsymbol{d})) - \gamma \boldsymbol{d} + \boldsymbol{f}_{\text{s}} \text{ in } \hat{\Omega}^{\text{s}}, \\ \boldsymbol{\sigma}_{\text{s}}(\boldsymbol{d}) \cdot \boldsymbol{n}_{\text{s}} = \boldsymbol{g}_{\text{s}} \text{ on } \partial\hat{\Omega}^{\text{s}} \setminus \hat{\Sigma}, \end{cases} \tag{2}$$

where $\boldsymbol{\sigma}_{\text{s}}(\boldsymbol{d})$ is the first Piola–Kirchoff stress tensor, γ is a coefficient accounting for possible viscoelastic effects, while $\boldsymbol{g}_{\text{s}}$ represents the normal traction on external boundaries. Appropriate models have to be chosen for the structure depending on the specific problem at hand. The reader may refer, e.g., to [1, 3, 10, 18].

Similarly to what we have noticed for (1), problem (2) can not define univocally the unknown $\boldsymbol{d}$ because a boundary value on $\hat{\Sigma}$ is missing.

When coupling the two problems together, the "missing" boundary conditions are indeed supplemented by suitable matching conditions on the reference interface $\hat{\Sigma}$. More precisely, if we denote by λ the interface variable corresponding to the displacement $\boldsymbol{d}$ on $\hat{\Sigma}$, at any time the coupling conditions on the reference interface $\hat{\Sigma}$ are

$$\begin{aligned} \mathcal{A}_t &= \lambda, \\ \boldsymbol{u} \circ \mathcal{A}_t &= \dot{\boldsymbol{d}}_{\hat{\Sigma}}, \\ (\boldsymbol{\sigma}_{\text{f}}(\boldsymbol{u}, p) \cdot \boldsymbol{n}_{\text{f}}) \circ \mathcal{A}_t + \boldsymbol{\sigma}_{\text{s}}(\boldsymbol{d}) \cdot \boldsymbol{n}_{\text{s}} &= 0, \end{aligned} \tag{3}$$

where $\dot{\boldsymbol{d}}_{\hat{\Sigma}}$ denotes the temporal derivative of $\boldsymbol{d}_{|\hat{\Sigma}}$.

The system of equations (1)-(3) identifies our coupled fluid-structure problem.

2.1 Fluid and structure operators

We suppose the problem to be discretized in time. When the solution is available at time t^n, we look for the solution at the new time level $t^{n+1} = t^n + \delta t$. When no ambiguity occurs, all the quantities will be referred to at time $t = t^{n+1}$.

In view of domain decomposition formulation we introduce two interface operators S_f and S_s as follows.

S_f is the Dirichlet–to–Neumann map in Ω_t^f,

$$S_\mathrm{f} : \lambda \mapsto \text{find } (\boldsymbol{u}, p, \mathcal{A}_t) : \begin{cases} \mathrm{Fluid}(\boldsymbol{u}, p, \mathcal{A}_t; \boldsymbol{u}_\mathrm{in}, \boldsymbol{g}_\mathrm{f}, \boldsymbol{f}_\mathrm{f}) \\ \boldsymbol{u}_{|\Sigma_t} \circ \mathcal{A}_t = (\lambda - \boldsymbol{d}^n_{|\hat{\Sigma}})/\delta t \end{cases} \\ \mapsto \sigma_\mathrm{f} = (\boldsymbol{\sigma}_\mathrm{f}(\boldsymbol{u}, p) \cdot \boldsymbol{n}_\mathrm{f})_{|\Sigma_t} \circ \mathcal{A}_t,$$

while S_s is the Dirichlet–to–Neumann map in $\hat{\Omega}^\mathrm{s}$,

$$S_\mathrm{s} : \lambda \ \mapsto \ \text{find } \boldsymbol{d} : \begin{cases} \mathrm{Str}(\boldsymbol{d}; \boldsymbol{g}_\mathrm{s}, \boldsymbol{f}_\mathrm{s}) \\ \boldsymbol{d}_{|\hat{\Sigma}} = \lambda \end{cases} \quad \mapsto \ \sigma_\mathrm{s} = \boldsymbol{\sigma}_\mathrm{s}(\boldsymbol{d}) \cdot \boldsymbol{n}_\mathrm{s}.$$

Let us introduce the corresponding homogeneous operators that will serve as building blocks in the construction of our preconditioners. For any given λ, we define the homogeneous Dirichlet–to–Neumann maps in Ω_t^f and $\hat{\Omega}^\mathrm{s}$, respectively, by

$$\begin{aligned} \bar{S}_{\mathrm{f},\lambda}(\delta\lambda) &= S_\mathrm{f}(\lambda + \delta\lambda) - S_\mathrm{f}(\lambda) \\ \bar{S}_{\mathrm{s},\lambda}(\delta\lambda) &= S_\mathrm{s}(\lambda + \delta\lambda) - S_\mathrm{s}(\lambda). \end{aligned} \tag{4}$$

Next we formally introduce the tangent operators for the fluid and structure problems. First of all, let Fluid'_λ (respectively, Str'_λ) denote the linearized fluid (respectively, structure) problem around λ (the reader may refer to [7, 8] for a precise definition), where we consider homogeneous boundary conditions on $\partial\Omega_t^\mathrm{f} \setminus \Sigma_t$ and on $\partial\hat{\Omega}^\mathrm{s} \setminus \hat{\Sigma}$ for the fluid and the structure, respectively, and null body forces for both problems.

The definitions of the tangent operators are as follows.

For the fluid:

(*i*) Dirichlet–to–Neumann tangent map in Ω_t^f about a given point λ. For any increment $\delta\lambda$, $\delta\sigma_\mathrm{f} = S'_\mathrm{f}(\lambda) \cdot \delta\lambda$ is computed as follows:

$$S'_\mathrm{f}(\lambda) : \delta\lambda \mapsto \text{find } (\delta\boldsymbol{u}, \delta p, \delta\mathcal{A}_t) : \begin{cases} \mathrm{Fluid}'_\lambda(\delta\boldsymbol{u}, \delta p, \delta\mathcal{A}_t) \\ \delta\boldsymbol{u}_{|\Sigma_t} \circ \delta\mathcal{A}_t = \frac{\delta\lambda}{\delta t} \end{cases} \\ \mapsto \delta\sigma_\mathrm{f} = (\boldsymbol{\sigma}_\mathrm{f}(\delta\boldsymbol{u}, \delta p) \cdot \boldsymbol{n}_\mathrm{f})_{|\Sigma_t} \circ \delta\mathcal{A}_t.$$

(*ii*) Neumann–to–Dirichlet tangent map about a given point λ. For any increment $\delta\sigma_\mathrm{f}$, $\delta\lambda = (S'_\mathrm{f}(\lambda))^{-1}\, \delta\sigma_\mathrm{f}$ is computed as follows:

$$(S'_{\rm f}(\lambda))^{-1} : \delta\sigma_{\rm f} \mapsto \text{find } (\delta\boldsymbol{u}, \delta p, \delta\mathcal{A}_t) : \begin{cases} \text{Fluid}'_\lambda(\delta\boldsymbol{u}, \delta p, \delta\mathcal{A}_t) \\ (\boldsymbol{\sigma}_{\rm f}(\delta\boldsymbol{u}, \delta p) \cdot \boldsymbol{n}_{\rm f})_{|\Sigma_t} \circ \delta\mathcal{A}_t = \delta\sigma_{\rm f} \end{cases}$$
$$\mapsto \ \delta\lambda = \delta t\, \delta\boldsymbol{u}_{|\Sigma_t} \circ \delta\mathcal{A}_t.$$

For the structure:

(*iii*) Dirichlet–to–Neumann tangent map about a given point λ. For any increment $\delta\lambda$, $\delta\sigma_{\rm s} = S'_{\rm s}(\lambda) \cdot \delta\lambda$ is computed as follows:

$$S'_{\rm s}(\lambda) : \delta\lambda \mapsto \text{find } \delta\boldsymbol{d} : \begin{cases} \text{Str}'_\lambda(\delta\boldsymbol{d}) \\ \delta\boldsymbol{d}_{|\hat{\Sigma}} = \delta\lambda \end{cases} \quad \mapsto \ \delta\sigma_{\rm s} = \boldsymbol{\sigma}_{\rm s}(\delta\boldsymbol{d}) \cdot \boldsymbol{n}_{\rm s} \text{ on } \hat{\Sigma}.$$

(*iv*) Neumann–to–Dirichlet tangent map about a given point λ. For any increment $\delta\sigma_{\rm s}$, $\delta\lambda = (S'_{\rm s}(\lambda))^{-1}\, \delta\sigma_{\rm s}$ is computed as follows:

$$(S'_{\rm s}(\lambda))^{-1} : \delta\sigma_{\rm s} \mapsto \text{ find } \delta\boldsymbol{d} : \begin{cases} \text{Str}'_\lambda(\delta\boldsymbol{d}) \\ \boldsymbol{\sigma}_{\rm s}(\delta\boldsymbol{d}) \cdot \boldsymbol{n}_{\rm s} = \delta\sigma_{\rm s} \end{cases} \quad \mapsto \ \delta\lambda = \delta\boldsymbol{d}_{|\hat{\Sigma}} \ .$$

Remark 1 (special cases). (a) In some cases it might occur that the model adopted for the structure has a lower dimension than that of the fluid (see, e.g., [18], where a two-dimensional model is used for the fluid whereas a one-dimensional model is used for the structure). In such cases the structure domain reduces to the interface Σ_t. The methods presented here still apply; however, in that case the application of the operator $S_{\rm s}$ implies simply a computation of the residue.

(b) When the fluid or the structure are linear like in the test case addressed in Sect. 5, the tangent maps are equivalent to the homogeneous operators, precisely

$$\bar{S}_{{\rm f},\lambda}(\delta\lambda) = S'_{\rm f}(\lambda) \cdot \delta\lambda \qquad \text{or} \qquad \bar{S}_{{\rm s},\lambda}(\delta\lambda) = S'_{\rm s}(\lambda) \cdot \delta\lambda.$$

3 The problem at the interface

With the help of the maps $S_{\rm f}$ and $S_{\rm s}$ previously introduced, there are at least three ways to express the coupled problem in terms of the interface variable λ. From the mathematical side they are all equivalent. However, they can inspire different iterative methods.

3.1 Fixed point formulation

The most obvious way to reformulate the coupled problem (1)-(3) is through the following fixed point formulation: find an interface displacement λ such that

$$S_{\rm s}^{-1}(-S_{\rm f}(\lambda)) = \lambda. \tag{5}$$

The explanation is as follows: the displacement λ generates the stress $S_{\rm f}(\lambda)$ on the interface. The stress applied to the structure is then $-S_{\rm f}(\lambda)$ and its corresponding displacement has to reproduce λ.

When treating fluid-structure interaction problems, this is the usual way to consider the coupling. The existing algorithms are conceived to solve equation (5). See for example [8, 11, 16].

Fixed point iterations

A standard algorithm to solve problem (5) is based on relaxed fixed point iterations. One iteration of the fixed point algorithm reads: for a given λ^k, do

$$\begin{array}{l} \sigma_{\rm f}^k = S_{\rm f}(\lambda^k), \\ \bar{\lambda}^k = S_{\rm s}^{-1}(-\sigma_{\rm f}^k), \\ \lambda^{k+1} = \lambda^k + \omega^k(\bar{\lambda}^k - \lambda^k), \end{array} \tag{6}$$

where all equalities are valid on $\hat{\Sigma}$. The choice of the relaxation parameter ω^k is crucial for the convergence of the method (see [2] for a recent analysis). With this purpose, in [14] the authors apply a generalization to the vector case of the Aitken acceleration technique. This allows a dynamic (and automatic) choice of the relaxation parameter, according to the formula

$$\omega^k = -\frac{\left(\bar{\lambda}^k - \lambda^k - \bar{\lambda}^{k-1} + \lambda^{k-1}\right) \cdot \left(\lambda^k - \lambda^{k-1}\right)}{\left\|\bar{\lambda}^k - \lambda^k - \bar{\lambda}^{k-1} + \lambda^{k-1}\right\|^2}.$$

In fact, as pointed out in [4], this choice of ω^k is the one that minimizes the norm of

$$\left\|\left(\lambda^k - \lambda^{k-1}\right) + \omega\left(\bar{\lambda}^k - \lambda^k - \bar{\lambda}^{k-1} + \lambda^{k-1}\right)\right\|.$$

The main advantage of this method is that the fluid and structure problems are solved independently. In fact, each step k of the algorithm (6) implies:

1. apply $S_{\rm f}$ to a given displacement λ^k, that is solve the fluid problem in $\Omega_t^{\rm f}$ with boundary condition $\boldsymbol{u}_{|\Sigma_t} \circ \mathcal{A}_t = (\lambda^k - \boldsymbol{d}^n_{|\hat{\Sigma}})/\delta t$ on $\hat{\Sigma}$; then compute the stress $\sigma_{\rm f}^k = (\boldsymbol{\sigma}_{\rm f}(\boldsymbol{u}^k, p^k) \cdot \boldsymbol{n}_{\rm f})_{|\Sigma_t} \circ \mathcal{A}_t$ on the interface $\hat{\Sigma}$;
2. apply the inverse of $S_{\rm s}^{-1}$ to $-\sigma_{\rm f}^k$, that is solve the structure problem in $\hat{\Omega}^{\rm s}$ with boundary condition $\sigma_{\rm s}(\boldsymbol{d}^k) \cdot \boldsymbol{n}_{\rm s} = -\sigma_{\rm f}^k$ on $\hat{\Sigma}$; then compute the correction $\overline{\lambda}^k$ of the displacement at the iterate k.

The main drawback of this method is that it may result in a slow convergence rate.

3.2 Rootfinding formulation

A formulation of the interface problem slightly different from (5) (cf. [7, 8, 9, 12]) is: find an interface displacement λ such that

$$\Phi(\lambda) = 0, \quad \text{where } \Phi(\lambda) = S_{\mathrm{s}}^{-1}(-S_{\mathrm{f}}(\lambda)) - \lambda. \tag{7}$$

Newton algorithm

Let $J(\lambda)$ denote the Jacobian of $S_{\mathrm{s}}^{-1}(-S_{\mathrm{f}}(\lambda))$ in λ. Given λ^0, for $k \geq 0$, a step of the Newton algorithm associated to problem (7) reads:

$$\begin{array}{l} \sigma_{\mathrm{f}}^k = S_{\mathrm{f}}(\lambda^k), \\ \bar{\lambda}^k = S_{\mathrm{s}}^{-1}(-\sigma_{\mathrm{f}}^k), \\ (J(\lambda^k) - Id)\mu^k = -(\bar{\lambda}^k - \lambda^k), \\ \lambda^{k+1} = \lambda^k + \omega^k \mu^k. \end{array} \tag{8}$$

When $J(\lambda)$ represents the exact Jacobian, ω^k can be taken equal to one. Otherwise, it can be computed by a line search technique (see, e.g., [17]).

Note that the Jacobian of $S_{\mathrm{s}}^{-1}(-S_{\mathrm{f}}(\lambda^k))$ in λ^k has the following expression

$$J(\lambda^k) = -\left[S_{\mathrm{s}}'\left(S_{\mathrm{s}}^{-1}(-S_{\mathrm{f}}(\lambda^k))\right)\right]^{-1} \cdot S_{\mathrm{f}}'(\lambda^k) = -\left[S_{\mathrm{s}}'\left(\bar{\lambda}^k\right)\right]^{-1} \cdot S_{\mathrm{f}}'(\lambda^k). \tag{9}$$

We point out that algorithm (8) requires an additional step with respect to (6): in fact we have to solve the linear system with matrix $J(\lambda^k) - Id$. Indeed, while the computation of $\left[S_{\mathrm{s}}'\left(\bar{\lambda}^k\right)\right]^{-1} \cdot \delta\sigma$ (for any given $\delta\sigma$) does only require the derivative with respect to the state variable at the interface, the computation of $S_{\mathrm{f}}'(\lambda^k) \cdot \delta\lambda$ implies also shape derivatives, since a variation in λ determines a variation of the fluid domain. This is a nontrivial task. In the literature, several approaches have been proposed to solve exactly the tangent problem [7], or else to approximate it by either simpler models for the fluid [5, 8], or through finite differences schemes [9, 13, 20]. However, the lack of *a priori* criteria for selecting optimal finite difference infinitesimal steps may lead to a reduction of the overall convergence speed [8].

3.3 Domain decomposition (or Steklov-Poincaré) formulation

The computational domain is naturally split into the fluid domain Ω_t^{f} and the structure Ω_t^{s}. It will be clear from the context, whether we impose Dirichlet or Neumann boundary conditions on the interface Σ_t, while on $\partial\Omega_t^{\mathrm{f}} \setminus \Sigma_t$ and $\partial\Omega_t^{\mathrm{s}} \setminus \Sigma_t$ the problems have always the same kind of boundary conditions. We refer to [19] for a general setting of domain decomposition methods.

The Steklov–Poincaré interface equation associated to the differential problem is: find λ such that

$$S_{\mathrm{f}}(\lambda) + S_{\mathrm{s}}(\lambda) = 0. \tag{10}$$

Note that the dependence on the data is hidden in the definition of the operators S_{f} and S_{s}. This is necessary since the problem at hand is nonlinear.

Remark that equation (10) can formally be retrieved by applying S_{s} to both sides of (5).

Preconditioned (nonlinear) Richardson method

Since the Steklov–Poincaré problem (10) is nonlinear, the Richardson method must be interpreted in a slightly different way than what is done in the literature for the linear case (see, e.g., [19]). Given λ^0, for $k \geq 0$, the iterative method reads:

$$\begin{aligned} &\sigma_{\mathrm{f}}^k = S_{\mathrm{f}}(\lambda^k), \\ &\sigma_{\mathrm{s}}^k = S_{\mathrm{s}}(\lambda^k), \\ &\sigma^k = -\left(\sigma_{\mathrm{f}}^k + \sigma_{\mathrm{s}}^k\right), \\ &\mu^k = P^{-1}\sigma^k, \\ &\lambda^{k+1} = \lambda^k + \omega^k \mu^k, \end{aligned} \tag{11}$$

with appropriate choice of the scalar ω^k. Every equation should still be intended on $\hat{\Sigma}$. The preconditioner P, that must be chosen appropriately, maps the interface variable onto the space of normal stresses, say $P : \Lambda \to Y$. Note that if $(S_{\mathrm{f}} + S_{\mathrm{s}})$ is affine, if P is taken equal to $\bar{S}_{\mathrm{f}} + \bar{S}_{\mathrm{s}}$ and $\omega^k = 1$, then the Richardson method converges in one iteration. It is also possible to choose a preconditioner which depends on the iterate λ^k or more generally on the iteration step k. In these cases we will denote it by P_k.

A general strategy to compute the relaxation parameter ω^k is given by :

$$\omega^k = -\frac{\left(\mu^k - \mu^{k-1}\right) \cdot \left(\lambda^k - \lambda^{k-1}\right)}{\left\|\mu^k - \mu^{k-1}\right\|^2}. \tag{12}$$

This value of ω^k is the one that minimizes the norm

$$\left\|\left(\lambda^k - \lambda^{k-1}\right) + \omega\left(\mu^k - \mu^{k-1}\right)\right\|,$$

over all possible values of ω. This criterium generalizes to the vector case the Aitken extrapolation technique (see [4]).

At each step k, algorithm (11) requires to solve independently the fluid and the structure problems and to apply a preconditioner. Precisely,

1. apply S_{f} to λ^k, that is solve the fluid problem as already illustrated for algorithm (6);
2. apply S_{s} to λ^k, that is solve the structure problem with boundary condition $\boldsymbol{d}^k_{|\hat{\Sigma}} = \lambda^k$ on $\hat{\Sigma}$ and compute the stress $\sigma_{\mathrm{s}}^k = \sigma_{\mathrm{s}}(\boldsymbol{d}^k) \cdot \boldsymbol{n}_{\mathrm{s}}$ on $\hat{\Sigma}$;
3. apply the preconditioner P^{-1} to the total stress σ^k on $\hat{\Sigma}$.

Note that steps 1. and 2. can be performed in parallel.

Remark 2. If no preconditioner is used, then at the differential level P should be intended as being the projection operator $\mathcal{I}$ from the space of displacements Λ to the space of stresses Y, so that

$$\lambda^{k+1} = \lambda^k + \omega^k \mathcal{I}^{-1} \sigma^k \in \Lambda. \tag{13}$$

At the algebraic level, this remark can be omitted since in that case we are always dealing with vectors of $\mathbb{R}^n$.

The crucial issue is how we can set up a preconditioner (more precisely, a scaling operator) in order for the iterative method to converge as quickly as possible. We address this problem in the next subsections.

4 Preconditioners for the domain decomposition formulation

In this section we discuss some classical choices of the preconditioner for the Richardson method applied to the domain decomposition approach. We also compare the proposed preconditioners to the fixed point and Newton strategies that we have illustrated in Sects. 3.1 and 3.2.

4.1 Dirichlet–Neumann and Neumann–Neumann preconditioners

We define a generic linear preconditioner (more precisely, its inverse):

$$P_k^{-1} = \alpha_{\mathrm{f}}^k \, S_{\mathrm{f}}'(\lambda^k)^{-1} + \alpha_{\mathrm{s}}^k \, S_{\mathrm{s}}'(\lambda^k)^{-1}, \tag{14}$$

for two scalars α_{f}^k and α_{s}^k. Another possibility is to use the operators $\bar{S}_{\mathrm{f},\lambda^k}$ and $\bar{S}_{\mathrm{s},\lambda^k}$ instead of $S_{\mathrm{f}}'(\lambda^k)$ and $S_{\mathrm{s}}'(\lambda^k)$, respectively:

$$P_k^{-1} = \alpha_{\mathrm{f}}^k \, \bar{S}_{\mathrm{f},\lambda^k}^{-1} + \alpha_{\mathrm{s}}^k \, \bar{S}_{\mathrm{s},\lambda^k}^{-1}. \tag{15}$$

In the special case in which both S_{f} and S_{s} are linear, since the tangent problems are equivalent to the homogeneous problems, (14) and (15) coincide. Otherwise, (15) is a nonlinear operator and can be considered as an approximation of (14) to be used in order to avoid the solution of the linearized problems.

From (15) we retrieve the following special cases:

– If $\alpha_{\mathrm{f}}^k = 1$ and $\alpha_{\mathrm{s}}^k = 0$, then

$$P_k^{-1} = P_{DN}^{-1} = \bar{S}_{\mathrm{f},\lambda^k}^{-1}.$$

P_{DN} is called a *Dirichlet-Neumann preconditioner* and

$$P_{DN}^{-1}(\sigma^k) = \bar{S}_{\mathrm{f},\lambda^k}^{-1} \left(-S_{\mathrm{f}}(\lambda^k) - S_{\mathrm{s}}(\lambda^k) \right);$$

– If $\alpha_{\mathrm{f}}^k = 0$ and $\alpha_{\mathrm{s}}^k = 1$, then

$$P_k^{-1} = P_{ND}^{-1} = \bar{S}_{\mathrm{s},\lambda^k}^{-1}.$$

This is called a *Neumann-Dirichlet preconditioner* and

$$P_{ND}^{-1}(\sigma^k) = \bar{S}_{\mathrm{s},\lambda^k}^{-1}\left(-S_{\mathrm{f}}(\lambda^k) - S_{\mathrm{s}}(\lambda^k)\right);$$

– If $\alpha_{\mathrm{f}}^k + \alpha_{\mathrm{s}}^k = 1$, then

$$P_k^{-1} = P_{NN}^{-1} = \alpha_{\mathrm{f}}^k \bar{S}_{\mathrm{f},\lambda^k}^{-1} + \alpha_{\mathrm{s}}^k \bar{S}_{\mathrm{s},\lambda^k}^{-1}$$

which is called a *Neumann-Neumann preconditioner*.

In the Dirichlet–Neumann (or the Neumann–Dirichlet) case the computational effort of a Richardson step may be reduced to the solution of only one Dirichlet problem in one subdomain and one Neumann problem in the other.

For both cases (14), (15), it is possible to choose the parameters α_{f}^k, α_{s}^k and ω^k dynamically in the following way. We define $\omega_{\mathrm{f}}^k = \omega^k \alpha_{\mathrm{f}}^k$ and $\omega_{\mathrm{s}}^k = \omega^k \alpha_{\mathrm{s}}^k$ and we look for ω_{f}^k and ω_{s}^k that minimize

$$\left\| \left(\lambda^k - \lambda^{k-1}\right) + \omega_{\mathrm{f}}\left(\mu_{\mathrm{f}}^k - \mu_{\mathrm{f}}^{k-1}\right) + \omega_{\mathrm{s}}\left(\mu_{\mathrm{s}}^k - \mu_{\mathrm{s}}^{k-1}\right) \right\|,$$

over all possible values of ω_{f} and ω_{s}, which corresponds to solving the linear system

$$A^T A \begin{pmatrix} \omega_{\mathrm{f}}^k \\ \omega_{\mathrm{s}}^k \end{pmatrix} = -A^T \left(\lambda^k - \lambda^{k-1}\right), \tag{16}$$

where A is the two column matrix

$$A = \left(\left(\mu_{\mathrm{f}}^k - \mu_{\mathrm{f}}^{k-1}\right); \left(\mu_{\mathrm{s}}^k - \mu_{\mathrm{s}}^{k-1}\right)\right).$$

Again, this can be regarded as a generalized Aitken criterium (see [4]). Finally, we set

$$\lambda^{k+1} = \lambda^k + \omega_{\mathrm{f}}^k \mu_{\mathrm{f}}^k + \omega_{\mathrm{s}}^k \mu_{\mathrm{s}}^k.$$

Note that this automatic choice generalizes the one carried out in (12).

Remark 3. In the linear case we obtain the simple expressions

$$P_{DN}^{-1}(\sigma^k) = S_{\mathrm{f}}^{-1}(-S_{\mathrm{s}}(\lambda^k)) - \lambda^k;$$

$$P_{ND}^{-1}(\sigma^k) = S_{\mathrm{s}}^{-1}(-S_{\mathrm{f}}(\lambda^k)) - \lambda^k.$$

The convergence of the preconditioned Richardson algorithm (11) with P_{DN}^{-1} or P_{ND}^{-1} is proved, thanks to the equivalence with fixed point iterations, in [2]. The proof can be extended to P_{NN}^{-1} using the classical theory of domain decomposition (cf. [19, §4]). A complete proof for the nonlinear case will be given in a future work.

Comparison with fixed point iterations

The previously introduced fixed point iterations (6) can be regarded as special instances of the general preconditioned domain decomposition algorithm (11) for a suitable choice of the preconditioner.

In fact, let us consider the special case when the structure model is linear (but not necessarily the one of the fluid). Then, if we choose $\alpha_{\mathrm{f}}^k = 0$ and $\alpha_{\mathrm{s}}^k = 1$, the algorithm (11) is equivalent to the fixed-point algorithm (6) (see section 3.1). Indeed, from (11)

$$\mu^k = (S_{\mathrm{s}}'(\lambda^k))^{-1} \left(-S_{\mathrm{f}}(\lambda^k) - S_{\mathrm{s}}(\lambda^k)\right) = S_{\mathrm{s}}^{-1} \left(-S_{\mathrm{f}}(\lambda^k)\right) - \lambda^k,$$

hence $\lambda^{k+1} = \lambda^k + \omega^k(\bar{\lambda}^k - \lambda^k)$, which coincides with the last equality of (6).

4.2 The Robin-Robin preconditioner

A further possibility is offered by the following preconditioner

$$P_{RR} = \frac{1}{\gamma_{\mathrm{f}} + \gamma_{\mathrm{s}}} \left(\gamma_{\mathrm{f}}\mathcal{I} + S_{\mathrm{f}}'(\lambda^k)\right) \mathcal{I} \left(\gamma_{\mathrm{s}}\mathcal{I} + S_{\mathrm{s}}'(\lambda^k)\right),$$

where $\mathcal{I}$ is the projection operator from Λ to Y (see Remark 2), while γ_{f} and γ_{s} are positive parameters which can be chosen using a suitable error minimization strategy (see [6]). We call P_{RR} a *Robin-Robin preconditioner*.

4.3 The Newton algorithm on the Steklov-Poincaré equation

The genuine Newton algorithm applied to the Steklov-Poincaré problem (10) is retrieved by using the algorithm (11) (with $\omega^k = 1$) and choosing P at the step k as

$$P_k = S_{\mathrm{f}}'(\lambda^k) + S_{\mathrm{s}}'(\lambda^k). \tag{17}$$

Note that to invert P_k one must use a (preconditioned) iterative method and may approximate the tangent problems to accelerate the computations.

Comparison with the Newton algorithm (8) on problem (7)

The Richardson algorithm (11) for the Steklov-Poincaré formulation (10) with preconditioner given by (14) (with $\alpha_{\mathrm{f}}^k = \alpha_{\mathrm{s}}^k = 1$) is not equivalent to the Newton algorithm (8) applied to problem (7). In fact, the Newton algorithm (8) could be regarded as a Richardson method (11), choosing however a nonlinear preconditioner defined as

$$P_k(\mu) = S_{\mathrm{s}} \left(S_{\mathrm{s}}'(\bar{\lambda}^k)^{-1} \cdot \left(S_{\mathrm{f}}'(\lambda^k) + S_{\mathrm{s}}'(\bar{\lambda}^k)\right) \cdot \mu \right), \tag{18}$$

where $\bar{\lambda}^k = S_{\mathrm{s}}^{-1}(-S_{\mathrm{f}}(\lambda^k))$.

In this case, for $\sigma^k = -(S_\mathrm{f}(\lambda^k) + S_\mathrm{s}(\lambda^k))$, we would obtain

$$\begin{aligned} P_k^{-1}\left(\sigma^k\right) &= \left(\ S_\mathrm{f}'(\lambda^k) + S_\mathrm{s}'(\bar{\lambda}^k)\ \right)^{-1} \cdot S_\mathrm{s}'(\bar{\lambda}^k)\ \cdot\ S_\mathrm{s}^{-1}\left(-S_\mathrm{f}(\lambda^k) - S_\mathrm{s}(\lambda^k)\right) \\ &= \left(\left[S_\mathrm{s}'(\bar{\lambda}^k)\right]^{-1} \cdot S_\mathrm{f}'(\lambda^k) + Id\right)^{-1}\ \left(S_\mathrm{s}^{-1}(-S_\mathrm{f}(\lambda^k)) - \lambda^k\right). \end{aligned}$$

We see that this is equivalent to (8). In fact (9) is equal to the first bracket in the last line.

Remark 4. Note that if (only) the structure is linear, the preconditioner defined in (18) is also linear and becomes

$$P_k = S_\mathrm{f}'(\lambda^k) + S_\mathrm{s}'(\lambda^k),$$

which is exactly (17). This is a Newton method applied to (7) or (10). However, we would like to remark that the domain decomposition approach allows us to set up a completely parallel solver. In fact, the fluid and the structure subproblems can be computed simultaneously (and independently) for both the residue computation (operators S_f and S_s) and the application of the preconditioner (operators S_f' and/or S_s').

Fixed point	Rootfinding	Steklov-Poincaré
$S_\mathrm{s}^{-1}(-S_\mathrm{f}(\lambda)) = \lambda$	$S_\mathrm{s}^{-1}(-S_\mathrm{f}(\lambda)) - \lambda = 0$	$S_\mathrm{f}(\lambda) + S_\mathrm{s}(\lambda) = 0$
Fixed point iter.	Newton iter.	Prec. Richardson iter.
$\sigma_\mathrm{f}^k = S_\mathrm{f}(\lambda^k)$ $\overline{\lambda}^k = S_\mathrm{s}^{-1}(-\sigma_\mathrm{f}^k)$ $\lambda^{k+1} = \lambda^k + \omega^k(\overline{\lambda}^k - \lambda^k)$	$\sigma_\mathrm{f}^k = S_\mathrm{f}(\lambda^k)$ $\overline{\lambda}^k = S_\mathrm{s}^{-1}(-\sigma_\mathrm{f}^k)$ $(J(\lambda^k) - Id)\mu^k = -(\overline{\lambda}^k - \lambda^k)$ $\lambda^{k+1} = \lambda^k + \omega^k\mu^k$	$\sigma^k = S_\mathrm{f}(\lambda^k) + S_\mathrm{s}(\lambda^k)$ $\mu^k = P^{-1}(-\sigma^k)$ $\lambda^{k+1} = \lambda^k + \omega^k\mu^k$
1 flow solve 1 structure solve	1 flow solve 1 structure solve 1 Jacobian solve	1 flow solve 1 structure solve 1 precond. solve

Table 1. Comparison among fixed point, rootfinding and Steklov-Poincaré approaches.

5 A linear test case

We have considered a simple model for the fluid and the structure that has been proposed by Causin, Gerbeau and Nobile in [2]. Due to the linearity of

the problem, we could not test the effectiveness of the tangential operators as preconditioners; we leave this test to a future work where we shall consider a general nonlinear context. However, even in this simple case the domain decomposition approach that we advocate can provide better convergence results than those obtained using the classical fixed point approach.

The domain under consideration is a rectangle Ω $6\,cm$ long and $1\,cm$ wide as represented in Fig. 1. The one-dimensional wall $\hat{\Sigma}$ is on the top of the domain and coincides with the interface. We consider a pressure wave entering the computational domain from the left and a symmetry condition on the x-axis. We consider only small displacements and neglect the domain deformation in the fluid domain. For this reason we will drop the dependence on t of both Ω_t^{f} and Σ_t.

The inviscid incompressible fluid is modeled as follows:

$$\mathrm{Fluid}(\boldsymbol{u}, p; p_{\mathrm{in}}, 0) : \begin{cases} \rho_{\mathrm{f}} \dfrac{\partial \boldsymbol{u}}{\partial t} + \nabla p = 0 \text{ in } \Omega^{\mathrm{f}}, \ \forall t, \\ \mathrm{div}\boldsymbol{u} = 0 \text{ in } \Omega^{\mathrm{f}}, \\ p = p_{\mathrm{in}}(t) \text{ on } \Gamma^{\mathrm{in}}, \\ p = 0 \text{ on } \Gamma^{\mathrm{out}}, \\ \frac{\partial p}{\partial n} = 0 \text{ on the } x\text{-axis}, \end{cases} \tag{19}$$

where $\rho_{\mathrm{f}} = 1 g/cm^2$ is the blood density.

The structure is described by a string model on $\hat{\Sigma}$ with homogeneous Dirichlet values as boundary conditions at the end points of $\hat{\Sigma}$:

$$\mathrm{Str}(d; 0) : \left\{ \rho_{\mathrm{s}} h_{\mathrm{s}} \frac{\partial^2 d}{\partial t^2} - kG h_{\mathrm{s}} \frac{\partial^2 d}{\partial x^2} + \frac{E h_{\mathrm{s}}}{1 - \nu^2} \frac{d}{R_0^2} = f_{\mathrm{s}}, \right. \tag{20}$$

where we set the wall density $\rho_{\mathrm{s}} h_{\mathrm{s}} = 1.1 g/cm^2$, the Young modulus $E = 7.5 \cdot 10^5 dynes/cm^2$, the Poisson coefficient $\nu = 0.5$ and the shear modulus $kG = 2.5 \cdot 10^5 dynes/cm^2$.

The coupling conditions on the interface $\hat{\Sigma}$ impose the continuity of the normal velocity:

$$\boldsymbol{u} \cdot \boldsymbol{n}_{\mathrm{f}} (= u_2) = \frac{\partial d}{\partial t} \tag{21}$$

as well as the equilibrium of the stresses

$$f_{\mathrm{s}} = -p. \tag{22}$$

In view of the momentum equation restricted to $\hat{\Sigma}$, (21) becomes

$$\frac{\partial p}{\partial n} = \rho_{\mathrm{f}} \frac{\partial^2 d}{\partial t^2}. \tag{23}$$

We consider the interface variable $\lambda = \partial p / \partial n$; the associated Steklov-Poincaré interface problem expresses the equation (22) and takes the following form

$$S_{\mathrm{f}}(\lambda) + S_{\mathrm{s}}(\lambda) = \chi \quad \text{on } \hat{\Sigma}, \tag{24}$$

where $S_{\mathrm{f}}(\lambda)$ and $S_{\mathrm{s}}(\lambda)$ are linear Steklov-Poincaré operators, while χ is a right-hand side accounting for the boundary data. (We warn the reader that all along the previous sections the right hand side was indeed incorporated in the definition of the nonlinear Steklov-Poincaré operators).

Using the notations of Sect. 2.1, the operators in (24) may be written explicitly as

$$S_{\mathrm{f}} : \lambda \mapsto \text{ find } (\boldsymbol{u}(\lambda), p(\lambda)) : \begin{cases} \mathrm{Fluid}(\boldsymbol{u}(\lambda), p(\lambda); 0, 0) \\ \dfrac{\partial p(\lambda)}{\partial n} = \lambda \text{ on } \hat{\Sigma} \end{cases}$$
$$\mapsto S_{\mathrm{f}}(\lambda) = p(\lambda)_{|\hat{\Sigma}}, \tag{25}$$

and

$$\begin{aligned} S_{\mathrm{s}} : \lambda \mapsto \text{find } d(\lambda) : \frac{\partial^2 d(\lambda)}{\partial t^2} &= \frac{\lambda}{\rho_{\mathrm{f}}} \\ \mapsto S_{\mathrm{s}}(\lambda) = \rho_{\mathrm{s}} h_{\mathrm{s}} \frac{\partial^2 d(\lambda)}{\partial t^2} - kGh_{\mathrm{s}} \frac{\partial^2 d(\lambda)}{\partial x^2} &+ \frac{Eh_{\mathrm{s}}}{1-\nu^2} \frac{d(\lambda)}{R_0^2} \quad \text{on } \hat{\Sigma}. \end{aligned} \tag{26}$$

For the numerical approximation, the fluid equations (19) reduce to the Laplace equation for the pressure

$$\mathrm{Fluid}(p; p_{\mathrm{in}}, 0) : \quad \Delta p = 0 \text{ in } \Omega^{\mathrm{f}},$$

with the same boundary conditions as in (19). Then, both the pressure p and the displacement d are discretized using the Galerkin method with $\mathbb{P}1$ (piecewise linear) finite elements. Moreover, the Newmark method is used for the time-discretization of the structure problem.

In (26), d can be computed as the solution of an ordinary partial differential equation. At the discrete level, d^{n+1} can be retrieved by finite differences from d^n and $\dot{d}^n$ computed at the previous time step by the Newmark method.

We have applied the Richardson method (11) considering the following preconditioners:

a1) $P^{-1} = P_{DN}^{-1} = \bar{S}_{\mathrm{f}}^{-1}$, i.e. the fluid problem acts as preconditioner;
b1) $P^{-1} = P_{ND}^{-1} = \bar{S}_{\mathrm{s}}^{-1}$, i.e. the structure problem acts as preconditioner;
c1) $P^{-1} = P_{NN}^{-1} = \alpha_{\mathrm{f}}^k \bar{S}_{\mathrm{f}}^{-1} + \alpha_{\mathrm{s}}^k \bar{S}_{\mathrm{s}}^{-1}$, i.e. both problems enter in the preconditioner.

Since the operators S_{f} and S_{s} are linear, they coincide with S'_{f} and S'_{s}, respectively.

The acceleration parameters ω^k for preconditioners a1) and b1) and α_{f}^k, α_{s}^k for c1) have been computed using the Aitken extrapolation techniques (12) and (16), respectively.

We have considered several time steps δt and space discretization parameters h in order to test the three above preconditioners. The results are reported in table 2.

As we can see both preconditioners P_{DN}^{-1} and P_{ND}^{-1} give satisfactory convergence results and we remark that in both cases the relaxation parameters ω^k computed using the Aitken approach belong to the range of stability indicated in [2]. However, the Neumann-Neumann type preconditioner c1) allows us to improve the convergence rate of the algorithm, sensibly reducing the number of iterations required to satisfy a convergence test with tolerance 10^{-5}.

Precond.	P_{DN}^{-1}		P_{ND}^{-1}		P_{NN}^{-1}		
δt	ω_{mean}	iter.	ω_{mean}	iter.	$(\omega\alpha_{\rm f})_{mean}$	$(\omega\alpha_{\rm s})_{mean}$	iter.
1.*e*-4	0.26	45	0.33	101	0.16	0.10	33
1.*e*-5	0.26	44	0.29	87	0.16	0.10	33
1.*e*-6	0.26	44	0.29	87	0.17	0.09	33

Precond.	P_{DN}^{-1}		P_{ND}^{-1}		P_{NN}^{-1}		
h	ω_{mean}	iter.	ω_{mean}	iter.	$(\omega\alpha_{\rm f})_{mean}$	$(\omega\alpha_{\rm s})_{mean}$	iter.
0.3	0.35	33	0.29	86	0.23	0.07	27
0.15	0.26	44	0.29	87	0.16	0.10	33
0.1	0.22	54	0.30	86	0.13	0.11	36
0.05	0.16	75	0.29	85	0.09	0.14	43
0.015	0.09	131	0.36	107	0.04	0.19	55

Table 2. Number of iterations for the choice of the preconditioners defined in a1), b1), c1) and mean value of the acceleration parameters. The values $h = 0.15$ and $\delta t = 1.e\text{-}5$ have been used for the upper and the lower table, respectively.

In Fig. 2 we have represented the relaxation parameters $\omega^k\alpha_{\rm f}^k$ and $\omega^k\alpha_{\rm s}^k$ (obtained for fixed $h = 0.15$ and $\delta t = 1.e\text{-}5$) which characterize the preconditioner P_{NN}^{-1} defined in c1) and whose mean values have been reported in table 2. Notice that they sensibly vary at each iteration.

Finally, we have tested the Richardson method (11) with preconditioner P_{NN}^{-1} as in c1) with respect to different values of the wall density $\rho_{\rm s}h_{\rm s}$, i.e. we have multiplied and divided by 10 the actual density. We have set $h = 0.15$ and $\delta t = 1.e\text{-}5$; as shown in table 3, the Aitken acceleration (16) can effectively compute the relaxation parameters assuring good convergence results.

In a second stage, we have slightly changed our framework by considering as interface variable the stresses on $\hat{\Sigma}$: $\sigma = -p\,(= \sigma_{\rm s})$ on $\hat{\Sigma}$ (rather than the displacement λ given by (23)). This yields a generalization of the so-called FETI method, which in the case of standard elliptic boundary value problems can be seen as the dual approach to (24) (see [21]). In fact, in this case we have to solve the interface equation

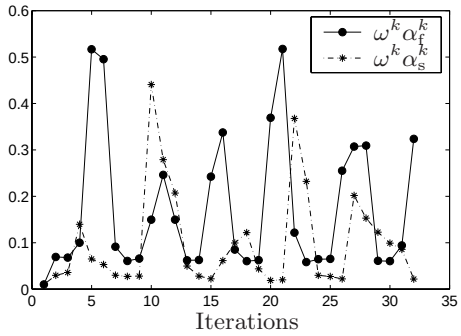

Fig. 2. Relaxation parameters $\omega^k\alpha_{\rm f}^k$ and $\omega^k\alpha_{\rm s}^k$ computed using (16).

wall density	$(\omega\alpha_{\rm f})_{mean}$	$(\omega\alpha_{\rm s})_{mean}$	iter.
$0.1 \cdot \rho_{\rm s}h_{\rm s}$	0.60	0.005	15
$1 \cdot \rho_{\rm s}h_{\rm s}$	0.16	0.10	33
$10 \cdot \rho_{\rm s}h_{\rm s}$	0.02	0.60	25

Table 3. Number of iterations with respect to different wall densities $\rho_{\rm s}h_{\rm s}$ for the preconditioner c1) corresponding to the values $h = 0.15$, $\delta t = 1.e\text{-}5$.

$$S_{\rm f}^{-1}(\sigma) + S_{\rm s}^{-1}(\sigma) = \widetilde{\chi} \tag{27}$$

defined through the inverses of the Steklov-Poincaré operators (25) and (26). A similar framework can be set up for the nonlinear coupled problem (1)-(3) as well.

Here the preconditioners become:

a2) $P^{-1} = \bar{S}_{\rm f}$;
b2) $P^{-1} = \bar{S}_{\rm s}$;
c2) $P^{-1} = \alpha_{\rm f}^k \bar{S}_{\rm f} + \alpha_{\rm s}^k \bar{S}_{\rm s}$

and, again, the Aitken strategy has been used to accelerate the convergence.

We have considered several time steps δt and space discretization parameters h in order to test the effectiveness of the preconditioners based on the FETI approach. The results are reported in table 4.

Also in this case we can see that the combination of both the fluid and the structure operators gives a better preconditioner than each one taken separately.

6 Conclusions

In this paper we have considered a general fluid-structure interaction problem. For its numerical solution we have analyzed three different approaches,

Precond.	$\bar{S}_\mathrm{f}$		$\bar{S}_\mathrm{s}$		$\alpha_\mathrm{f}^k \bar{S}_\mathrm{f} + \alpha_\mathrm{s}^k \bar{S}_\mathrm{s}$		
δt	ω_{mean}	iter.	ω_{mean}	iter.	$(\omega\alpha_\mathrm{f})_{mean}$	$(\omega\alpha_\mathrm{s})_{mean}$	iter.
1.*e*-4	0.19	61	0.42	54	0.10	0.17	36
1.*e*-5	0.18	59	0.36	63	0.10	0.17	37
1.*e*-6	0.18	60	0.36	64	0.10	0.17	37

Precond.	$\bar{S}_\mathrm{f}$		$\bar{S}_\mathrm{s}$		$\alpha_\mathrm{f}^k \bar{S}_\mathrm{f} + \alpha_\mathrm{s}^k \bar{S}_\mathrm{s}$		
h	ω_{mean}	iter.	ω_{mean}	iter.	$(\omega\alpha_\mathrm{f})_{mean}$	$(\omega\alpha_\mathrm{s})_{mean}$	iter.
0.3	0.17	56	0.50	45	0.08	0.27	31
0.15	0.18	59	0.36	63	0.10	0.17	37
0.1	0.18	61	0.31	77	0.11	0.13	40
0.05	0.19	63	0.22	109	0.12	0.08	46
0.015	0.20	64	0.13	191	0.15	0.03	56

Table 4. Number of iterations for the choice of the preconditioners a2), b2), c2) and mean value of the acceleration parameters. The values $h = 0.15$ and $\delta t = 1.e\text{-}5$ have been used for the upper and the lower table, respectively.

one based on a fixed-point iteration, another on Newton's method, a further on domain decomposition iterations. In particular, we have focused on the domain decomposition reformulation and we have proposed a general Richardson algorithm to solve the associated Steklov-Poincaré interface equation, with several choices of preconditioner.

The analogies with the two other approaches are discussed, and a numerical test on a simplified linear model for blood-flow simulation in an arterial vessel is investigated. The results show in particular that the so-called Neumann-Neumann preconditioner has the potential of handling efficiently significant variations of numerical discretization parameters as well as physical parameters.

Acknowledgement. This research has been supported by the Swiss National Science Foundation (Project 20-101800), by the European Community's RTN Project HPRN-CT-2002-00270 HaeMOdel, by the Italian reseach project Cofin2003-MIUR "Numerical Modeling for Scientific Computing and Advanced Applications" and by INDAM.

References

1. S. Anicic and A. Léger. Formulation bidimensionnelle exacte du modèle de coque 3D de Kirchhoff-Love. *C. R. Acad. Sci. Paris Sér. I Math.*, 329(8):741–746, 1999.
2. P. Causin, J.-F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. Technical Report 5084, INRIA, 2004.

3. D. Chapelle and K. Bathe. *The Finite Element Analysis of Shells - Fundamentals*. Springer Verlag, 2003.
4. S. Deparis. *Numerical analysis of axisymmetric flows and methods for fluid-structure interaction arising in blood flow simulation*. PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL), 2004.
5. S. Deparis, J.-F. Gerbeau, and X. Vasseur. GMRES preconditioning and accelerated Quasi-Newton algorithm and application to fluid structure interaction. In preparation, 2004.
6. M. Discacciati. An operator-splitting approach to nonoverlapping domain decomposition. Technical Report 14, EPFL, 2004. Submitted to Numer. Math.
7. M. Fernández and M. Moubachir. A Newton method using exact Jacobians for solving fluid-structure coupling. Technical Report 5085, INRIA, 2004.
8. J.-F. Gerbeau and M. Vidrascu. A quasi-Newton algorithm based on a reduced model for fluid structure problems in blood flows. *M2AN*, 37(4):631–648, 2003.
9. M. Heil. An efficient solver for the fully-coupled solution of large-displacement fluid-structure interaction problems. *Comput. Methods Appl. Mech. Engrg.*, 2003. In press.
10. P. Le Tallec. *Introduction à la Dynamique des Structures*. Ellipse, Paris, 2000.
11. P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. *Comput. Methods Appl. Mech. Engrg.*, 190(24-25):3039–3067, 2001.
12. H. Matthies and J. Steindorf. Numerical efficiency of different partitioned methods for fluid-structure interaction. *Z. Angew. Math. Mech.*, 2(80):557–558, 2000.
13. H. Matthies and J. Steindorf. Partitioned strong coupling algorithms for fluid-structure interaction. *Computer & Structures*, 81:805–812, 2003.
14. D. P. Mok and W. A. Wall. Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. In K. S. W. Wall, editor, *Trends in Computational Structural Mechanics*. K.U. Bletzinger,CIMNE, Barcelona, 2001.
15. D. P. Mok, W. A. Wall, and E. Ramm. Accelerated iterative substructuring schemes for instationary fluid-structure interaction. In K. Bathe, editor, *Computational Fluid and Solid Mechanics*, pages 1325–1328. Elsevier, 2001.
16. F. Nobile. *Numerical approximation of fluid-structure interaction problems with application to haemodynamics*. Thesis n. 2548, École Polytechnique Fédérale de Lausanne (EPFL), 2001.
17. A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer, New York – Berlin – Heidelberg, 2000.
18. A. Quarteroni, M. Tuveri, and A. Veneziani. Computational vascular fluid dynamics: problems, models and methods. *Comp. Vis. Science*, 2:163–197, 2000.
19. A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford Science Publications, Oxford, 1999.
20. T. Tezduyar. Finite element methods for fluid dynamics with moving boundaries and interfaces. *Arch. Comput. Methods Engrg.*, 8:83–130, 2001.
21. A. Toselli and O. Widlund. *Domain Decomposition Methods*. Springer, New York, 2004. In press.