

# Fluid-structure algorithms based on Steklov-Poincaré operators

S. Deparis\*, M. Discacciati†, G. Fourestey‡, A. Quarteroni†§

April 25, 2005

## Abstract

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

## 1 Introduction

Fluid-structure interaction problems are of utmost importance in applied mathematics. They range from aeroelastic problems, such as airflows around rigid structures, to haemodynamics, including for instance blood flow in large arteries. From the numerical viewpoint, fluid-structure interactions require the solution of coupled fluid and structure models. In aeroelastic simulations (see [1; 2; 3]), where the inertia of the structure is much greater than the one of the fluid, the computation does not require sub-cycling to achieve convergence (see [4; 5; 6]). However, this is not the case in blood flow simulations: here, since the density of the structure is comparable to the density of the fluid, the stability of numerical simulations of fluid-structure interactions relies heavily on the accuracy in solving the nonlinear coupled problem at each time step [7; 8; 9; 10; 11]. Consequently, implicit schemes must be used in order to achieve energetic balance and stability.

An investigation of these aspects was presented in [12], where a numerical analysis of a simplified fluid-structure interaction problems has been carried out for implicit and staggered algorithms taking into account the so-called added-mass effect. The authors show why numerical instabilities may occur under these

---

\*Mechanical Engineering Department MIT, 77 Mass Ave Rm 3-264, Cambridge MA 02139, USA, [deparis@mit.edu](mailto:deparis@mit.edu)

†IACS, Chair of Modeling and Scientific Computing, École Polytechnique Fédérale de Lausanne, CH-1015, Lausanne, Switzerland, [marco.discacciati@oeaw.ac.at](mailto:marco.discacciati@oeaw.ac.at)

‡[gilles.fourestey@epfl.ch](mailto:gilles.fourestey@epfl.ch)

§MOX, Politecnico di Milano, P.zza Leonardo da Vinci 32, 20133 Milano, Italy, [alfio.quarteroni@epfl.ch](mailto:alfio.quarteroni@epfl.ch)

combinations of physical parameters when using loosely coupled time advancing schemes.

Standard strategies to solve the nonlinear strongly coupled problem are fixed point based methods [13]. Unfortunately, these methods are very slow to converge (even if several acceleration strategies may improve their efficiency) and in some cases may fail to converge [10; 12; 14].

Recent advances suggest the use of Newton based methods for their fast convergence [14; 15; 16; 17]. They rely on the evaluation of the Jacobian associated to the fluid-solid coupled state equations. More precisely, the critical step consists in the evaluation of the *cross* Jacobian [18], which expresses the sensitivity of the fluid state to solid motions. This differentiation can be made using finite difference approximations (see, e.g., [18]), or by replacing the tangent operator of the coupled system by a simpler one [19; 20; 21]. However, in both cases these approximations may deteriorate or prevent the overall convergence. A Newton method with exact Jacobian has been investigated both mathematically and numerically in [15].

In this paper, we present another possible strategy: to adopt numerical algorithms based on domain decomposition techniques, which exploit the physically decoupled structure of the problem itself, and allow its solution, at a given time step, to be obtained through a sequence of independent solves involving each subproblem separately.

A first approach in this direction can be found in [11; 22], where the coupling between Stokes equations and a linearized shell model is considered. The global problem is reduced to a linear interface equation where the only unknown is the displacement of the interface separating the fluid and the structure. The analysis of the Steklov-Poincaré operators associated to the fluid and shell models is developed, and a Richardson scheme with the shell operator acting as preconditioner is proposed and tested.

Another instance is presented by Mok and Wall [23], who proposed an iterative substructuring method requiring, at each step, the independent solution of a fluid and a structure subproblem, supplemented with suitable Dirichlet or Neumann boundary condition on the interface.

One of the advantages of such an approach is that the whole problem is reduced to an equation involving only interface variables. In this respect, it can be regarded as a special instance of *heterogeneous domain decomposition problems* which arise whenever in the approximation of certain physical phenomena, two (or more) different kinds of boundary value problems hold within two disjoint subregions of the computational domain (see, e.g., [24]).

The outline of this paper is as follows. We will first describe the general formulations for the fluid and the structure. Then, we will define the numerical methods associated with these formulations. The third part will be dedicated to the interface equations associated with the coupled problem. Finally, we will show some numerical results produced by a 3D research code for blood-wall interaction in a simple 3D cylindrical vessel as well as in a more complex bifurcating channel representing the human carotid artery.

## 2 Problem setting

When dealing with fluid-structure interactions under relatively large displacement, to describe the evolution of the fluid and the structure domain in time, we adopt the ALE (Arbitrary Lagrangian Eulerian) formulation for the fluid (see [25; 26]) and a purely Lagrangian framework for the structure. We denote by  $\Omega(t)$  the moving domain composed of the deformable structure  $\Omega^s(t)$  and the fluid sub-domain  $\Omega^f(t)$ . The evolution of  $\Omega(t)$  can be described using an injective mapping (see fig. 1):

$$\mathbf{x} : \Omega_0 \times \mathbb{R}^+ \longrightarrow \mathbb{R}^3. \quad (1)$$

The position of any point  $x_0 \in \Omega_0^f$  and  $x_0 \in \Omega_0^s$  at a time  $t$  is given by

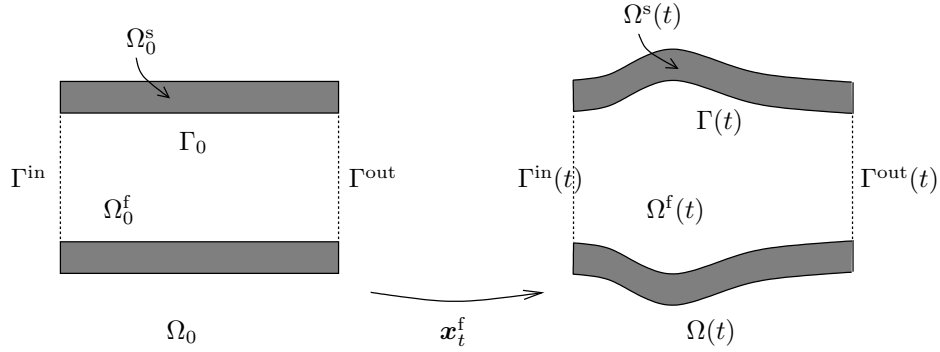


Figure 1: ALE mapping

$\mathbf{x}^f(x_0, t) = \mathbf{x}|_{\Omega_0^f}$  and  $\mathbf{x}^s(x_0, t) = \mathbf{x}|_{\Omega_0^s}$ . In particular, if we denote by  $\mathbf{d}^s(x_0, t)$  the displacement of the solid at a time  $t$ , we can define the following mapping:

$$\begin{aligned} \forall t, \Omega_0^s &\rightarrow \Omega^s(t), \\ x_0 &\rightarrow \mathbf{x}^s(x_0, t) = x_0 + \mathbf{d}^s(x_0, t), \quad x_0 \in \Omega_0^s. \end{aligned} \quad (2)$$

Likewise, we can define the mapping for the fluid domain:

$$\begin{aligned} \forall t, \mathbf{x}_t^f : \Omega_0^f &\rightarrow \Omega^f(t), \\ x_0 &\rightarrow \mathbf{x}^f(x_0, t) = x_0 + \mathbf{d}^f(x_0, t), \quad x_0 \in \Omega_0^f. \end{aligned} \quad (3)$$

Since the fluid displacement and velocity must match on the interface  $\Gamma_0$  with the structure ones, the fluid domain displacement  $\mathbf{d}^f$  can be defined as an extension of the solid interface displacement  $\mathbf{d}_{|\Gamma_0}^s$ :

$$\mathbf{d}^f = Ext(\mathbf{d}_{|\Gamma_0}^s).$$

$Ext$  can be chosen, e.g., as the harmonic extension operator. In that case, the fluid displacement is computed by solving

$$\begin{cases} -\kappa \Delta \mathbf{d}^f = 0 & \text{in } \Omega_0^f \\ \mathbf{d}_{|\Gamma_0}^f = \mathbf{d}_{|\Gamma_0}^s, \end{cases} \quad (4)$$

where  $\kappa$  is a properly chosen diffusion coefficient. However, we have to ensure that the mapping  $\mathbf{x}_t^f$  is a diffeomorphism for all  $t$  and that  $\mathbf{x}_t^f(x_0)$  is differentiable with respect to  $t$  for all  $x_0$  in  $\Omega_0^f$ . In other cases, different extension strategies may be pursued see e.g., [27; 28], and [29]. Finally, we introduce the following notations:

$$\mathbf{w}^f = \frac{\partial \mathbf{x}_t^f}{\partial t} = \frac{\partial \mathbf{d}^f}{\partial t}, \quad (5)$$

$$\mathbf{F} = \frac{\partial \mathbf{x}_t^f}{\partial x_0}, \quad (6)$$

$$J = \det \mathbf{F}, \quad (7)$$

$$\left. \frac{\partial \mathbf{u}}{\partial t} \right|_{x_0} (x, t) = \frac{d\mathbf{u}(\mathbf{x}_t^f(x_0), t)}{dt} \text{ with } x_0 = \mathbf{x}_t^{f^{-1}}(x), \quad (8)$$

which represent the velocity and the deformation gradient of the domain.

We assume the fluid to be Newtonian, viscous and incompressible, so that its behavior is described by the following fluid state problem: given the boundary data  $\mathbf{u}_{\text{in}}$ ,  $\mathbf{g}_f$  and  $\mathbf{f}_f$ , as well as  $\mathbf{w}^f$  and the forcing term  $\mathbf{d}^f$ , the velocity field  $\mathbf{u}$  and the pressure  $p$  satisfy the momentum and continuity equations:

$$\left\{ \begin{array}{l} \rho_f \left( \left. \frac{\partial \mathbf{u}}{\partial t} \right|_{x_0} + (\mathbf{u} - \mathbf{w}^f) \cdot \nabla \mathbf{u} \right) - \text{div}[\boldsymbol{\sigma}_f(\mathbf{u}, p)] = \mathbf{f}_f \quad \text{in } \Omega^f(t), \\ \text{div } \mathbf{u} = 0 \quad \text{in } \Omega^f(t), \\ \mathbf{u} = \mathbf{u}_{\text{in}} \quad \text{on } \Gamma^{\text{in}}(t), \\ \boldsymbol{\sigma}_f(\mathbf{u}, p) \cdot \mathbf{n}_f = \mathbf{g}_f \quad \text{on } \Gamma^{\text{out}}(t), \end{array} \right. \quad (9)$$

where  $\rho_f$  is the fluid density,  $\mu$  its viscosity,  $\boldsymbol{\sigma}_f(\mathbf{u}, p) = -pId + 2\mu\boldsymbol{\epsilon}(\mathbf{u})$  the Cauchy stress tensor ( $Id$  is the identity matrix,  $\boldsymbol{\epsilon}(\mathbf{u}) = (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)/2$  the strain rate tensor). Note that (9) does not univocally define a solution  $(\mathbf{u}, p)$  as no boundary data are prescribed on the interface  $\Gamma(t)$ .

Similarly, for given vector functions  $\mathbf{g}_s$ ,  $\mathbf{f}_s$ , we consider the following structure problem whose solution is  $\mathbf{d}$ :

$$\left\{ \begin{array}{l} \rho_s \frac{\partial^2 \mathbf{d}^s}{\partial t^2} - \text{div}_{|x_0}(\boldsymbol{\sigma}_s(\mathbf{d}^s)) = \mathbf{f}_s \quad \text{in } \Omega_0^s, \\ \boldsymbol{\sigma}_s(\mathbf{d}^s) \cdot \mathbf{n}_s = \mathbf{g}_s \quad \text{on } \partial\Omega_0^s \setminus \Gamma_0, \end{array} \right. \quad (10)$$

where  $\boldsymbol{\sigma}_s(\mathbf{d}^s)$  is the first Piola–Kirchhoff stress tensor,  $\gamma$  is a coefficient accounting for possible viscoelastic effects, while  $\mathbf{g}_s$  represents the normal traction on external boundaries and  $\mathbf{n}_f$  and  $\mathbf{n}_s$  are the outward normals of respectively the fluid and the solid domain. Similarly to what we have noticed for (9), problem (10) can not define univocally the unknown  $\mathbf{d}^s$  because a boundary value on  $\Gamma_0$  is missing.

As a simple example, in our numerical simulations we have used the Saint-Venant Kirchhoff three-dimensional elastic model (see, e.g., [30]) where the solid stress is defined as

$$\boldsymbol{\sigma}_s = 2\mu^l \boldsymbol{\epsilon}(\mathbf{d}^s) + \lambda^l \text{div}(\mathbf{d}^s)Id.$$

Here,  $\epsilon(\mathbf{d}^s) = \frac{1}{2} (\nabla \mathbf{d}^s + (\nabla \mathbf{d}^s)^T)$ ,  $\mu^l$  and  $\lambda^l$  are the Lamé constants. Other models could be chosen for the structure depending on the specific problem at hand. The reader may refer, e.g., to [31; 32; 33].

When coupling the two problems together, the “missing” boundary conditions are indeed supplemented by suitable matching conditions on the reference interface  $\Gamma_0$ . More precisely, if we denote by  $\lambda(t)$  the interface variable corresponding to the displacement  $\mathbf{d}^s$  on  $\Gamma_0$ , at any time  $t$  the coupling conditions on the reference interface  $\Gamma_0$  are

$$\begin{cases} \mathbf{x}_t^f &= x_0 + \lambda \\ \mathbf{u} \circ \mathbf{x}_t^f &= \frac{\partial \lambda}{\partial t}, \\ (\boldsymbol{\sigma}_f(\mathbf{u}, p) \cdot \mathbf{n}_f) \circ \mathbf{x}_t^f &= -\boldsymbol{\sigma}_s(\mathbf{d}^s) \cdot \mathbf{n}_s. \end{cases} \quad (11)$$

The system of equations (3), (9)-(11) identifies our coupled fluid-structure problem.

### 3 Decoupled weak formulation

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  $t^{n+1} = t^n + \delta t$ . If no ambiguity occurs, all the quantities will be referred to at time  $t = t^{n+1}$ .

If we are given a displacement of the interface  $\lambda(t^{n+1})$  at the time  $t^{n+1}$ , we can find its harmonic extension on the fluid domain by solving the following variational formulation of (4):

find  $\mathbf{d}_{t^{n+1}}^f \in H_0^1(\Omega_0^f)$  such that

$$\begin{cases} \int_{\Omega_0^f} \nabla \mathbf{d}_{t^{n+1}}^f \cdot \nabla \phi = 0 & \forall \phi \in H_0^1(\Omega_0^f) \\ \mathbf{d}_{t^{n+1}}^f = \lambda(t^{n+1}) & \text{on } \Gamma_0. \end{cases} \quad (12)$$

Then we set  $\mathbf{w}_{|\Gamma^{n+1}}^{f,n+1} = (\mathbf{d}_{t^{n+1}}^f - \mathbf{d}_{t^n}^f) / \delta t$  to compute the velocity of the structure domain and we compute the velocity and pressure of the fluid at time  $t^{n+1}$  by solving:

find  $(\mathbf{u}^{n+1}, p^{n+1}) = (\mathbf{u}(t^{n+1}), p(t^{n+1})) \in V^f(t^{n+1}) \times Q^f(t^{n+1})$  such that

$$\begin{cases} \frac{1}{\delta t} \int_{\Omega^f(t^{n+1})} \rho_f \mathbf{u}^{n+1} \mathbf{v}^f + \int_{\Omega^f(t^{n+1})} \rho_f [(\mathbf{u}^{n+1} - \mathbf{w}^{f,n+1}) \cdot \nabla \mathbf{u}^{n+1}] \mathbf{v}^f \\ + \mu \int_{\Omega^f(t^{n+1})} \boldsymbol{\sigma}_f(\mathbf{u}^{n+1}, p^{n+1}) \mathbf{v}^f = \frac{1}{\delta t} \int_{\Omega^f(t^{n+1})} \rho_f \mathbf{u}^n \mathbf{v}^f + \int_{\Gamma^{\text{out}}(t^{n+1})} \mathbf{g}^f \mathbf{v}^f \\ \int_{\Omega^f(t^{n+1})} q^f \operatorname{div} \mathbf{u}^{n+1} = 0 \end{cases} \quad (13)$$

for all  $(\mathbf{v}^f, q^f) \in V^f(t^{n+1}) \times Q^f(t^{n+1})$ , with

$$\begin{aligned} V^f(t) &= \{ \mathbf{v}^f \mid \mathbf{v}^f \circ \mathbf{x}_t^f \in H^1(\Omega_0^f)^3, \mathbf{v}^f = \mathbf{0} \text{ on } \Gamma^t \}, \\ Q^f(t) &= \{ q^f \mid q^f \circ \mathbf{x}_t^f \in L^2(\Omega_0^f) \}, \end{aligned}$$

and where the fluid domain  $\Omega^f(t^{n+1})$  is given by:

$$\Omega^f(t^{n+1}) = \mathbf{x}_{t^{n+1}}^f(\Omega_0^f).$$

We can then compute  $(\boldsymbol{\sigma}_f(\mathbf{u}^{n+1}, p^{n+1}) \cdot \mathbf{n}_f) \circ \mathbf{x}_t^f$  on  $\Gamma_0$ , which by (11) has to be equal to the structure normal stresses. We are assuming  $\mathbf{u}_{in} = \mathbf{0}$ , otherwise the modification is straightforward.

On the structure side, given the same displacement  $\lambda(t^{n+1})$ , we can define the structure variational problem as:

find  $(\mathbf{d}^{s,n+1}, \mathbf{w}^{s,n+1}) = (\mathbf{d}^s(t^{n+1}), \mathbf{w}^s(t^{n+1})) \in V^s(t^{n+1}) \times V^s(t^{n+1})$  such that

$$\begin{cases} \frac{2}{\delta t^2} \int_{\Omega_0^s} \rho_s \mathbf{d}^{s,n+1} \mathbf{v}^s - \frac{2}{\delta t^2} \int_{\Omega_0^s} \rho_s (\mathbf{d}^{s,n+1} + \delta t \mathbf{w}^{s,n+1}) \mathbf{v}^s + \int_{\Omega_0^s} \boldsymbol{\sigma}_s(\mathbf{d}^{s,n+1}) \cdot \nabla \mathbf{v}^s = 0 \\ \mathbf{w}^{s,n+1} = \frac{2}{\delta t} (\mathbf{d}^{s,n+1} - \mathbf{d}^{s,n} - \mathbf{w}^{s,n}) \\ \mathbf{d}^{s,n+1} = \lambda(t^{n+1}) \quad \text{on } \Gamma_0, \end{cases} \quad (14)$$

for all  $\mathbf{v}^s \in V^s$ , with  $V^s = \{\mathbf{v}^s \in H^1(\Omega_0^s)^3 \mid \mathbf{v}^s = 0 \text{ on } \partial\Omega_0^s \setminus \Gamma_0\}$ . As for the fluid, we can then compute the structure normal stresses on the interface as  $\boldsymbol{\sigma}_s(\mathbf{d}^{s,n+1}) \cdot \mathbf{n}_s$  on  $\Gamma_0$ .

If for a given interface displacement  $\lambda(t^{n+1})$  the fluid and structure normal stresses are at equilibrium, it means that the fluid-structure problem has been correctly solved. In general we impose the equilibrium in weak form, i.e., (for  $t = t^{n+1}$ ):

$$\int_{\Gamma(t)} \boldsymbol{\sigma}_f(\mathbf{u}, p) \cdot \mathbf{n}_f \mathbf{v}^f + \int_{\Gamma_0} \boldsymbol{\sigma}_s(\mathbf{d}^s) \cdot \mathbf{n}_s \mathbf{v}^s = 0 \quad \forall (\mathbf{v}^f, \mathbf{v}^s) \in V^f(t) \times V^s$$

such that  $\mathbf{v}^f \circ \mathbf{x}_t^f = \mathbf{v}^s$  on  $\Gamma_0$ . Both integrals can be computed as residuals of the weak form of the equations.

## 4 The interface equations associated to the coupled problem

We consider the coupled problem at a particular time  $t = t^{n+1}$ . In order to write the interface equation associated to the global fluid-structure problem, we introduce a fluid and structure operator as follows.

Let  $S_f$  be the Dirichlet-to-Neumann (D-t-N) fluid map such that to any given interface displacement  $\lambda$  it associates the normal stress

$$S_f(\lambda) = \boldsymbol{\sigma}_f := (\boldsymbol{\sigma}_f(\mathbf{u}, p) \cdot \mathbf{n}_f) \circ \mathbf{x}_t^f \text{ on } \Gamma_0,$$

where  $(\mathbf{u}, p)$  is the solution of the Navier-Stokes problem (13). On the other hand, we denote by  $S_s$  the D-t-N operator associated to the structure in  $\Gamma_0$  such

that to any given displacement  $\lambda$  of the interface  $\Gamma_0$  associates the normal stress exerted by the structure on  $\Gamma_0$ :

$$S_s(\lambda) = \sigma_s := (\boldsymbol{\sigma}_s(\mathbf{d}^s) \cdot \mathbf{n}_s) \text{ on } \Gamma_0,$$

where  $\mathbf{d}^s$  is the solution of (14).

Remark that in general  $S_f$  and  $S_s$  are nonlinear and their definitions involve also the terms due to the boundary conditions and the forcing terms.

Concerning the inverse of the solid operator, we can define  $S_s^{-1}$  as a Neumann-to-Dirichlet (N-t-D) map that at any given normal stress  $\sigma$  on  $\Gamma_0$  it associates the interface displacement  $\lambda(t^{n+1}) = \mathbf{d}^{s,n+1}$  by solving a structure problem analogous to (14), but with the Neumann boundary condition

$$\boldsymbol{\sigma}_s(\mathbf{d}^s) \cdot \mathbf{n}_s = \sigma \text{ on } \Gamma_0$$

and then computing the restriction on  $\Gamma_0$  of the displacement of the structure domain.

Moreover, we denote by  $S'_s$  the tangent operator associated to the structure problem and by  $(S'_s)^{-1}$  its inverse. The latter is a N-t-D map that to any given normal stress  $\sigma$  on  $\Gamma_0$  associates the corresponding displacement  $\lambda(t^{n+1})$  of the interface by solving the linearized structure problem with boundary condition  $\boldsymbol{\sigma}_s(\mathbf{d}^s) \cdot \mathbf{n}_s = \sigma$  on  $\Gamma_0$ . Analogously, by  $(S'_f)^{-1}$  we denote the inverse of the tangent operator  $S'_f$ . This is also a N-t-D map that for any given normal stress  $\sigma$  on  $\Gamma_0$  computes the corresponding displacement  $\lambda(t^{n+1})$  of the interface through the solution of linearized Navier-Stokes equations with the boundary condition  $(\boldsymbol{\sigma}_f(\mathbf{u}, p) \cdot \mathbf{n}_f) \circ \mathbf{x}^f = \sigma$  on  $\Gamma_0$ .

Using the definitions of the operators  $S_f$  and  $S_s$  and of their inverses, we can express the coupled fluid-structure problem in terms of the solution  $\lambda$  of a nonlinear equation defined only on  $\Gamma_0$ . More precisely, we can envisage three possible formulations for the interface equation which are all equivalent from a mathematical point of view, but give rise to different iterative methods.

First, we have the fixed-point formulation:

$$\text{find } \lambda \text{ such that } S_s^{-1}(-S_f(\lambda)) = \lambda \text{ on } \Gamma_0. \quad (15)$$

This is a classical formulation in fluid-structure interaction problems, but it is worth pointing out that here the fixed point is the displacement of the sole interface, whereas the classical fixed point algorithm is applied to the displacement of the whole solid domain (see, e.g., [10]).

The second possible approach is a slight modification of the previous equation (15):

$$\text{find } \lambda \text{ such that } S_s^{-1}(-S_f(\lambda)) - \lambda = 0 \text{ on } \Gamma_0, \quad (16)$$

which is more suitable for setting up a Newton iterative method. Again, this is applied solely to the interface displacement, instead of the whole solid displacement (see, e.g., [15]).

Finally, we have the Steklov-Poincaré equation:

$$\text{find } \lambda \text{ such that } S_f(\lambda) + S_s(\lambda) = 0 \text{ on } \Gamma_0. \quad (17)$$

## 4.1 Fixed point iterations

A standard algorithm to solve problem (15) is based on relaxed fixed point iterations. One iteration of the fixed point algorithm reads: for a given  $\lambda^k$ , compute

$$\lambda^{k+1} = \lambda^k + \omega^k (\bar{\lambda}^k - \lambda^k), \quad (18)$$

where

$$\bar{\lambda}^k = S_s^{-1}(-S_f(\lambda^k)).$$

The choice of the relaxation parameter  $\omega^k$  is crucial for the convergence of the method (see [12] for a recent analysis). An effective strategy for computing  $\omega^k$  is given by Aitken's method (see [13; 23; 34]).

The fluid and structure problems are solved separately and sequentially. In fact, each step  $k$  of the algorithm (18) implies:

1. apply  $S_f$  to a given displacement  $\lambda^k$ , i.e., compute the extension of  $\lambda^k$  to the entire fluid domain, solve the fluid problem in  $\Omega^f(t)$  with boundary condition  $\mathbf{u}|_{\Gamma(t)} \circ \mathbf{x}_t^f = (\lambda^k - \mathbf{d}_{|\Gamma(t)}^{f,n})/\delta t$  on  $\Gamma_0$ ; then compute the stress  $\sigma_f^k = (\boldsymbol{\sigma}_f(\mathbf{u}^k, p^k) \cdot \mathbf{n}_f)|_{\Gamma(t)} \circ \mathbf{x}_t^f$  on the interface;
2. apply the inverse of  $S_s$  to  $-\sigma_f^k$ , i.e., solve the structure problem in  $\Omega_s^s$  with boundary condition  $\sigma_s(\mathbf{d}^{s,k}) \cdot \mathbf{n}_s = -\sigma_f^k$  on  $\Gamma_0$ ; then compute the correction  $\bar{\lambda}^k$  of the displacement at the iterate  $k$ .

In general, the main drawback of this method is its slow convergence rate.

## 4.2 Newton algorithm

The Newton algorithm exploits the formulation (16). Let  $J(\lambda)$  denote the Jacobian of  $S_s^{-1}(-S_f(\lambda))$  in  $\lambda$ . Given  $\lambda^0$ , for  $k \geq 0$ , a step of the Newton algorithm associated to problem (16) reads:

$$\begin{aligned} (J(\lambda^k) - Id)\mu^k &= -(S_s^{-1}(-S_f(\lambda^k)) - \lambda^k), \\ \lambda^{k+1} &= \lambda^k + \omega^k \mu^k. \end{aligned} \quad (19)$$

In general, the parameter  $\omega^k$  can be computed, e.g., by a line search technique (see [35]). Note that the Jacobian of  $S_s^{-1}(-S_f(\lambda^k))$  in  $\lambda^k$  has the following expression:

$$J(\lambda^k) = -[S'_s(S_s^{-1}(-S_f(\lambda^k)))]^{-1} \cdot S'_f(\lambda^k) = -[S'_s(\bar{\lambda}^k)]^{-1} \cdot S'_f(\lambda^k). \quad (20)$$

The solution of the linear system (19) can be obtained using an iterative matrix-free method such as GMRES. We remark that while the computation of  $[S'_s(\bar{\lambda}^k)]^{-1} \cdot \delta\sigma$  (for any given  $\delta\sigma$ ) does only require the derivative with respect to the state variable at the interface, the computation of  $S'_f(\lambda^k) \cdot \delta\lambda$  implies also shape derivatives, since a variation in  $\lambda$  determines a variation of the fluid domain. This is a non-trivial task. In the literature, several approaches have been proposed to



solve exactly the tangent problem [15], or else to approximate it by either simpler models for the fluid [19; 21], or through finite differences schemes [9; 16; 18] (however, the lack of *a priori* criteria for selecting optimal finite difference infinitesimal steps may lead to a reduction of the overall convergence speed [21]).

### 4.3 Domain decomposition (or Steklov-Poincaré) formulation

A common drawback of the algorithms presented so far is that their implementation is purely sequential, while the domain decomposition formulation may allow us to set up parallel algorithms to solve the interface equation (17). Let us consider for example the preconditioned Richardson method.

Since the Steklov-Poincaré problem (17) 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., [24]). Given  $\lambda^0$ , for  $k \geq 0$ , the iterative method reads:

$$P(\lambda^{k+1} - \lambda^k) = \omega^k (-S_f(\lambda^k) - S_s(\lambda^k)) \quad (21)$$

with appropriate choice of the scalar  $\omega^k$ . Every equation should still be intended on  $\Gamma_0$ . The preconditioner  $P$ , that must be chosen appropriately, maps the interface variable onto the space of normal stresses, and may depend on the iterate  $\lambda^k$  or more generally on the iteration step  $k$ . In these cases we will denote it by  $P_k$ . The acceleration parameter  $\omega^k$  can be computed via the Aitken technique (see [34]). At each step  $k$ , algorithm (21) requires to solve separately the fluid and the structure problems and then to apply a preconditioner. Precisely,

1. apply  $S_f$  to  $\lambda^k$ , i.e., compute the extension of  $\lambda^k$  to the entire fluid domain, solve the fluid problem as already illustrated for algorithm (18), and compute the normal stress  $\sigma_f^k$ ;
2. apply  $S_s$  to  $\lambda^k$ , i.e., solve the structure problem with boundary condition  $\mathbf{d}_{|\Gamma(t)}^{s,k} = \lambda^k$  on  $\Gamma(t)$  and compute the normal stress  $\sigma_s^k$ ;
3. apply the preconditioner  $P^{-1}$  to the total stress  $\sigma^k = \sigma_f^k + \sigma_s^k$  on the interface.

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

### 4.4 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. 4.1 and 4.2.

#### 4.4.1 Dirichlet–Neumann and Neumann–Neumann preconditioners

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

$$P_k^{-1} = \alpha_f^k S_f'(\lambda^k)^{-1} + \alpha_s^k S_s'(\lambda^k)^{-1}, \quad (22)$$

for two given scalars  $\alpha_f^k$  and  $\alpha_s^k$ . From (22) we retrieve the following special cases:

- If  $\alpha_f^k = 0$  and  $\alpha_s^k = 1$ , then

$$P_k^{-1} = P_{DN}^{-1} = S_s'(\lambda^k)^{-1}.$$

We call it a *Dirichlet-Neumann preconditioner*, and

$$P_{DN}^{-1}(\sigma^k) = S_s'(\lambda^k)^{-1} (-S_f(\lambda^k) - S_s(\lambda^k));$$

- If  $\alpha_f^k = 1$  and  $\alpha_s^k = 0$ , then

$$P_k^{-1} = P_{ND}^{-1} = S_f'(\lambda^k)^{-1}.$$

We call  $P_{DN}$  a *Neumann-Dirichlet preconditioner* and

$$P_{ND}^{-1}(\sigma^k) = S_f'(\lambda^k)^{-1} (-S_f(\lambda^k) - S_s(\lambda^k));$$

- If  $\alpha_f^k + \alpha_s^k = 1$ , then

$$P_k^{-1} = P_{NN}^{-1} = \alpha_f^k S_f'(\lambda^k)^{-1} + \alpha_s^k S_s'(\lambda^k)^{-1}$$

and we call it *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.

It is possible to choose the parameters  $\alpha_f^k$ ,  $\alpha_s^k$  and  $\omega^k$  dynamically using a generalized Aitken technique.

**Remark 1** *If we consider a linear structure model and if we choose  $\alpha_f^k = 0$  and  $\alpha_s^k = 1$ , the algorithm (21) is equivalent to the fixed-point algorithm (18) (see Sect. 4.1). Indeed, from (21)*

$$\mu^k = (S_s'(\lambda^k))^{-1} (-S_f(\lambda^k) - S_s(\lambda^k)) = S_s^{-1} (-S_f(\lambda^k)) - \lambda^k,$$

hence  $\lambda^{k+1} = \lambda^k + \omega^k(\bar{\lambda}^k - \lambda^k)$ , which coincides with the last equality of (18).

## 4.5 The Newton algorithm on the Steklov-Poincaré equation

The genuine Newton algorithm applied to the Steklov-Poincaré problem (17) is retrieved by using the algorithm (21) (with  $\omega^k = 1$ ) and choosing at the step  $k$

$$P_k = S'_f(\lambda^k) + S'_s(\lambda^k). \quad (23)$$

Note that in order to invert  $P_k$  one must use a (preconditioned) iterative method (e.g., GMRES) and may approximate the tangent problems to accelerate the computations. The resolution algorithm now reads:

$$\begin{aligned} [S'_f(\lambda^k) + S'_s(\lambda^k)] \mu^k &= (-S_f(\lambda^k) - S_s(\lambda^k)) \\ \lambda^{k+1} &= \lambda^k + \omega^k \mu^k. \end{aligned} \quad (24)$$

As for the classical Newton approach,  $\mu^k$  can be computed using an iterative matrix-free method. Given a solid state displacement  $\lambda^k$ , the domain decomposition-Newton method thus reads: for  $k \geq 0$ ,

1. update the residual  $S_f(\lambda^k) + S_s(\lambda^k)$  by solving the fluid and the structure sub-problems;
2. solve the linear system (24) via the GMRES method in order to compute  $\mu^k$ ;
3. update the displacement  $\lambda^{k+1}$ . In our application we take  $\omega^k = 1$ , but in some cases it may be necessary to adopt a linesearch or an Aitken strategy.

The GMRES solver should be preconditioned in order to accelerate the convergence rate. To this aim, one can use the previously defined domain decomposition preconditioners. In our numerical tests, we have considered the Dirichlet-Neumann preconditioner  $S_s^{-1}$ , so that the preconditioned matrix of the GMRES method becomes:

$$[S_s^{-1}(\lambda^k)] \cdot [S'_f(\lambda_k) + S'_s(\lambda_k)] \quad (25)$$

### 4.5.1 Comparison with the Newton algorithm (19) on problem (16)

The Richardson algorithm (21) for the Steklov-Poincaré formulation (17) with preconditioner given by (22) (with  $\alpha_f^k = \alpha_s^k = 1$ ) is *not* equivalent to the Newton algorithm (19) applied to problem (16). In fact, the Newton algorithm (19) could be regarded as a Richardson method (21), choosing however a nonlinear preconditioner defined as

$$P_k(\mu) = S_s \left( S'_s(\bar{\lambda}^k)^{-1} \cdot (S'_f(\lambda^k) + S'_s(\bar{\lambda}^k)) \cdot \mu \right), \quad (26)$$

where  $\bar{\lambda}^k = S_s^{-1}(-S_f(\lambda^k))$ .

In this case, for  $\sigma^k = -(S_f(\lambda^k) + S_s(\lambda^k))$ , we would obtain

$$\begin{aligned} P_k^{-1}(\sigma^k) &= (S'_f(\lambda^k) + S'_s(\bar{\lambda}^k))^{-1} \cdot S'_s(\bar{\lambda}^k) \cdot S_s^{-1}(-S_f(\lambda^k) - S_s(\lambda^k)) \\ &= ([S'_s(\bar{\lambda}^k)]^{-1} \cdot S'_f(\lambda^k) + Id)^{-1} (S_s^{-1}(-S_f(\lambda^k)) - \lambda^k). \end{aligned}$$

We see that this is equivalent to (19). In fact (20) is equal to the first bracket in the last line.

**Remark 2** *Note that if (only) the structure is linear, the preconditioner defined in (26) is also linear and becomes*

$$P_k = S'_f(\lambda^k) + S'_s(\lambda^k),$$

*which is exactly (23). This is a Newton method applied to (16) or (17). 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 residual computation (operators  $S_f$  and  $S_s$ ) and the application of the preconditioner (operators  $S'_f$  and/or  $S'_s$ ).*

## 5 Numerical results

### 5.1 Straight cylindrical vessel

We compare the domain decomposition algorithms with the classical fixed point and Newton methods. We consider three different preconditioners:

1. the Dirichlet-Neumann preconditioner (here denoted as ‘Steklov-Poincaré DN’), i.e., the preconditioner is equal to the structure tangent operator  $S'_s$ ;
2. the Neumann-Neumann preconditioner (here denoted ‘Steklov-Poincaré NN’), i.e., a linear combination of the structure problem  $S'_s$  and an approximation of the linearized fluid problem  $S'_f$ . In particular, the last one is linearized by neglecting any shape derivative;
3. the domain decomposition-Newton method illustrated in Sect. 4.5 (‘DD-Newton’). The fluid tangent problem is considered as in [15] in its exact form. To invert (24) we apply the GMRES method either unpreconditioned, or preconditioned by ‘Steklov-Poincaré DN’.

The simulations were performed on a dual 2.8 Ghz Pentium 4 Xeon with 3 GB of RAM. The fluid is discretized by  $\mathbb{P}_1$ -bubble/ $\mathbb{P}_1$  finite elements and the solid by  $\mathbb{P}_1$  finite elements. All the methods give the same solution up to the tolerance required.

We simulate a pressure wave in a straight cylinder of length 5cm and radius 5cm at rest. The structure, whose thickness is 5mm, is considered linear and

clamped at both the inlet and the outlet. The fluid viscosity is set to  $\mu = 0.03$ , the Lamé constants to  $\mu^l = 1.15 \cdot 10^6$  and  $\lambda^l = 1.73 \cdot 10^6$ , the densities to  $\rho^f = 1$  and  $\rho^s = 1.2$ . We impose zero body forces and homogeneous Dirichlet boundary conditions on  $\partial\Omega_0^s \setminus \Gamma_0$ .

The fluid and the structure, both three dimensional, are initially at rest and a pressure (a normal stress, actually) of  $1.3332 \cdot 10^4$  *dynes/cm<sup>2</sup>* is set on the inlet for a time of  $3 \cdot 10^{-3}$  s. We used two meshes:

- a coarse mesh with 1050 nodes (4680 elements) for the fluid and 1260 nodes (4800 elements) for the solid (fig. 2);
- a fine mesh with 2860 nodes (14100 elements) for the fluid and 2340 nodes (9000 elements) for the solid (fig. 3).

A comparison with the classical coupling formulation have been conducted and results are displayed in tables 1 and 2. In these tables, ‘FS evals’ stands for the average number of evaluation per time step of either (15) or (17), while ‘Tangent evals’ represents the average number of evaluations of the corresponding linearized system per time step. We can see that, using the preconditioned Richardson method (21), a decrease in the number of FS evaluations with respect to the classical fixed point algorithm is obtained. However, the computational time of the domain decomposition formulation is slightly higher than that of the fixed point formulation. The reason is that the domain decomposition formulation requires to solve, at each iteration, the fluid and the structure subproblems, as well as the associated tangent problems, while the latter are indeed skipped by the fixed point procedure. Furthermore, since the solid operator is linear, the two approaches are very similar and since our research code is sequential, the parallel structure of the Steklov-Poincaré formulation (17) is not capitalized.



Figure 2: Coarse fluid (left) and structure (right) meshes



Figure 3: Refined fluid (left) and structure (right) meshes

Table 1: Comparison of the number of sub-iterations for the fixed point algorithm and the domain decomposition algorithm (coarse mesh)

Coarse mesh, $\Delta t = 0.001$			
Method	FS evals	Tangent evals	CPU time
Fixed point	19.8	0	1h16'
Steklov-Poincaré DN	19.8	19.8	1h17'
Steklov-Poincaré NN	17.9	17.9	1h42'
$\Delta t = 0.0005$			
Method	FS evals	Tangent evals	CPU time
Fixed point	32.1	0	3h27'
Steklov-Poincaré DN	29.2	29.2	3h50'
Steklov-Poincaré NN	22	22	4h20'

We compare now the Newton method (16) and the domain decomposition-Newton algorithm (see Sect. 4.5). In both cases, the Jacobians (20) and (23) are computed exactly (cf. [15]) and inverted by a GMRES method. The number of Newton iterations is equivalent, although the inversion of the Jacobian in ‘DD-Newton’ needs more GMRES iterations. Preconditioning GMRES by ‘Steklov-Poincaré DN’ reduces these iterations to the same as in ‘Newton’ and the CPU time is then equivalent. As before, the reasons reside in the linearity of the structure model and in the fact that our code is sequential.

The next steps are therefore to set up more sophisticated preconditioners for the Jacobian system, derived either from the classical domain decomposition

Table 2: Comparison of the number of sub-iterations for the fixed point algorithm and the domain decomposition algorithm (fine mesh)

Refined mesh, $\Delta t = 0.001$			
Method	FS evals	Tangent evals	CPU time
Fixed point	19.9	0	4h28'
Steklov-Poincaré DN	19.5	19.5	4h40'
Steklov-Poincaré NN	17.7	17.7	6h12'
$\Delta t = 0.0005$			
Method	FS evals	Tangent evals	CPU time
Fixed point	33	0	12h40'
Steklov-Poincaré DN	29.6	29.6	12h50'
Steklov-Poincaré NN	22.1	22.1	15h44'

theory or from lower dimensional models (in a multiscale approach, cf. [36]), and to consider a non-linear structure. The latter is of particular interest for example in haemodynamics when dealing with complex realistic geometries with relatively large displacements.

Table 3: Convergence time comparison between the exact Newton and the domain decomposition-Newton methods (coarse mesh)

Coarse mesh, $\Delta t = 0.001$			
Method	FS eval	Tangent evals	CPU time
Newton	3	12	0h56'
DD-Newton	3	24	1h30'
DD-Newton DN precondition	3	12	0h58'
$\Delta t = 0.0005$			
Newton	3	17	1h55'
DD-Newton	3	29	3h30'
DD-Newton DN precondition	3	17	2h10'
$\Delta t = 0.0001$			
Method	FS eval	Tangent evals	CPU time
Newton	3	19	11h41'
DD-Newton	3	35	16h21'
DD-Newton DN precondition	3	19	12h39'

Figure 4 shows the pressure wave propagation computed on the coarse mesh with a time step of  $\delta t = 1e^{-3}s$  at time  $t = 0.005s, 0.01s, 0.015s,$  and  $0.02s$ . The deformation is amplified by a factor 12. We can see that, at time  $t = 0.01s$ ,

Table 4: Convergence comparison of the computational time for the exact Newton and domain decomposition-Newton methods (fine mesh)

Refined mesh, $\Delta t = 0.001$			
Method	FS eval	Tangent evals	CPU time
Newton	3	12	3h39'
DD-Newton	3	30	4h56'
DD-Newton DN precondition	3	12	3h45'
$\Delta t = 0.0005$			
Newton	3	14	8h31'
DD-Newton	3	35	10h50'
DD-Newton DN precondition	3	14	8h40'
$\Delta t = 0.0001$			
Method	FS eval	Tangent evals	CPU time
Newton	3	19	26h40'
DD-Newton	3	37	40h26'
DD-Newton DN precondition	3	19	27h01'

the deformation reaches the end of the tube. Afterwards, at time  $t = 0.015s$  and  $t = 0.02s$ , a backward wave propagation is observed. This phenomenon can be explained by the fact that we clamped the structure and that we impose a vanishing fluid normal stress at the outlet. Setting proper boundary conditions in the case of physiological simulations is beyond the scope of this paper. The interested reader may refer, e.g., to the multiscale geometrical approach advocated in Quarteroni and Formaggia [36].

## 5.2 Carotid bifurcation

We simulate a pressure wave in the carotid bifurcation using the same fluid and structure characteristics as in the previous simulations. We solve the coupling using our ‘DD-Newton DN precondition’ algorithm. The mesh that we have used was computed using an original realistic geometry first proposed in [37]. The cast was produced by D. Liepsch (Fachhochschule München) and the computational model was developed by K. Perktold (Technische Universität Graz) (see [37] for more details).

The fluid and the structure are initially at rest and a pressure of  $1.3332 \cdot 10^4 \text{ dynes/cm}^2$  is set on the inlet for a time of  $3 \cdot 10^{-3} s$ . The average inflow diameter is  $0.67 \text{ cm}$ , the time step used is  $\delta t = 1e - 04$  and the total number of iterations is 200. Figure 5 displays the mesh used for the computations, Figure 6 the deformation at different time steps, while figures 7 to 10 show the displacement of the lower part of the carotid (the displacement is amplified 12 times). In these last figures, the vectors represent the velocity of the structure.



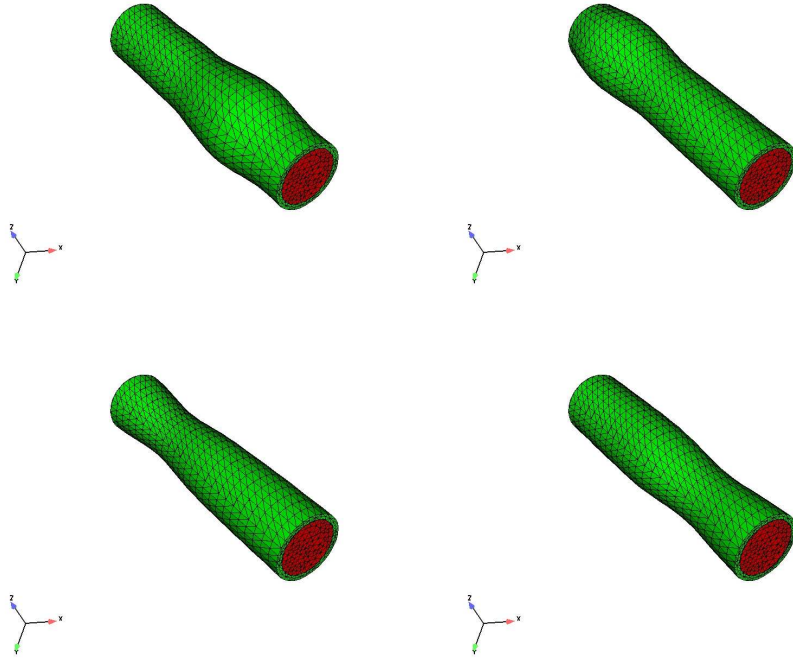


Figure 4: Fluid and solid solutions at time  $t = 0.005s$  (upper left),  $t = 0.010s$  (upper right),  $t = 0.015s$  (lower left),  $t = 0.020s$  (lower right) (coarse mesh)

Figure 11 represents the inflow flux computed at each iteration. We can observe three distinct reflections: After iteration 30, i.e., after  $3 \cdot 10^{-3}s$ , no pressure is imposed in the inflow, resulting in the decrease of the inflow flux. We can see that shortly after this phase, the blood flows backward for a short period of time. The same phenomenon is observed between iteration 108 and 150 (i.e., at time  $0.0108s$  and  $0.015s$ ). This happens after the pressure pulse enters the bifurcation, and the carotid wall shrinks at the intersection. The third reflection is caused by the pressure wave leaving the computational domain. The second reflection is physiological and is relevant in haemodynamics. The two others could be avoided by considering more sophisticated boundary conditions.

## 6 Conclusion

We have presented some new strong implicit coupled schemes to solve fluid-structure interaction problems stemming from a reformulation of the global problem as a Steklov-Poincaré interface equation (17). With respect to the



Figure 5: Coarse fluid (left) and structure (right) meshes

classical Newton or fixed point algorithms this approach requires to solve a nonlinear system whose dimension coincides with the number of degrees of freedom of the structural displacement on the interface, instead of those of the whole solid subdomain.

Firstly, we propose to solve the interface system using Aitken-accelerated Richardson iterations preconditioned by either Dirichlet-Neumann or Neumann-Neumann scaling operators (see Sect. 4.3). Numerical results have shown that the computational costs of these schemes is almost comparable to those of the fixed point method.

Then, we present a more sophisticated approach derived by combining our domain decomposition formulation with a Newton method. We obtain a domain decomposition-Newton method with a Dirichlet-Neumann preconditioned Jacobian, which requires the same computational effort as the classical Newton method, on a physically significant problem.

However, we point out that the results here presented have been obtained using a fully sequential code which do not exploit the possibility of solving local subproblems in a fully parallel setting. Indeed, this is a major advantage of the domain decomposition strategy, since the evaluation of the residual  $S_f(\lambda^k) + S_s(\lambda^k)$  at each Richardson iteration does not require to consider the structure and fluid operators in any specified order, as for the classical formulations (18) and (19).

In our simulations the computational cost for the solution of the structure problem is negligible with respect to that for the fluid. A parallel setting would be a real advantage when the effort to solve the structure problem becomes comparable to that needed for the fluid, e.g., for nonlinear structure models.

Moreover, the choice of suitable preconditioners which exploit, e.g., reduced models for the fluid part, is currently being investigated and it should lead to a

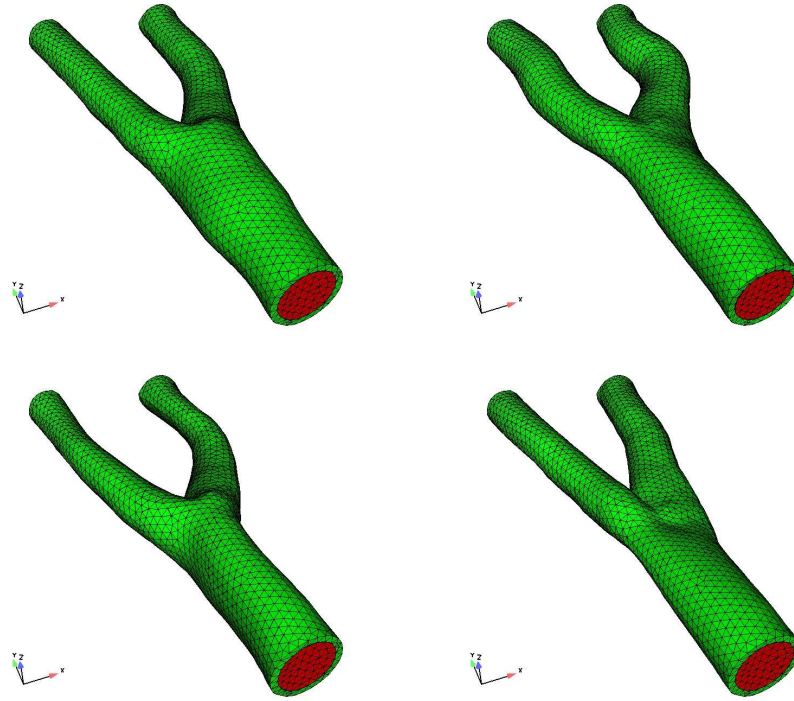


Figure 6: Carotid deformation at time  $t = 0.005s$  (upper left),  $t = 0.010s$  (upper right),  $t = 0.015s$  (lower left),  $t = 0.020s$  (lower right)

further reduced computational cost of the preconditioning step.

## Acknowledgments

All the computations in this paper were performed using a 3D research code developed at the École Polytechnique Fédérale de Lausanne, the Politecnico di Milano and INRIA Rocquencourt ([www.lifev.org](http://www.lifev.org), project manager at EPFL: Christophe Prud'homme [christophe.prudhomme@epfl.ch](mailto:christophe.prudhomme@epfl.ch)).

This research has been supported by the Swiss National Science Foundation (Project number 20-101-800) and by INDAM project “Integrazione di sistemi complessi in biomedicina: modelli, simulazioni, rappresentazioni”. The authors would also like to thank Miguel A. Fernandez for his expertise and precious advices on this work.

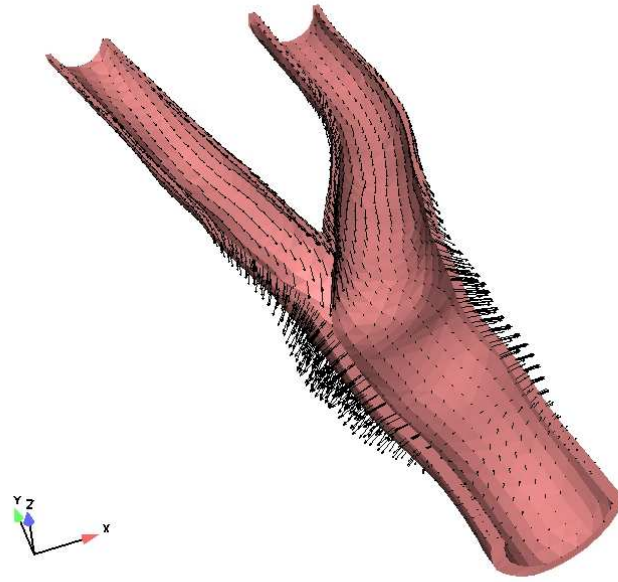


Figure 7: Structure deformation and velocity at time  $t = 0.005s$

## References

- [1] C. Farhat, M. Lesoinne, P. L. Tallec, Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity, *Comput. Methods Appl. Mech. Engrg* 157 (1998) 95–114.
- [2] C. Farhat, G. van der Zee, P. Geuzaine, Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity, *Comput. Methods Appl. Mech. Engrg.* (in press).
- [3] S. Piperno, C. Farhat, Partitioned procedures for the transient solution of coupled aeroelastic problems - Part II: Energy transfer analysis and three-dimensional applications, *Comput. Methods Appl. Mech. Engrg.* 190 (2001) 3147–3170.
- [4] G. Fourestey, S. Piperno, A second-order time-accurate ALE Lagrange-Galerkin method applied to wind engineering and control of bridge profiles, *Comput. Methods Appl. Mech. Engrg.* 193 (2004) 4117–4137.

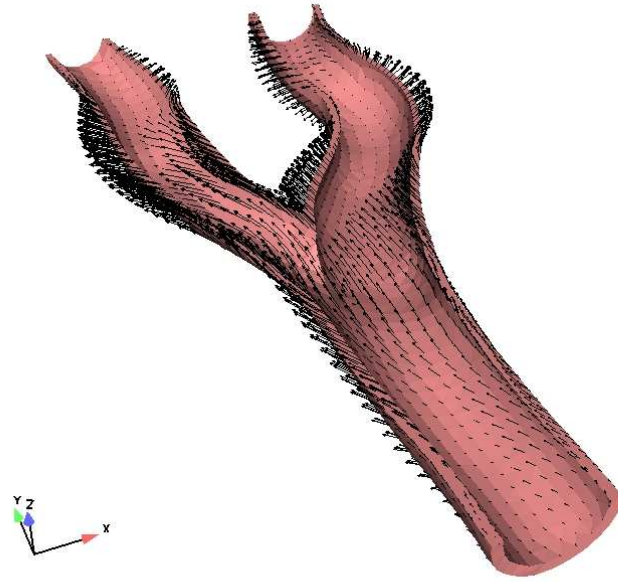


Figure 8: Structure deformation and velocity at time  $t = 0.010s$

- [5] S. Piperno, Explicit/implicit fluid-structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations, *Int. J. Num. Meth. Fluids* 25 (1997) 1207–1226.
- [6] S. Piperno, Numerical simulation of aeroelastic instabilities of elementary bridge decks, Tech. Rep. 3549, INRIA (1998).
- [7] S. Deparis, M. Fernández, L. Formaggia, Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions, *M2AN* 37 (4) (2003) 601–616.
- [8] H. Matthies, J. Steindorf, Partitioned but strongly coupled iteration schemes for nonlinear fluid-structure interaction, *Computer & Structures* 80 (2002) 1991–1999.
- [9] H. Matthies, J. Steindorf, Partitioned strong coupling algorithms for fluid-structure interaction, *Computer & Structures* 81 (2003) 805–812.
- [10] F. Nobile, Numerical Approximation of Fluid-Structure Interaction Problems with Application to Haemodynamics, Ph.D. thesis, École Polytechnique Fédérale de Lausanne (2001).

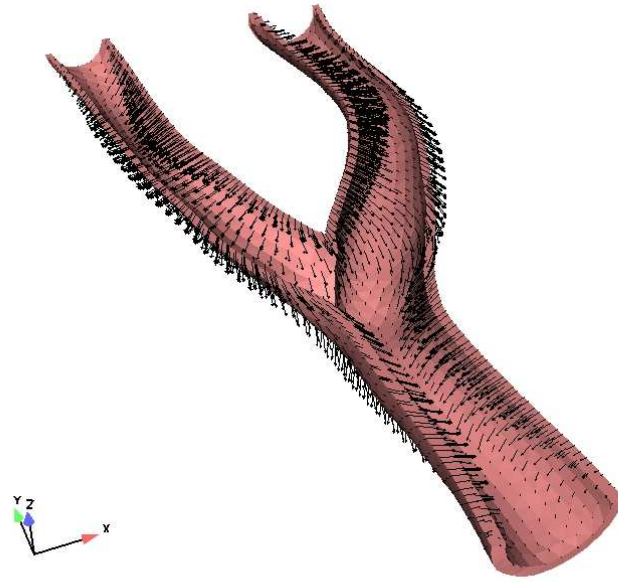


Figure 9: Structure deformation and velocity at time  $t = 0.015s$

- [11] P. L. Tallec, J. Mouro, Fluid structure interaction with large structural displacements, *Comput. Methods Appl. Mech. Engrg.* 190 (2001) 3039–3067.
- [12] P. Causin, J.-F. Gerbeau, F. Nobile, Added-mass effect in the design of partitioned algorithms for fluid-structure problems, *Tech. Rep. 5084*, INRIA (2004).
- [13] M. Cervera, R. Codina, M. Galindo, On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems, *Engrg. Comput.* 13 (6) (1996) 4–30.
- [14] S. Deparis, Numerical Analysis of Axisymmetric Flows and Methods for Fluid-Structure Interaction Arising in Blood Flow Simulation, Ph.D. thesis, École Polytechnique Fédérale de Lausanne (2004).
- [15] M. Fernández, M. Moubachir, A Newton method using exact jacobians for solving fluid-structure coupling, *Computers and Structures* 83 (2-3) (2005) 127–142.
- [16] M. Heil, An efficient solver for the fully coupled solution of large-

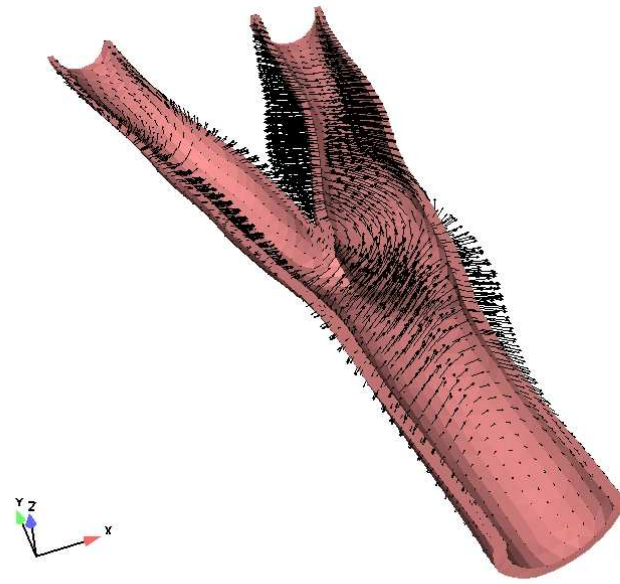


Figure 10: Structure deformation and velocity at time  $t = 0.020s$

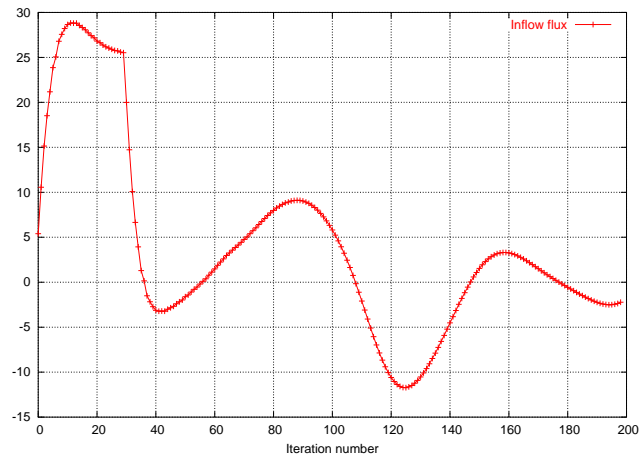


Figure 11: Inflow flux for the carotid geometry

displacement fluid-structure interaction problems, *Comput. Methods Appl. Mech. Engrg.* 193 (1-2) (2004) 1–23.

- [17] H. Matthies, J. Steindorf, Numerical efficiency of different partitioned methods for fluid-structure interaction, *Z. Angew. Math. Mech.* 2 (80) (2000) 557–558.
- [18] T. Tezduyar, Finite element methods for fluid dynamics with moving boundaries and interfaces, *Arch. Comput. Methods Engrg.* 8 (2001) 83–130.
- [19] S. Deparis, J. Gerbeau, X. Vasseur, GMRES preconditioning and accelerated quasi-Newton algorithm and application to fluid structure interaction, in preparation (2004).
- [20] J.-F. Gerbeau, V. Vidrascu, P. Frey, Fluid-structure interaction in blood flows on geometries coming from medical imaging, *Tech. Rep.* 5052, INRIA (2003).
- [21] J. Gerbeau, M. Vidrascu, A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows, *M2AN* 37 (4) (2003) 663–680.
- [22] J. Mouro, Interactions Fluide Structure en Grads Déplacements. Résolution Numérique et Application aux Composants Hydrauliques Automobiles, Ph.D. thesis, École Polytechnique, Paris (1996).
- [23] D. P. Mok, W. A. Wall, Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures, in: K. Schweizerhof, W. Wall (Eds.), *Trends in Computational Structural Mechanics*, K.U. Bletzinger, CIMNE, Barcelona, 2001.
- [24] A. Quarteroni, A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, Oxford, 1999.
- [25] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodríguez-Ferran, Arbitrary Lagrangian-Eulerian methods, in: *The Encyclopedia of Computational Mechanics*, Vol. I, Chapt. 14, Wiley, 2004, pp. 413–437.
- [26] T. Hughes, W. Liu, T. Zimmermann, Lagrangian-Eulerian finite element formulation formulation for incompressible flows, *Comp. Methods Appl. Mech. Engrg.* 29 (1981) 329–349.
- [27] J. Batina, Unsteady Euler airfoil solutions using unstructured dynamic meshes, *AAIA* 28 (1990) 1381–1388.
- [28] G. Fourestey, Simulation Numérique et Contrôle Optimal d’Interactions Fluide Incompressible/Structure par une Méthode de Lagrange-Galerkin d’Ordre 2. Applications aux Ouvrages d’Art, Ph.D. thesis, École Nationale des Ponts et Chaussées (2002).
- [29] M. Lesoinne, C. Farhat, Stability analysis of dynamic meshes for transient aeroelastic computations, AIAA, Proceedings of the 11th AIAA Computational Fluid Dynamics Conference, Orlando, Florida. Paper 93-3325.



- [30] P. G. Ciarlet, *Mathematical elasticity. Vol. II*, Vol. 27 of *Studies in Mathematics and its Applications*, North-Holland Publishing Co., Amsterdam, 1997, theory of plates.
- [31] D. Chapelle, K. Bathe, *The Finite Element Analysis of Shells - Fundamentals*, Springer, New York, 2003.
- [32] P. L. Tallec, *Introduction à la Dynamique des Structures*, Ellipse, Paris, 2000.
- [33] A. Quarteroni, M. Tuveri, A. Veneziani, Computational vascular fluid dynamics: problems, models and methods, *Comp. Vis. Science* 2 (2000) 163–197.
- [34] S. Deparis, M. Discacciati, A. Quarteroni, A domain decomposition framework for fluid-structure interaction problems, in: *Proceedings of the Third International Conference on Computational Fluid Dynamics (ICCFD3)*, Toronto, July 2004, 2004, submitted.
- [35] A. Quarteroni, R. Sacco, F. Saleri, *Numerical Mathematics*, Springer, New York-Berlin-Heidelberg, 2000.
- [36] A. Quarteroni, L. Formaggia, *Modelling of Living Systems*, *Handbook of Numerical Analysis*, Elsevier Science, Amsterdam, 2002, Ch. *Mathematical Modelling and Numerical Simulation of the Cardiovascular System*, submitted.
- [37] G. Karner, K. Perktold, M. Hofer, D. Liesch, Flow characteristics in an anatomically realistic compliant carotid artery bifurcation model, *Computer Methods in Biomechanics and Biomedical Engineering* (2) (1999) 171–185.