PARALLEL ALGORITHMS FOR FLUID-STRUCTURE INTERACTION PROBLEMS IN HAEMODYNAMICS

PAOLO CROSETTO *, SIMONE DEPARIS *, GILLES FOURESTEY *, AND ALFIO QUARTERONI *†

Abstract. The increasing computational load required by most applications and the limits in hardware performances affecting scientific computing contributed in the last decades to the development of parallel software and architectures. In Fluid-Structure Interaction (FSI, in short) for haemodynamic applications, parallelization and scalability are key issues (see [20]). In this work we introduce a class of parallel preconditioners for the FSI problem obtained by exploiting the block-structure of the linear system. We stress the possibility of extending the approach to a general linear system with a block-structure, then we provide a bound in the condition number of the preconditioned system in terms of the conditioning of the preconditioned diagonal blocks, finally we show that the construction and evaluation of the devised preconditioner is modular. The preconditioners are tested on a benchmark 3D geometry discretized in both a coarse and a fine mesh, as well as on two physiological aorta geometries. The simulations that we have performed show an advantage in using the block preconditioners introduced and confirm our theoretical results.

Key words. Blood-Flow Models , Fluid-Structure Interaction , Finite Elements , Preconditioners , Parallel Algorithms

AMS subject classifications. 65M60 , 65F08 , 65Y05 , 76Z05

1. Introduction . The modeling of the cardiovascular system is receiving increasing attention from both the medical and mathematical environments because of, from the one hand, the great influence of haemodynamics on cardiovascular diseases ([20] chap 1), and, from the other hand, its challenging complexity that keeps open the debate about the setting up of appropriate models and algorithms. A wide variety of approaches can be found in literature, dealing with different formulations of the problem and solution strategies.

In this introduction we refrain from describing the models that can be used to simulate the physiological behavior of the arterial vessels; for that we address the interested reader to [20]. We give instead an overview of some of the most popular methodologies to solve numerically the coupled system of equations arising from the haemodynamic model: those that describe the flow-field variables (blood velocity and pressure) and those that govern the mechanical deformation of the vessel walls (the "structure"). The first distinction comes from the formulation of the problem.

A common choice in the FSI context is to describe the fluid equations using an Arbitrary Lagrangian-Eulerian frame of reference (see e.g. [31]). The advantage with respect to an Eulerian description is that the coupling can be satisfied exactly on the fluid-structure interface. However the introduction of a new equation for the fluid domain motion is required, and its dependence on the solution of the FSI problem introduces a further nonlinearity.

A different approach consists of a space-time formulation which adopts the Eulerian framework. Usually, the latter involves a discretization of the computational domain in *time slabs*, and each solution in a time slab is computed sequentially (see [37, 25], or [7] for a description of the formulation). Other approaches are based on a standard Eulerian formulation with a method to keep track of the fluid-structure interface [11, 39]. With the latter approach the computation of the fluid domain displacement is avoided, however the coupling conditions cannot be imposed exactly on the interface.

Once the system of equations describing the physical problem is set up, a further optional step consists of splitting the global system into *subdomain problems*, i.e. the domain is split into a fluid and a solid problem leading to standard domain decomposition (DD) approaches. In this DD-like context, Dirichlet-Neumann schemes [29, 30, 18], are the most popular ones adopted in FSI. Robin-Neumann and Robin-Robin schemes are applied in [3] to the FSI context, while other standard domain decomposition strategies (e.g. Neumann-Neumann, FETI) are described in [38]. Another

*IACS-Chair of Modelling and Scientific Computing (CMCS), EPFL, CH-1015 Lausanne, Switzerland (paolo.crosetto@epfl.ch, simone.deparis@epfl.ch, gilles.fourestey@epfl.ch, alfio.quarteroni@epfl.ch)

†MOX-Dipartimento di Matematica "F. Brioschi", Politecnico di Milano, I-20133 Milano, Italy

similar option consists of reformulating the problem on the fluid-structure interface through the Steklov-Poincaré operators, see e.g. [14].

All these strategies correspond to a particular choice of the subdomains and of the interface conditions assigned in the course of the subdomain iterations. Following the definitions given in [9] all these reformulations of the problem can be qualified as *nonlinear preconditioners*. These domain decomposition schemes are particularly suited to the case when separate (and independent) solvers for the subdomain problems are available, because the solution of the global system can be obtained through repeated solutions of the subdomain problems (this property is often referred to as *modularity*).

The choice of the time discretization introduces further distinctions among the methods. The fully coupled nonlinear problem can be discretized in time by considering all the terms in the equations implicitly, that leads to a *fully implicit* method [7, 37, 24, 27, 6, 15]. This is the most stable but also most expensive choice. A large variety of alternative time discretizations can be devised. E.g. a *Geometry-Convective Explicit* discretization can be found in [4], where the moving geometry is taken at the previous time step and the convective term is treated partly explicitly (see Section §3.2 for details). Even in the space-time framework the fluid domain in a time slab can be extrapolated using the informations relative to previous time slabs, e.g. [37, 25]. Other choices concern the way in which the coupling conditions (in a DD-like scheme) are imposed. We can devise three main classes regarding the coupling strategies:

- Strongly (implicitly) coupled schemes. An extra loop enforces exactly the coupling condition. As a result they can require a variable amount of *outer iterations*, depending on the algorithm used and on physical parameters [29, 30, 3, 4].
- Fractional step (semi-implicit coupling) schemes[1]. They involve a splitting of the system in two (or several) problems, a solution of both and a successive correction of the fluid velocity. These schemes in general require a time step restriction to reach convergence [19, 1, 34].
- Weakly (explicitly) coupled, or staggered, schemes. These methods are very cheap since they require just one solution of each subproblem per time step. In some cases they are proved to be unconditionally unstable [10], however recent studies show that this instability can be overcome, at the expense of introducing suitable dissipation terms [8].

A natural way to handle the nonlinearity is based on the use of the Aitken accelerated fixed point algorithm in all its variants [29, 21, 4]. Each fixed point iteration requires one residual evaluation.

Otherwise the time discretized problem can be linearized via Newton's method, either exact, as in [7, 24, 18, 37, 27], or inexact, as in [6, 21, 12, 23, 13]. In the Newton/quasi-Newton approaches the Jacobian matrix is often available only as matrix-vector multiplication (it is the case in [18]). In these cases a matrix free method must be employed to solve exactly the Jacobian system. Each iteration of this method requires a solve of the linearized subproblems. Thus the cost of each nonlinear iteration is the cost of one residual evaluation plus a variable number of solutions to the linearized subproblems.

For what concerns the fully coupled discretized equations where no domain decomposition were employed, the key aspect that characterizes the different methodologies is the choice of the preconditioner. In fact by choosing block preconditioners such as block Jacobi or block Gauss-Seidel, the preconditioned system can be solved in a modular fashion. These strategies are the algebraic version of the domain decomposition algorithms cited above. Approximating the Schur complements in a block LU factorization is a strategy tested in [5] for a fractional step scheme; this method corresponds to a different algebraic splitting of the FSI linear system.

A similar strategy adopted in [33] in a different context uses the approximation of the block LU factorization as a preconditioner for GMRES within a strongly coupled scheme, a choice justified by the analysis of the condition number of the preconditioned system carried out in [2].

In the FSI literature, for the sake of classification the term *monolithic* scheme is sometimes

[1]The splitting that occurs in these methods usually is not obtained through a domain decomposition because it couples the equations for the solid displacement and for the fluid pressure (see [1, 34])

Time Discr.	System Formulation		Solution Algorithm		Preconditioner
FI/CE	Newton		GMRES/direct		P_{AS}, P_{GS}, $P_{AS}(P_{GS})$ ...
	inexact Newton		GMRES/direct		P_{AS}, P_{GS}, $P_{AS}(P_{GS})$ ...
	DD	Newton	out. GMRES/Rich.	inn. GMRES/dir.	P_{sub}
	DD	inexact N.	out. GMRES/Rich.	inn. GMRES/dir.	P_{sub}
	DD	Fixed Point		inn. GMRES/dir.	P_{sub}
GCE	Linear System		GMRES/direct		P_{AS}, P_{GS}, $P_{AS}(P_{GS})$ ...
			out. GMRES/Rich.	inn. GMRES/dir.	P_{sub}

TABLE 1.1

Methodologies for the solution of fluid-structure interaction problems ("Rich" stands for Richardson, "dir" for direct, " out" and "inn" for outer, respectively inner, iterations).

used with different meanings: either to describe a situation in which the global system is solved *in one go* [29], or *as a whole* [24], or when all the equations are solved *simultaneously in fully coupled fashion* [7]. Other authors define the monolithic approach as a method that requires the *development of new software and new solution methods for each of these coupled applications*, in opposition to the partitioned approach, where *the methods and software systems which have been developed for either application will continue to be used* [30].

As we did not find in literature an unambiguous definition that allows to univocally define the methods described above we prefer to refrain from using in the following the notations *monolithic* and *partitioned*.

A picture representing some of the methodologies listed above is given by Table 1.1. With P_{AS} we denote the algebraic additive Schwarz preconditioners, that will be used throughout this paper, while P_{GS} represents a generic block Gauss-Seidel preconditioner. The preconditioner P_{sub} refers to the linear systems on the subdomains (or sub-blocks in the algebraic case).

The last line corresponds to the splittings performed at algebraic level. We remark that the same kind of splitting can be performed also on the (approximated) Jacobian system, we did not include these schemes in the picture because, at the best of our knowledge, there is no relevant literature dealing with them.

This paper is organized as follows. In Section §2 we describe the model that we have chosen for the solution of the FSI system and compact notations are devised to represent the different problems at hands. Section §3 describes in detail some of the solution approaches cited above and the different time discretizations and Newton variants that will be used throughout our simulations. In Section §4 e claass of block triangular preconditioners is described, that can be represented as the product of two matrices, and can be computed in a parallel and scalable fashion. Furthermore an analysis of these preconditioners is carried out, and a bound on the condition number of the preconditioned system is derived. Then follows a synthetic description of the algebraic additive Schwarz preconditioning strategy. Sections §5 and §6 describe in detail the GCE and CE time discretizations, respectively showing the form of the preconditioners proposed in both cases. Finally, Section §7 deals with the numerical results obtained with the finite element library LifeV, where the computations confirm the good spectral qualities of the class of preconditioners considered, and numerical tests on different meshes varying the physical parameters are performed.

2. Physical Model. The domain under consideration is composed by a fluid and a compliant structure (see Figure 2.1). The model is described by the coupling of Navier–Stokes equations for the fluid and of linear St. Venant Kirchhoff equations for the structure [20].

We write the Navier–Stokes equations in an ALE framwork [31]; this allows to arbitrarily set the fluid computational domain boundaries at the fluid-structure interface, as well as to fix the inflow and outflow sections.

As mentioned in Section §1 this choice yields a further equation for the fluid domain displacement. A popular choice, adopted in the present paper, is to compute the displacement $\mathbf{d}_\mathrm{f}$ of the fluid domain $\Omega_t^\mathrm{f} \subset \mathbb{R}^3$ as an harmonic extension of the interface displacement $\mathbf{d}_\mathrm{s}|_{\Gamma_o}$, where Γ_o is

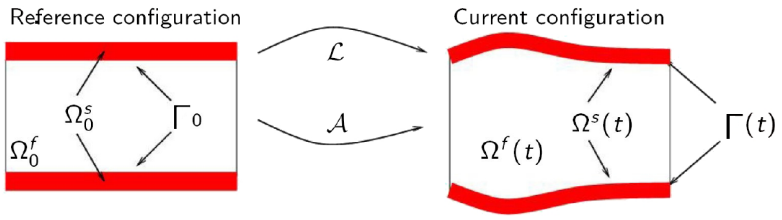

FIG. 2.1. *Reference and current configuration with Lagrangian and ALE mappings.*

the interface between the fluid and the solid in the initial configuration:

$$\begin{cases} -\Delta \mathbf{d}_{\mathrm{f}} &= 0 & \text{in } \Omega_o^{\mathrm{f}} \\ \mathbf{d}_{\mathrm{f}} &= \mathbf{d}_{\mathrm{s}}|_{\Gamma_o} & \text{on } \Gamma_o, \end{cases} \tag{2.1}$$

where $\Omega_o^{\mathrm{f}} \subset \mathbb{R}^3$ is the fluid reference domain. The ALE mapping is then defined as

$$\begin{aligned} \mathcal{A}_t : \Omega_o^{\mathrm{f}} &\to \Omega_t^{\mathrm{f}} \\ \mathbf{x}_o &\mapsto \mathcal{A}_t(\mathbf{x}_o) = \mathbf{x}_o + \mathbf{d}_{\mathrm{f}}(\mathbf{x}_o). \end{aligned}$$

We should ensure that for each time t the mapping $\mathcal{A}_t$ is a diffeomorphism. This is in general the case for blood flow simulations [31]; at the discrete level, our code checks that the measure of the finite element tetrahedra remains positive at each time step.

Let

$$\mathbf{w} = \left.\frac{\partial \mathcal{A}_t}{\partial t}\right|_{\mathbf{x}o} = \left.\frac{\partial \mathbf{d}_{\mathrm{f}}}{\partial t}\right|_{\mathbf{x}o} \tag{2.2}$$

be the fluid domain velocity. The Navier-Stokes equations in the ALE form read

$$\begin{cases} \rho_{\mathrm{f}} \left.\dfrac{\partial \mathbf{u}_{\mathrm{f}}}{\partial t}\right|_{\mathbf{x}o} + (\rho_{\mathrm{f}}(\mathbf{u}_{\mathrm{f}} - \mathbf{w}) \cdot \nabla)\mathbf{u}_{\mathrm{f}} - \nabla \cdot \sigma_{\mathrm{f}} = 0 & \text{in } \Omega_t^{\mathrm{f}} \\ \nabla \cdot \mathbf{u}_{\mathrm{f}} = 0 & \text{in } \Omega_t^{\mathrm{f}}, \end{cases} \tag{2.3}$$

where $\sigma_{\mathrm{f}} = -pI + \mu_{\mathrm{f}} \frac{(\nabla \mathbf{u}_{\mathrm{f}} + (\nabla \mathbf{u}_{\mathrm{f}})^T)}{2}$ is the Cauchy stress tensor. The boundary conditions associated to these equations can be subdivided in two kinds: inflow/outflow boundary conditions, to be imposed at the artificial boundaries, possibly by lower dimensional models, and fluid-structure interaction coupling conditions, which we describe later on.

For simplicity in this work we consider a linear isotropic St. Venant-Kirchoff model to describe the solid displacement; the extension of our algorithms to a nonlinear structure dynamic is straightforward. The associated equation reads

$$\rho_{\mathrm{s}} \frac{\partial^2 \mathbf{d}_{\mathrm{s}}}{\partial t^2} - \nabla \cdot \sigma_{\mathrm{s}}^o = 0 \text{ in } \Omega_o^{\mathrm{s}}, \tag{2.4}$$

where $\sigma_{\mathrm{s}}^o = \lambda \mathrm{tr}(\epsilon) + 2\mu_{\mathrm{s}}\epsilon$, with $\epsilon = \frac{(\nabla \mathbf{d}_{\mathrm{s}} + (\nabla \mathbf{d}_{\mathrm{s}})^T)}{2}$, is the Piola stress tensor and Ω_o^{s} is the solid domain in the reference configuration. The coupling between the fluid and the structure is provided by three conditions:

- the continuity of the velocity at the FS interface

$$\mathbf{u}_{\mathrm{f}} \circ \mathcal{A}_t = \frac{d\mathbf{d}_{\mathrm{s}}}{dt} \text{ on } \Gamma_o; \tag{2.5}$$

- the continuity of stresses at the FS interface

$$\sigma_{\mathrm{s}}^o \mathbf{n}^o = J_{\mathrm{s}} \sigma_{\mathrm{f}} F_{\mathrm{s}}^{-T} \mathbf{n}^o \quad \text{on } \Gamma_o, \tag{2.6}$$

  where $\mathbf{n}^o$ is the outward normal to the solid (or fluid) reference domain, $F_{\mathrm{s}} = I + \nabla \mathbf{d}_{\mathrm{s}}$ is the solid deformation gradient and J_{s} its determinant;

- the geometric adherence already introduced in equation (2.1)

$$\mathbf{d}_\mathrm{f} = \mathbf{d}_\mathrm{s} \text{ on } \Gamma_o. \tag{2.7}$$

2.1. Notations. The derivation of the weak formulation of the coupled problem is standard and will be omitted for the sake of brevity (see [14]). From now on, however, the equations defining the coupled problem will be intended in the sense of distributions. To express, at a fixed time t, the equations describing the coupled problem in a compact form, we introduce the operators F, S, and G. The *fluid problem*

$$\mathsf{F}(\mathbf{u}, \mathbf{d}_\mathrm{s}, \mathbf{d}_\mathrm{f}) = 0 \tag{2.8}$$

represents the fluid momentum and continuity equations (2.3) with interface boundary conditions given by (2.5); the unknowns are the fluid velocity and pressure, grouped in the vector variable $\mathbf{u}$. The fluid problem is coupled with both the geometry problem, through the dependence on the fluid domain displacement $\mathbf{d}_\mathrm{f}$ and on the associated ALE velocity $\mathbf{w}$, see (2.2), and to the structure dynamics, through the boundary condition (2.7) on the fluid-structure interface Γ_o, depending on $\mathbf{d}_\mathrm{s}$. The *structure or solid problem*

$$\mathsf{S}(\mathbf{u}, \mathbf{d}_\mathrm{s}) = 0 \tag{2.9a}$$

represents the solid equation (2.4) with interface boundary condition given by (2.6); the unknown is the solid displacement $\mathbf{d}_\mathrm{s}$, the problem is coupled with the fluid quantities $\mathbf{u}$ through the boundary condition (2.6) on Γ_o.

To be more precise the interface condition (2.6) for the solid problem is expressed in a weak form by equating the variational residuals of the two momentum equations (2.3) and (2.4) tested against test functions that match at the fluid-structure interface (this is called weak stress continuity, see e.g. [31]). Thus, as the boundary condition depends on an integral on the fluid interface, the solid problem formally depends also on the fluid domain displacement $\mathbf{d}_\mathrm{f}$. For this reason in the following, instead of (2.9a) we write the solid problem as

$$\mathsf{S}(\mathbf{u}, \mathbf{d}_\mathrm{s}, \mathbf{d}_\mathrm{f}) = 0, \tag{2.9b}$$

keeping in mind that the dependence on $\mathbf{d}_\mathrm{f}$ is implicitly taken into account by the residuals.

The *geometry problem*

$$\mathsf{G}(\mathbf{d}_\mathrm{s}, \mathbf{d}_\mathrm{f}) = 0 \tag{2.10}$$

represents the harmonic extension problem (2.1), with unknown $\mathbf{d}_\mathrm{f}$. The latter is coupled with $\mathbf{d}_\mathrm{s}$ through the boundary condition on Γ_o.

3. Solution Approach . An important distinction to be made when dealing with multi-physics problems is between modular and non modular approaches. As already pointed out in the Introduction the formers allow the recycling of existing solvers for the different subproblems coupled in the model, while the latter requires the implementation of an ad hoc solver. A non-modular solver cannot be used when there is no access to the subproblem matrices (i.e., when the subproblems solvers are handled as black boxes).

3.1. Modular versus Non-Modular Algorithms. The most popular modular algorithms to solve FSI problems for haemodynamics are those of Dirichlet-Neumann type. They consist in solving the coupled system at every time step using sequential solves of the geometry, fluid, and solid problems G, F, and S described above, and then iterate if necessary.

In particular the solution of the fluid problem is obtained by imposing the continuity of the velocity at the interface (2.5) while the solution of the solid problem is obtained by imposing a Neumann boundary condition (2.6) on the interface. The interface velocity used in the Dirichlet step can be computed for example by the finite difference $\frac{\mathbf{d}_\mathrm{s}^n - \mathbf{d}_\mathrm{s}^{n-1}}{\delta t}$, while the Neumann boundary

condition is imposed as the equality between the variational residuals of the momentum conservation equations (2.3) and (2.4). One of the drawbacks of this approach is that by imposing Dirichlet boundary at the fluid inlet and outlet, because of the Dirichlet interface conditions prescribed on the fluid problem, the mass balance is not necessarily satisfied in the fluid domain. This leads to an inconsistent fluid problem [28].

In contrast, a non-modular algorithm implements a solver for the coupled nonlinear system $(\mathsf{F}-\mathsf{S}-\mathsf{G})$. One advantage is that the subiterations of local solvers can be avoided. In this case an important role is played by the preconditioner used to solve the linear system that is obtained after discretization and linearization of the problem. Among the different choices described in literature, the domain decomposition preconditioners like the additive Schwarz preconditioner show to be effective [6].

Recent studies [4, 24, 27] have tested and compared an implementation of both the modular and non-modular methods, showing advantages in using the latter when the source codes are available, for every tested mesh and range of physical parameters.

3.2. Nonlinearities. The system of equations describing the FSI problem is highly nonlinear. Indeed, the nonlinearities are given by the Navier-Stokes convective term and by the displacement of the domain of the fluid problem (and possibly by the structure).

At any given time level the nonlinear system of equations reads

$$\begin{cases} \mathsf{F}(\mathbf{u},\mathbf{d}_\mathrm{s},\mathbf{d}_\mathrm{f})=0 \\ \mathsf{S}(\mathbf{u},\mathbf{d}_\mathrm{s},\mathbf{d}_\mathrm{f})=0 \\ \mathsf{G}(\mathbf{d}_\mathrm{s},\mathbf{d}_\mathrm{f})=0. \end{cases} \tag{3.1}$$

Notice that, for the sake of simplicity, we have omitted to indicate the dependence on h and δt of the discrete solution $(\mathbf{u},\mathbf{d}_\mathrm{s},\mathbf{d}_\mathrm{f})$, and, more importantly, of the functional laws F, S and G (the former might account for the presence of stabilization terms). We classify some of the possible time discretizations as follows:

- Fully Implicit (FI) time discretization. All the terms of the equations are considered implicitly, in particular the convective term of the fluid momentum equation reads $(\mathbf{u}_\mathrm{f}-\mathbf{w})\nabla\mathbf{u}_\mathrm{f}$.
- Convective Explicit (CE) time discretization. The convective term in the fluid momentum equation is approximated by $(\mathbf{u}_\mathrm{f}^*-\mathbf{w}^*)\nabla\mathbf{u}_\mathrm{f}$, where * denotes a convenient extrapolation, e.g., from the previous time step.
- Geometry-Convective Explicit (GCE) time discretization. The convective term is treated as in the Convective Explicit discretization and the domain Ω_t^f is extrapolated. In this case neither the fluid nor the solid problems depend on $\mathbf{d}_\mathrm{f}$. The problem is linear and the geometry problem G can be solved in a successive step separated from the fluid and structure problems. If $\mathbf{d}_\mathrm{f}^*$ denotes an extrapolation of the fluid domain displacement, problem (3.1) splits into the two subproblems

$$\begin{cases} \mathsf{F}(\mathbf{u},\mathbf{d}_\mathrm{s},\mathbf{d}_\mathrm{f}^*)=0 \\ \mathsf{S}(\mathbf{u},\mathbf{d}_\mathrm{s},\mathbf{d}_\mathrm{f}^*)=0 \end{cases} \tag{3.2}$$

and

$$\mathsf{G}(\mathbf{d}_\mathrm{s},\mathbf{d}_\mathrm{f})=0. \tag{3.3}$$

As previously noticed, in both FI and CE cases we end up with a nonlinear problem. Its linearization will be carried out by the Newton method, whose correction step on a general problem like (3.1) reads as follows: For any given $\mathbf{u}^k$, $\mathbf{d}_\mathrm{s}^k$ and $\mathbf{d}_\mathrm{f}^k$, find $\delta\mathbf{u}^k,\delta\mathbf{d}_\mathrm{s}^k,\delta\mathbf{d}_\mathrm{f}^k$ such that

$$\left.\begin{pmatrix} \frac{\partial\mathsf{F}}{\partial\mathbf{u}} & \frac{\partial\mathsf{F}}{\partial\mathbf{d}_\mathrm{s}} & \frac{\partial\mathsf{F}}{\partial\mathbf{d}_\mathrm{f}} \\ \frac{\partial\mathsf{S}}{\partial\mathbf{u}} & \frac{\partial\mathsf{S}}{\partial\mathbf{d}_\mathrm{s}} & \frac{\partial\mathsf{S}}{\partial\mathbf{d}_\mathrm{f}} \\ 0 & \frac{\partial\mathsf{G}}{\partial\mathbf{d}_\mathrm{s}} & \frac{\partial\mathsf{G}}{\partial\mathbf{d}_\mathrm{f}} \end{pmatrix}\right|_{(\mathbf{u}^k,\mathbf{d}_\mathrm{s}^k,\mathbf{d}_\mathrm{f}^k)} \begin{pmatrix} \delta\mathbf{u}^k \\ \delta\mathbf{d}_\mathrm{s}^k \\ \delta\mathbf{d}_\mathrm{f}^k \end{pmatrix} = -\begin{pmatrix} \mathsf{F}(\mathbf{u}^k,\mathbf{d}_\mathrm{s}^k,\mathbf{d}_\mathrm{f}^k) \\ \mathsf{S}(\mathbf{u}^k,\mathbf{d}_\mathrm{s}^k,\mathbf{d}_\mathrm{f}^k) \\ \mathsf{G}(\mathbf{d}_\mathrm{s}^k,\mathbf{d}_\mathrm{f}^k) \end{pmatrix}. \tag{3.4}$$

The matrix on the left is the Jacobian of the non-linear system (3.1). It will be noted J_{FI} or J_{CE} depending on whether the fully implicit or the convective-explicit case is considered (notice that the problem is linear in the GCE case). The cross derivatives of the S and F operators with respect to $\mathbf{d}_\mathrm{f}$ (*shape derivatives*) have a non-trivial expression which is calculated analytically in [18].

The solution of the FI and CE systems can be obtained by one of the following algorithms:

- the *(exact) Newton* algorithm; the shape derivatives are computed exactly;
- the *quasi-Newton* algorithm; the shape derivatives in (3.4) are approximated.

In the literature [37, 23] there are examples of approximation of the shape derivatives terms by finite differences, i.e., for the fluid problem,

$$D_{\mathbf{d}_\mathrm{f}}\mathsf{F}(\mathbf{u}_\mathrm{f}^k,\mathbf{d}_\mathrm{s}^k,\mathbf{d}_\mathrm{f}^k)\delta\mathbf{u}_\mathrm{f} \approx \frac{\mathsf{F}(\mathbf{u}_\mathrm{f}^k,\mathbf{d}_\mathrm{s}^k,\mathbf{d}_\mathrm{f}^k+\epsilon\delta\mathbf{d}_\mathrm{f})-\mathsf{F}(\mathbf{u}_\mathrm{f}^k,\mathbf{d}_\mathrm{s}^k,\mathbf{d}_\mathrm{f}^k)}{\epsilon}. \tag{3.5}$$

However this approach requires extra assembling of the fluid problem at each Newton step [27] and is computationally expensive. It is also possible to neglect the shape derivatives, in which case the Jacobian matrix is inherited by the computation of the residual in (3.1), we call J_{QN} the matrix resulting from the approximated Jacobian; this is simpler to implement and computationally less expensive, although in some circumstances it fails to converge [18] or the convergence rate is very low [23]. In [6] this method has been adopted, while in [22] a linear fluid model on a fixed domain is used to compute an approximation of the Jacobian matrix, leading to a very cheap Jacobian and "large" number of quasi-Newton iterations.

As in this work we focus on the geometric non-linearity we consider the CE and GCE time discretizations rather than the FI one, that would require the insertion of additional terms in the Jacobian matrix. However the extension to a FI case and a nonlinear structure model requires a small effort and will not be addressed here.

4. Preconditioning. In this section we discuss some different preconditioning strategies that can be applied on either the GCE system matrix (3.2), or the Jacobian matrix J_{CE} of the CE time discretization.

4.1. Block Preconditioners and Spectral Analysis. Let us address a general framework of a 4-blocks matrix

$$A = \begin{pmatrix} A_1 & D \\ B & A_2 \end{pmatrix}, \tag{4.1}$$

where the block lines correspond to different coupled problems, the coupling being expressed by the matrices B and D. The meaning of the different blocks in the CE and GCE contexts will become clear in the next sections. The only a-priori assumption on the matrices A_1 and A_2 is that they are invertible, therefore the following analysis holds true for any nonsingular linear system whose matrix has a 4-blocks structure like (4.1).

A possible preconditioning approach comes from the approximation of the block LU factorization

$$A = \begin{pmatrix} A_1 & D \\ B & A_2 \end{pmatrix} = \begin{pmatrix} I & 0 \\ BA_1^{-1} & I \end{pmatrix} \begin{pmatrix} A_1 & D \\ 0 & A_2 - BA_1^{-1}D \end{pmatrix}. \tag{4.2}$$

The preconditioner deriving from this expression is usually obtained by approximating the Schur complement $S = A_2 - BA_1^{-1}D$ (e.g. [5] for the FSI problem, [33] for the bidomain equations, [16, 17] for Navier–Stokes equations).

Another common approach is to use a block Gauss-Seidel preconditioner which consists in neglecting part of the coupling

$$P_{GS} = \begin{pmatrix} A_1 & 0 \\ B & A_2 \end{pmatrix} = \begin{pmatrix} I & 0 \\ BA_1^{-1} & I \end{pmatrix} \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}. \tag{4.3}$$

The left-preconditioned system thus reads

$$P_{GS}^{-1}A = \begin{pmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{pmatrix} \begin{pmatrix} A_1 & D \\ 0 & S \end{pmatrix}. \tag{4.4}$$

We denote the condition number of the preconditioned matrix (4.4) by

$$\delta_{GS} = K(P_{GS}^{-1}A).$$

The behavior of these preconditioners in a serial environment have been studied for instance in [23] for the FSI system, where in the computational results the preconditioner shows to be robust.

We now replace the diagonal blocks A_1 and A_2 of the preconditioner P_{GS} (4.3) by suitable preconditioners P_1 and P_2 In the following we estimate the influence of this approximation on the conditioning of the system. To this end we suppose that we have an estimate for the condition number of the two preconditioned blocks. Calling $\sigma_A^{\max} = \max_\sigma(\sigma(A))$ and $\sigma_A^{\min} = \min_\sigma(\sigma(A))$ respectively the maximum and the minimum singular values for the matrix A, then for $i \in \{1, 2\}$

$$\delta_i = K(P_i^{-1}A_i) = \sigma^{\max}_{P_i^{-1}A_i} / \sigma^{\min}_{P_i^{-1}A_i}.$$

We introduce a modified preconditioner of the form

$$P = \begin{pmatrix} P_1 & 0 \\ B & \alpha P_2 \end{pmatrix}, \tag{4.5}$$

where α is an arbitrary positive scalar. In order to bound the condition number of the matrix $P^{-1}A$ we compute explicitly P^{-1} as follows

$$P^{-1} = \begin{pmatrix} P_1^{-1} & 0 \\ -\frac{1}{\alpha}P_2^{-1}BP_1^{-1} & \frac{1}{\alpha}P_2^{-1} \end{pmatrix}.$$

Then the preconditioned matrix can be factorized as

$$P^{-1}A = \begin{pmatrix} I & 0 \\ \Sigma & I \end{pmatrix} \begin{pmatrix} P_1^{-1} & 0 \\ 0 & \frac{1}{\alpha}P_2^{-1} \end{pmatrix} \begin{pmatrix} A_1 & D \\ 0 & S \end{pmatrix}, \tag{4.6}$$

where $\Sigma = \frac{1}{\alpha}P_2^{-1}B(A_1^{-1}P_1 - I)$.

An upper bound of the condition number for the preconditioned system can be obtained using the inequality $K(AB) \leqslant K(A)K(B)$. We first find the singular values of the block lower triangular factor (that we note L).

PROPOSITION 4.1. *The maximum and minimum singular values of L satisfy*

$$\sigma_L^{\max} = \sqrt{1 + \frac{(\sigma_\Sigma^{\max})^2 + \sqrt{(\sigma_\Sigma^{\max})^4 + 4(\sigma_\Sigma^{\max})^2}}{2}}, \tag{4.7}$$

and

$$\sigma_L^{\min} = \sqrt{1 + \frac{(\sigma_\Sigma^{\max})^2 - \sqrt{(\sigma_\Sigma^{\max})^4 + 4(\sigma_\Sigma^{\max})^2}}{2}}. \tag{4.8}$$

Proof. In fact by the definition of the singular values we have

$$\sigma_L^{\max} = \sqrt{\max(|\operatorname{eigs}(L^T L)|)},$$

where

$$L^T L = I + \begin{pmatrix} \Sigma^T\Sigma & \Sigma^T \\ \Sigma & 0 \end{pmatrix}.$$

Let us consider the matrix on the right: its null eigenvalues correspond to unitary eigenvalues of L. Its nonzero eigenvalues can be written in function of the eigenvalues of $\Sigma^T\Sigma$. As a matter of fact, from the definition of eigenvalue,

$$\begin{pmatrix} \Sigma^T\Sigma & \Sigma^T \\ \Sigma & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \lambda \begin{pmatrix} v_1 \\ v_2 \end{pmatrix},$$

by formally substituting v_2, we obtain the following relation

$$\left(1 + \frac{1}{\lambda}\right) \Sigma^T\Sigma v_1 = \lambda v_1$$

that gives

$$\lambda = \frac{\lambda_{\Sigma^T\Sigma} \pm \sqrt{\lambda^2_{(\Sigma^T\Sigma)} + 4\lambda_{(\Sigma^T\Sigma)}}}{2} = \frac{(\sigma_\Sigma)^2 \pm \sqrt{(\sigma_\Sigma)^4 + 4(\sigma_\Sigma)^2}}{2}.$$

Eventually we get

$$\sigma_L = \sqrt{1 + \frac{(\sigma_\Sigma)^2 \pm \sqrt{(\sigma_\Sigma)^4 + 4(\sigma_\Sigma)^2}}{2}}.$$

The functions $f_\pm(x) = 2 + x \pm \sqrt{x^2 + 4x}$ are positive for $x \geqslant 0$, f_+ is increasing and f_- is decreasing; moreover $f_+(x) > f_-(y)$ for all x, y in $\mathbb{R}^+$. This implies the two identities (4.7) and (4.8).

□

After some standard algebra we can rewrite the condition number of the L factor as

$$K(L) = \frac{\sigma_L^{\max}}{\sigma_L^{\min}} = \frac{2 + (\sigma_\Sigma^{\max})^2 + \sqrt{(\sigma_\Sigma^{\max})^4 + 4(\sigma_\Sigma^{\max})^2}}{2} \equiv \kappa(\sigma_\Sigma^{\max}),$$

which means that the condition number of L only depends on the maximum singular value of $\Sigma = \frac{1}{\alpha} P_2^{-1} B (A_1^{-1} P_1 - I)$. Note that good conditioning of L strongly depends on how well $A_1^{-1} P_1$ approaches the identity.

We now rewrite the block diagonal and upper triangular factors in (4.6) as

$$\begin{pmatrix} P_1^{-1}A_1 & 0 \\ 0 & \frac{1}{\alpha}P_2^{-1}A_2 \end{pmatrix} \begin{pmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{pmatrix} \begin{pmatrix} A_1 & D \\ 0 & S \end{pmatrix} = \begin{pmatrix} P_1^{-1}A_1 & 0 \\ 0 & \frac{1}{\alpha}P_2^{-1}A_2 \end{pmatrix} P_{GS}^{-1} A,$$

whose condition number is bounded by

$$\frac{\max\{\sigma^{\max}_{P_1^{-1}A_1}, \frac{1}{\alpha}\sigma^{\max}_{P_2^{-1}A_2}\}}{\min\{\sigma^{\min}_{P_1^{-1}A_1}, \frac{1}{\alpha}\sigma^{\min}_{P_2^{-1}A_2}\}} \delta_{GS}. \tag{4.9}$$

As a result we obtain a bound for the condition number of the preconditioned linear system

$$K(P^{-1}A) \leqslant \kappa(\sigma_\Sigma^{\max}) \frac{\max\{\sigma^{\max}_{P_1^{-1}A_1}, \frac{1}{\alpha}\sigma^{\max}_{P_2^{-1}A_2}\}}{\min\{\sigma^{\min}_{P_1^{-1}A_1}, \frac{1}{\alpha}\sigma^{\min}_{P_2^{-1}A_2}\}} \delta_{GS}. \tag{4.10a}$$

If the maximum singular values of the preconditioned sub-blocks ($\sigma^{\max}_{P_1^{-1}A_1}$ and $\sigma^{\max}_{P_2^{-1}A_2}$) are available, we can improve this estimate by appropriately defining the scalar α. In particular, if $\alpha = \sigma^{\max}_{P_2^{-1}A_2} / \sigma^{\max}_{P_1^{-1}A_1}$ then (4.9) simplifies to $\max\{\delta_1, \delta_2\}\delta_{GS}$. Note that the maximum singular value $\kappa(\sigma_\Sigma^{\max})$ of $\Sigma = \frac{1}{\alpha} P_2^{-1} B (A_1^{-1} P_1 - I)$ depends on α; therefore the rescaling is useful if either $\alpha > 1$ (i.e., $\sigma^{\max}_{P_2^{-1}A_2} > \sigma^{\max}_{P_1^{-1}A_1}$), or if the gain in the bounding factor (4.9) justifies the loss in $\kappa(\sigma_\Sigma^{\max})$.

The rescaling α obviously does not affect the quantities δ_1 and δ_2, thus the final condition number estimate reads

$$K(P^{-1}A) \leqslant \kappa(\sigma_\Sigma^{\max}) \max\{\delta_1, \delta_2\}\delta_{GS}. \tag{4.10b}$$

Estimate (4.10b) shows that we have to choose the preconditioners for the diagonal blocks according to the following criteria:

- P_1 and P_2 are well suited preconditioners for the matrices A_1, A_2;
- $A_1^{-1}P_1$ is near the identity, such that $\kappa(\sigma_\Sigma^{\max})$ is small;

and then define α as an approximate ratio of the maximum singular values of the preconditioned sub-systems. The same estimate also suggests that in some circumstances the roles of A_1 and A_2 should be interchanged.

Thanks to this analysis, we are able to build parallel preconditioners of the coupled problem based only on the subproblems. Of utmost importance, the scalability of our problem depends on the coupling only through σ_Σ. Thus if we find a block GS preconditioner for which σ_Σ is bounded the scalability will not depend on the coupling, but only on the scalability properties of the sub-problems.

4.2. Algebraic Additive Schwarz Preconditioners. Our sub blocks preconditioners are based on the algebraic additive Schwarz preconditioner (AAS) that we briefly overview here, referring to [35, 38] for a detailed description and analysis of the Schwarz preconditioning methodology. We consider the generic linear system $A\mathbb{v} = \mathbb{b}$ (e.g. the discretization of system (3.4)). We call $V \subseteq \mathbb{R}^n$ the discrete space in which the solution of the linear system is defined. Following the notations used in [9], we introduce the set of indexes $S = \{1 \dots n\}$ representing the degrees of freedom of the system. Given an overlapping partition of S in I subsets, $\{S_i\}_{1 \leqslant i \leqslant I}$, we define the subspaces $V_i \subseteq V$ as $V_i = \{\mathbb{v} = (v_1, \dots v_n)^T \in \mathbb{R}^n | v_k = 0 \text{ if } k \notin S_i\}$ of dimension n_i. Next we introduce the restriction matrix $R_i \in \mathbb{R}^{n_i \times n}$ such that $(R_i)_{lj}\delta_{jk} = \delta_{lk}$, $l \in S_i$, $k, j \in S$ and δ is the Kronecker symbol. The prolongation matrix $R_i^T \in \mathbb{R}^{n \times n_i}$ transforms a short vector of n_i components into one with n components by keeping the original components and setting to zero the new ones. In this framework we define the one level AAS preconditioner associated to A as

$$P_{AS}^{-1}(A) = \sum_{i=0}^{I} R_i^T (R_i A R_i^T)^{-1} R_i.$$

It is possible to add a *coarse component* (corresponding to $i = 0$) to improve the condition number when the number of subdomains increases. The parallel structure of our code[2] includes a mesh partitioner based on the package `ParMETIS` and an AAS preconditioner handled by `IFPACK`, both embedded in the `Trilinos`[3] library. The code ensures that each subdomain is composed by a fluid part and a solid part not necessarily connected. The portions of fluid and solid domain are thus balanced throughout the processors (see Figure 7.1).

5. Geometry-Convective Explicit Time Discretization. In this section we focus on some choices of the preconditioner for the GCE linear system. As pointed out in (3.3) and (3.2), the equation for the fluid domain displacement can be solved in a separate step. We can express the coupled $\mathsf{F} - \mathsf{S}$ linear system resulting from the space-time discretization as

$$\left(\begin{array}{cc|cc|c} C_{\mathrm{ff}} & C_{\mathrm{f}\Gamma} & 0 & 0 & 0 \\ C_{\Gamma\mathrm{f}} & C_{\Gamma\Gamma} & 0 & 0 & I \\ \hline 0 & 0 & N_{\mathrm{ss}} & N_{\mathrm{s}\Gamma} & 0 \\ 0 & 0 & N_{\mathrm{s}\Gamma} & N_{\Gamma\Gamma} & -I \\ \hline 0 & I & 0 & -\Delta_t & 0 \end{array}\right) \left(\begin{array}{c} \mathbb{u}^{n+1} \\ \mathbb{u}_\Gamma^{n+1} \\ \hline \mathbb{d}_{\mathrm{s}}^{n+1} \\ \mathbb{d}_{\mathrm{s},\Gamma}^{n+1} \\ \hline \lambda^{n+1} \end{array}\right) = \left(\begin{array}{c} \mathbb{f}_{\mathrm{f}}^{n+1} \\ \mathbb{f}_{\Gamma\mathrm{f}}^{n+1} \\ \hline \mathbb{f}_{\mathrm{s}}^{n+1} \\ \mathbb{f}_{\Gamma s}^{n+1} \\ \hline -\mathbb{d}_{\mathrm{s},\Gamma}^{n}/\delta t \end{array}\right), \tag{5.1}$$

[2] LifeV, `http://www.lifev.org`

[3] `http://trilinos.sandia.gov`

where $\Delta_t = I/\Delta t$, $(\mathbb{u}, \mathbb{u}_\Gamma)$ is the discrete solution of the Navier-Stokes equations, while $(\mathbb{d}_s, \mathbb{d}_{s,\Gamma})$ is the discretized structure displacement. The quantities with label Γ are those associated with the nodes lying on the interface between fluid and structure. An augmented formulation is used to impose the continuity of the stresses and velocities at the interface, leading to the introduction of the new variables λ. In what follows the block matrix in (5.1) will be denoted by A.

5.1. Block Preconditioning. In this section we define the block-representation of the preconditioners for the matrix A. In particular we distinguish between two types of block Gauss-Seidel preconditioners corresponding to the situation described in (4.4) when we neglect either the fluid or the solid extradiagonal block. Referring to the matrix in (5.1) we call $P_{GS}^{(1)}$ the block G-S preconditioner obtained by neglecting the term $-I$, while we denote $P_{GS}^{(2)}$ the matrix obtained by neglecting the block $-\Delta_t$.

Notice that for $i \in \{1, 2\}$ one application of $(P_{GS}^{(i)})^{-1}$ implies the solution of a Dirichlet problem in the fluid subdomain and of a Neumann problem in the structure subdomain. As a consequence these preconditioners suffer of the same inconsistency dilemma of the Dirichlet-Neumann scheme when imposing Dirichlet conditions everywhere on the fluid boundary, as described in [28]. There are several ways to overcome this problem. One is to instead neglect the identity term appearing on the second block row of A. This would lead to a Neumann problem on the fluid domain and a Dirichlet problem on the solid. Another possibility is to substitute the last block row with a linear combination $R_5 \to \alpha_1 R_5 + \alpha_2(R_2 + R_4)$, with properly tuned parameters α_1 and α_2; then by neglecting the $-I$ term we obtain a Robin interface condition for the fluid problem.

We remark that the construction of both the preconditioners $P_{GS}^{(1)}$ and $P_{GS}^{(2)}$ is *modular*, in the sense that they can be obtained through the multiplication of two matrices containing the solid and the fluid blocks, respectively. In fact the factorization for $P_{GS}^{(1)}$ reads:

$$
\begin{aligned}
P_{GS}^{(1)} &= \left(\begin{array}{cc|cc|c}
C_{\mathrm{ff}} & C_{\mathrm{f}\Gamma} & 0 & 0 & 0 \\
C_{\Gamma\mathrm{f}} & C_{\Gamma\Gamma} & 0 & 0 & I \\
\hline
0 & 0 & N_{\mathrm{ss}} & N_{\mathrm{s}\Gamma} & 0 \\
0 & 0 & N_{\mathrm{s}\Gamma} & N_{\Gamma\Gamma} & 0 \\
\hline
0 & I & 0 & -\Delta_t & 0
\end{array}\right) \\
&= \left(\begin{array}{cc|cc|c}
I & 0 & 0 & 0 & 0 \\
0 & I & 0 & 0 & 0 \\
\hline
0 & 0 & N_{\mathrm{ss}} & N_{\mathrm{s}\Gamma} & 0 \\
0 & 0 & N_{\mathrm{s}\Gamma} & N_{\Gamma\Gamma} & 0 \\
\hline
0 & 0 & 0 & 0 & I
\end{array}\right)
\left(\begin{array}{cc|cc|c}
C_{\mathrm{ff}} & C_{\mathrm{f}\Gamma} & 0 & 0 & 0 \\
C_{\Gamma\mathrm{f}} & C_{\Gamma\Gamma} & 0 & 0 & I \\
\hline
0 & 0 & I & 0 & 0 \\
0 & 0 & 0 & I & 0 \\
\hline
0 & I & 0 & -\Delta_t & 0
\end{array}\right) = P_{GS,1}^{(1)} P_{GS,2}^{(1)} \,, \qquad (5.2)
\end{aligned}
$$

while the factorization for $P_{GS}^{(2)}$ takes the form

$$
\begin{aligned}
P_{GS}^{(2)} &= \left(\begin{array}{cc|cc|c}
C_{\mathrm{ff}} & C_{\mathrm{f}\Gamma} & 0 & 0 & 0 \\
C_{\Gamma\mathrm{f}} & C_{\Gamma\Gamma} & 0 & 0 & I \\
\hline
0 & 0 & N_{\mathrm{ss}} & N_{\mathrm{s}\Gamma} & 0 \\
0 & 0 & N_{\mathrm{s}\Gamma} & N_{\Gamma\Gamma} & -I \\
\hline
0 & I & 0 & 0 & 0
\end{array}\right) \\
&= \left(\begin{array}{cc|cc|c}
C_{\mathrm{ff}} & C_{\mathrm{f}\Gamma} & 0 & 0 & 0 \\
C_{\Gamma\mathrm{f}} & C_{\Gamma\Gamma} & 0 & 0 & I \\
\hline
0 & 0 & I & 0 & 0 \\
0 & 0 & 0 & I & 0 \\
\hline
0 & I & 0 & 0 & 0
\end{array}\right)
\left(\begin{array}{cc|cc|c}
I & 0 & 0 & 0 & 0 \\
0 & I & 0 & 0 & 0 \\
\hline
0 & 0 & N_{\mathrm{ss}} & N_{\mathrm{s}\Gamma} & 0 \\
0 & 0 & N_{\mathrm{s}\Gamma} & N_{\Gamma\Gamma} & -I \\
\hline
0 & 0 & 0 & 0 & I
\end{array}\right) . = P_{GS,1}^{(2)} P_{GS,2}^{(2)} \,, \qquad (5.3)
\end{aligned}
$$

These preconditioners have been considered in [24] in a serial context, where the factorization of the fluid and structure matrices can be computed through a direct or an iterative method. However a parallel approach would require either a parallel direct solver for the sub-blocks or a certain number of inner iterations to compute each matrix-vector product $P^{-1}\mathbb{x}$. Furthermore

frequently the number of inner iterations increases when the number of processors grows (it is the case for the one-level domain decomposition preconditioners).

The approach that we advocate in this work consists in substituting the fluid and solid sub-blocks with suitable preconditioners, whose factorization can be computed in parallel. More precisely, instead of $P_{GS}^{(1)}$ and $P_{GS}^{(2)}$, we use, respectively $P_{GS-AS}^{(1)} = P_{AS}(P_{GS,1}^{(1)}) \cdot P_{AS}(P_{GS,2}^{(1)})$ and $P_{GS-AS}^{(2)} = P_{AS}(P_{GS,1}^{(2)}) \cdot P_{AS}(P_{GS,2}^{(2)})$. According to the domain decomposition terminology this choice corresponds to employ inexact solvers for the solid and fluid subdomains. A spectral analysis for a similar kind of block triangular preconditioners for stabilized saddle point problems, with symmetric positive definite diagonal blocks, is carried out e.g. in [26, 36]. With this approach we avoid the inner iterations, since we can solve the local problems by LU factorization.

The following preconditioning techniques have been numerically compared:

1. one-level AAS preconditioner built using the matrix A in (5.1): $P_{AS}(A)$;
2. one-level AAS preconditioner built using an approximation of the type (4.4), obtained by neglecting the block $-I$ in matrix (5.1): $P_{GS-AS}^{(1)}$;
3. one-level additive Schwarz preconditioner built using a different approximation of (5.1), obtained neglecting the term $-\Delta_t$: $P_{GS-AS}^{(2)}$.

The preconditioners $P_{GS-AS}^{(1)}$ and $P_{GS-AS}^{(2)}$, besides preserving the modularity in their construction, have similar or better behavior of $P_{AS}(A)$ when increasing the number of processors. Furthermore, their factorization is cheaper in terms of computational time and memory usage than building the factorization of the whole matrix A. In our framework, one can chose different preconditioning techniques for the different sub-blocks, which is desirable for multiphysics systems, since physics-specific preconditioners can be used.

6. Convective Explicit Time Discretization. This section is devoted to the preconditioning techniques for the Jacobian system in the Newton algorithm for the convective-explicit time discretization of the FSI problem.

The Jacobian matrix appearing in (3.4) takes the form

$$J_{CE} = \left(\begin{array}{cc|cc|c|cc} C_{\mathrm{ff}} & C_{\mathrm{f}\Gamma} & 0 & 0 & 0 & \partial_{\mathbf{d}_\mathrm{f}} C_\mathrm{f} & \partial_{\mathbf{d}_{\mathrm{f},\Gamma}} C_\mathrm{f} \\ C_{\Gamma\mathrm{f}} & C_{\Gamma\Gamma} & 0 & 0 & I & \partial_{\mathbf{d}_\mathrm{f}} C_\Gamma & \partial_{\mathbf{d}_{\mathrm{f},\Gamma}} C_\Gamma \\ \hline 0 & 0 & N_{\mathrm{ss}} & N_{\mathrm{s}\Gamma} & 0 & 0 & 0 \\ 0 & 0 & N_{\Gamma s} & N_{\Gamma\Gamma} & -I & 0 & 0 \\ \hline 0 & I & 0 & -\Delta_t & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & H_{\mathrm{ff}} & H_{\mathrm{f}\Gamma} \\ 0 & 0 & 0 & -I & 0 & 0 & I \end{array}\right), \tag{6.1}$$

where the block rows correspond, from top to bottom, to the fluid problem, the solid problem, the velocity continuity through the interface, and the geometry problem. We used the notations $\partial_{\mathbf{x}} C_\mathrm{f} = \partial_{\mathbf{x}} C_{\mathrm{ff}} + \partial_{\mathbf{x}} C_{\mathrm{f}\Gamma}$ and $\partial_{\mathbf{x}} C_\Gamma = \partial_{\mathbf{x}} C_{\Gamma\mathrm{f}} + \partial_{\mathbf{x}} C_{\Gamma\Gamma}$. Since we consider a linear structure and we extrapolate the convective terms, the only nonlinearity comes from the dependence of the fluid problem on the domain displacement $\mathbf{d}_\mathrm{f}$. For this reason the assembling cost of the Jacobian matrix at each Newton iteration reduces to the cost of assembling the upper-right block representing the shape derivatives, while the remaining part is already computed to evaluate the current residual.

Preconditioning strategies like those described in Section §4.1 can be applied also to this linearized system. In particular we will consider in the following three preconditioners

- $P_{AS}(J_{CE})$;
- $P_{QN-AS} = P_{AS}(J_{QN})$;
- $P_{GS-AS} = P_{AS}(P_{GS,1})P_{AS}(P_{GS,2})$ where $P_{GS,1}$ and $P_{GS,2}$ are defined as follows

$$J_{QN} = \left(\begin{array}{cc|cc|c|cc} C_{\mathrm{ff}} & C_{\mathrm{f}\Gamma} & 0 & 0 & 0 & 0 & 0 \\ C_{\Gamma\mathrm{f}} & C_{\Gamma\Gamma} & 0 & 0 & I & 0 & 0 \\ \hline 0 & 0 & N_{\mathrm{ss}} & N_{\mathrm{s}\Gamma} & 0 & 0 & 0 \\ 0 & 0 & N_{\Gamma s} & N_{\Gamma\Gamma} & -I & 0 & 0 \\ \hline 0 & I & 0 & -\Delta_t & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & H_{\mathrm{ff}} & H_{\mathrm{f}\Gamma} \\ 0 & 0 & 0 & -I & 0 & 0 & I \end{array}\right) =$$

$$\left(\begin{array}{cc|cc|c|c|c} C_{\mathrm{ff}} & C_{\mathrm{f}\Gamma} & 0 & 0 & 0 & 0 & 0 \\ C_{\Gamma\mathrm{f}} & C_{\Gamma\Gamma} & 0 & 0 & I & 0 & 0 \\ \hline 0 & 0 & N_{\mathrm{ss}} & N_{\mathrm{s}\Gamma} & 0 & 0 & 0 \\ 0 & 0 & N_{\Gamma\mathrm{s}} & N_{\Gamma\Gamma} & -I & 0 & 0 \\ \hline 0 & I & 0 & -\Delta_t & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I \end{array}\right) \left(\begin{array}{cc|cc|c|c|c} I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & I & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & H_{\mathrm{ff}} & H_{\mathrm{f}\Gamma} \\ 0 & 0 & 0 & -I & 0 & 0 & I \end{array}\right) = P_{GS,1} P_{GS,2} \ ,$$

We can thus construct the preconditioner in a modular way, separating the FS block from the harmonic extension problem. It is worth noticing that in a serial framework as the domain is not partitioned the preconditioners P_{QN-AS} and P_{GS-AS} are equal and correspond to a block GS preconditioner. Other choices for block partitioning and ordering, like 3 block structures, are possible; here we present only the simplest ones.

7. Numerical Simulations. In this section we report the numerical behavior of the preconditioner choices presented so far. In both the cases considered, i.e., the GCE and the CE described above, our preconditioning strategy shows to be effective in terms of computational time. As a general comment, splitting the matrix before building the operators P_{AS} yelds better computational times and lower number of GMRES iterations (see Figures 7.2 to 7.9). The theoritical reason derives from the analysis in Section §4.1, the empirical one is in how the preconditioners couple the different blocks.

The simulations are performed using the FE library LifeV[4]. The AAS preconditioner used in all the simulations has a 2 layers overlap between the partitions. The FSI solver implemented in LifeV shows to be scalable in the matrix assembling and in the preconditioner computation steps. The only lack of scalability is due to the increase of GMRES iterations when the processors number grows. This can be perhaps avoided employing multilevel domain decomposition preconditioners for the sub blocks and keeping the same modular approach described here.

The behavior of the preconditioners is tested on unstructured 3D cylindrical meshes with different characteristic lengths h, and on physiological 3D geometries as well, computed through segmentation of medical images (e.g. CT scans or MRI). The finite elements chosen are $P1 - P1$ tetrahedra for the fluid, stabilized using the interior penalty technique described in [32], P1 tetrahedra for the structure.

The tests reported in this section were performed on cluster composed of blades with two 4-cores processors Intel Harptown (3.0 Ghz) each. The blades are interconnected through InfiniBand.

7.1. Geometry-Convective Explicit Time Discretization. A benchmark geometry, similar to the one proposed in [21] (and used e.g. in [14, 27, 18]) consists of a straight cylinder of length $10cm$ and with radius $0.5cm$ representing the fluid domain, surrounded by a structure of constant thickness $0.1cm$ (see Figure 7.1). The inlet boundary condition on the fluid domain is a pressure step function, taking the constant value $p_{in} = 1.33 \cdot 10^4 \frac{dyne}{cm^2}$ for $t \leqslant 0.003$. The boundary conditions for the outlet and for the external structure are of Neumann homogeneous type. The time step used is $\delta t = 10^{-3}$ while the parameters characterizing the model are fluid viscosity $\mu_{\mathrm{f}} = 0.03P$, the Young modulus $E = 3\ 10^6 \frac{dyne}{cm^2}$, the Poisson ratio $\nu = 0.3$ and the mass densities of fluid and structure $\rho_f = 1.0 \frac{g}{cm^3}$ and $\rho_s = 1.2 \frac{g}{cm^3}$. The tolerance for the GMRES solver is

[4] `http://www.lifev.org`

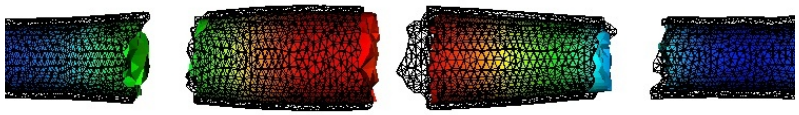

FIG. 7.1. *The cylindrical geometry used partitioned into* 4 *subdomains.*

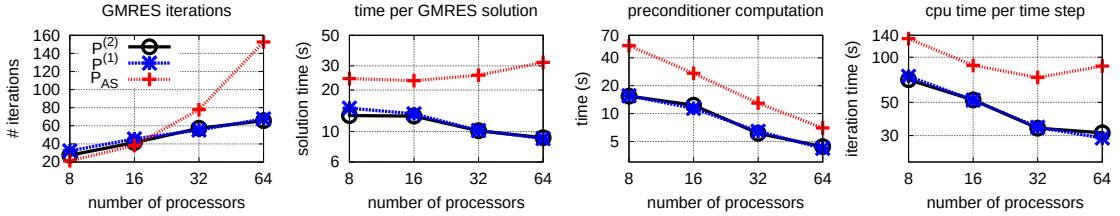

FIG. 7.2. *GCE — Simulations run on the coarser tube mesh.*

$\|\mathbb{r}\|_2/\|\mathbb{r}_0\|_2 < 10^{-6}$, where the norm $\|\cdot\|_2$ is the sum of the vector components squared and $\mathbb{r}$ is the linear residual. The values reported in the following are the mean over the first 30 time iterations.

The simulations run on the benchmark test show that the preconditioners of type $P^{(i)}_{GS-AS}$ for $i \in \{1, 2\}$ are more efficient than P_{AS} in terms of computational time and GMRES iterations. Furthermore the gap between these preconditioning strategies increases with the processors number. This result is validated with two different meshes (in Figures 7.2 and 7.3) and varying the ratio between the mass densities $r = \frac{\rho_f}{\rho_s}$ (in Figure 7.4). The two meshes considered are composed by 169267 and 578594 tetrahedra, corresponding to respectively 216441 and 630468 degrees of freedom. We observe that in view of the analysis performed in Section §4.1 the mild growth of the GMRES iterations in the modular cases can be explained with a boundedness of the singular value σ_Σ. The lack of scalability, due to the increasing number of GMRES iterations, is related to our choice of using the one level AAS preconditioner in the tests preformed. However the block preconditioners introduced in the present work can be built using any other preconditioning strategy.

7.2. Convective Explicit Time Discretization. The simulations on the benchmark geometry use the same physical parameters and boundary conditions as in the GCE case. The tolerance of the linear solver is set to 10^{-6} while the tolerance for the Newton method is $\|\mathbb{R}\|_2/\|\mathbb{R}_o\|_2 < 10^{-5}$ where $\mathbb{R}$ represents the nonlinear residual.

In Figure 7.5 we observe that the preconditioner P_{QN-AS} is cheaper to compute than $P_{AS}(J_{CE})$, while it has the same behavior in terms of GMRES iterations. Although P_{GS-AS} is cheaper to compute than the other choices it worsen faster when increasing the number of processors. This can be due to the growth of the singular value σ_Σ in the estimate (4.10a), when augmenting the number of processors. Thus P_{GS-AS} seems not to be a choice well suited for a massively parallel framework.

7.3. Physiological geometries. The preconditioners devised for the CE system are tested on physiological geometries (Figure 7.7), obtained through segmentation of medical images using VMTK[5]. The physical parameters chosen for the model are $\mu_f = 0.35P$, $E = 4\ 10^6 \frac{dyne}{cm^2}$, $\nu = 0.48$, $\rho_s = 1.2\frac{g}{cm^3}$ and $\rho_f = 1.0\frac{g}{cm^3}$. Since the purpose of these simulations are mainly to test the behavior of the preconditioners rather than to observe the dynamic of the flow during an entire heartbeat we only consider the mean values over the first 30 time steps ($0.03s$). The geometries

[5] `http://www.vmtk.org`

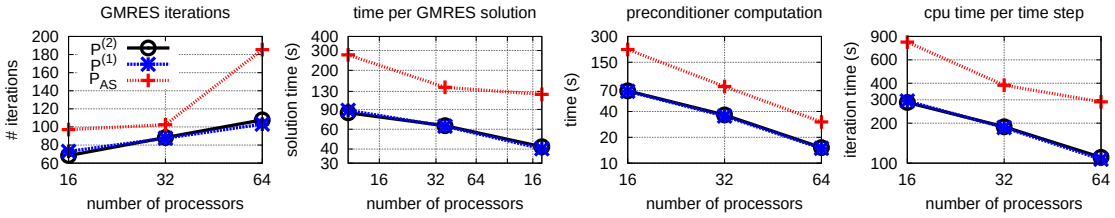

FIG. 7.3. *GCE — Simulations run on the finer tube mesh.*

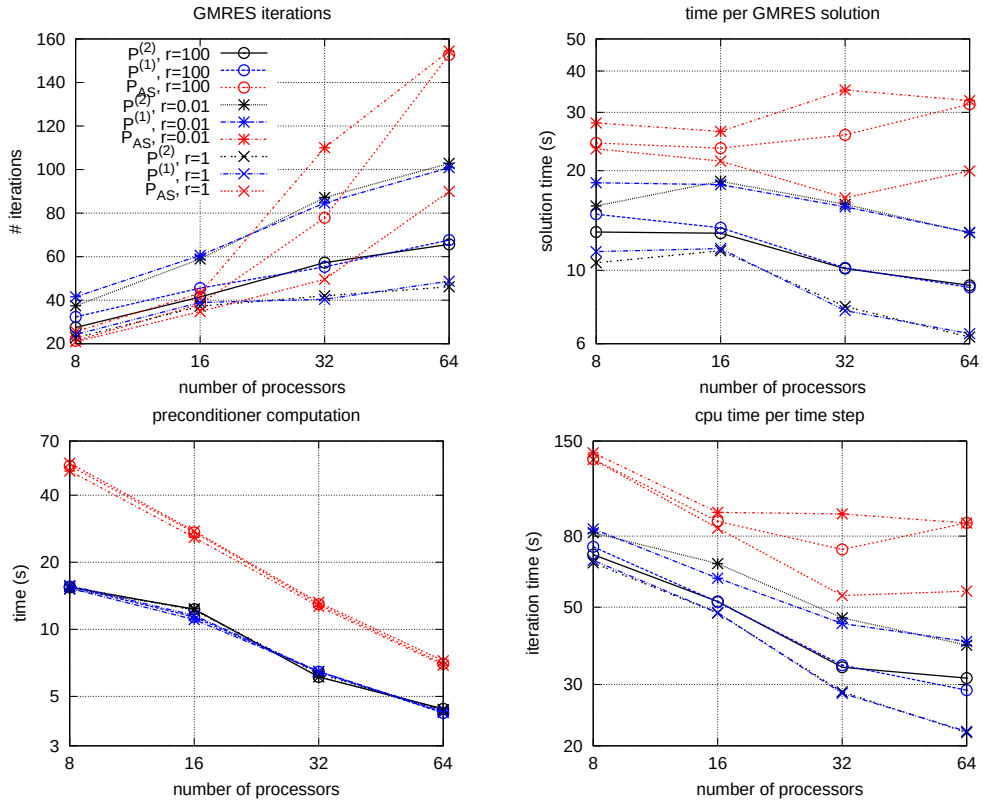

FIG. 7.4. *GCE — Comparison on the coarser tube mesh between the results obtained for different values of the ratio* $r = \rho_f/\rho_s$.

considered in our simulations represent an aorta, starting from the aortic valve, including the aortic arch, the thoracic and abdominal aorta. The inlet pressure imposed at the aortic valve is taken from physiological measurements corresponding to the beginning of an heartbeat cycle $p_{in} \approx 1.1 \cdot 10^5 \frac{dyne}{cm^2}$. The boundary conditions imposed at the outflows and on the external wall are of Neumann homogeneous type. Since no spurious reflection waves are originated in the first 30 time steps (see e.g. [31]), there is no need to impose absorbing boundary conditions at the outflows. The blood is considered at rest at the beginning of the simulation. The simulations are performed on two different meshes, with respectively 105810 and 380690 tetrahedra for a total of 135000 and 486749 degrees of freedom. We keep the same time step $\delta t = 10^{-3}$ used in the previous simulations.

We can notice that the preconditioner P_{GS-AS} worsen faster than the others when the number of processors increases (Figures 7.8 and 7.9). This phenomenon was already observed in the previous benchmark tests. The most convenient preconditioner among those considered turns out to be again P_{QN-AS}. The increasing of the global computational time when passing from 32 to

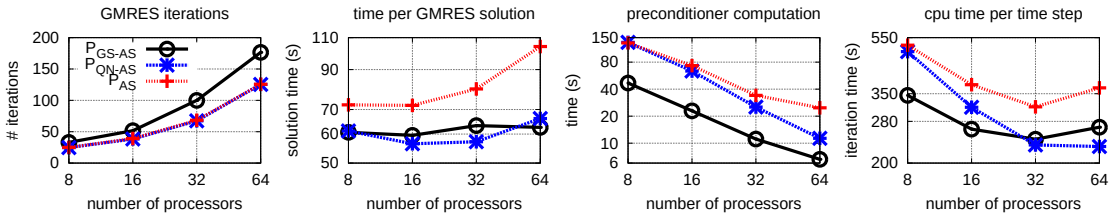

FIG. 7.5. *CE — Simulations run on the coarser tube mesh.*

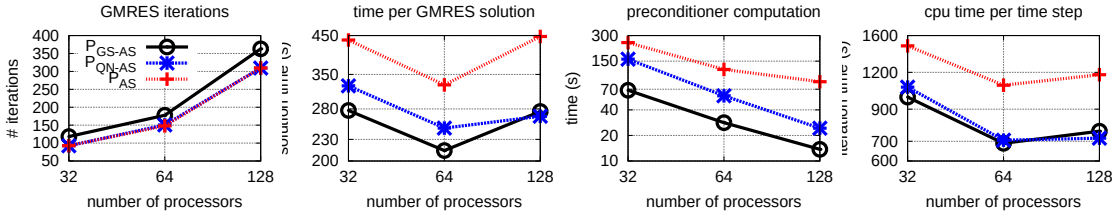

FIG. 7.6. *CE — Simulations run on the finer tube mesh.*

64 processors in the coarser case is due partly to the relatively small mesh size of the problem addressed and partly to the communication time affecting both the GMRES solution and the preconditioner computation.

8. Conclusions. In this work we focused on two strategies, that differ in the time discretization adopted, to solve the FSI problem in parallel and we set a terminology that identifies the methods among all the different approaches. The GCE time discretization leads to a linear discrete problem. It is obtained treating explicitly the fluid domain and it allows to solve the fluid geometry problem in a separate step. On the other hand in the CE approach the fluid domain is treated implicitly and at each time step the nonlinear problem is solved using the Newton scheme. A key issue in the parallel solution of the FSI problem is the choice of the preconditioner applied to the linear (Jacobian in the CE case) system. We introduced a parallel preconditioning technique suited for multiphysic problems (with a block structure) that exploits the characteristics of the subproblems. We proved a theoretical bound on the condition number for the preconditioned system. We showed that if the maximum singular value of Σ (see Section §4.1) is bounded, then the conditioning of the whole system and that of the subproblems have the same behavior. In particular the scalability is not influenced by the coupling. Furthermore another appealing characteristics of the parallel preconditioners introduced is their *modularity*, in the sense that they can be split into several factors, containing one subproblem each, thus for each of them it is possible to choose an appropriate preconditioning strategy. In this work we limited our study in this work to two factors, using the same preconditioning strategy (AAS) for both.

For both GCE and CE time discretizations the preconditioners are tested on different meshes and by varying the physical parameters. The comparisons show in most of the cases an evident advantage in using the modular type of preconditioners introduced. The parallel behavior of the preconditioners is independent of the mesh size and of the physical parameters in the tested cases.

Acknowledgments. We thank Jean Bonnemain for his contribution in the segmentation of the medical images. Acknowledgments for the support provided through the Swiss National Science Foundation under grant 200020-117587. The research leading to these results has also received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 224635.

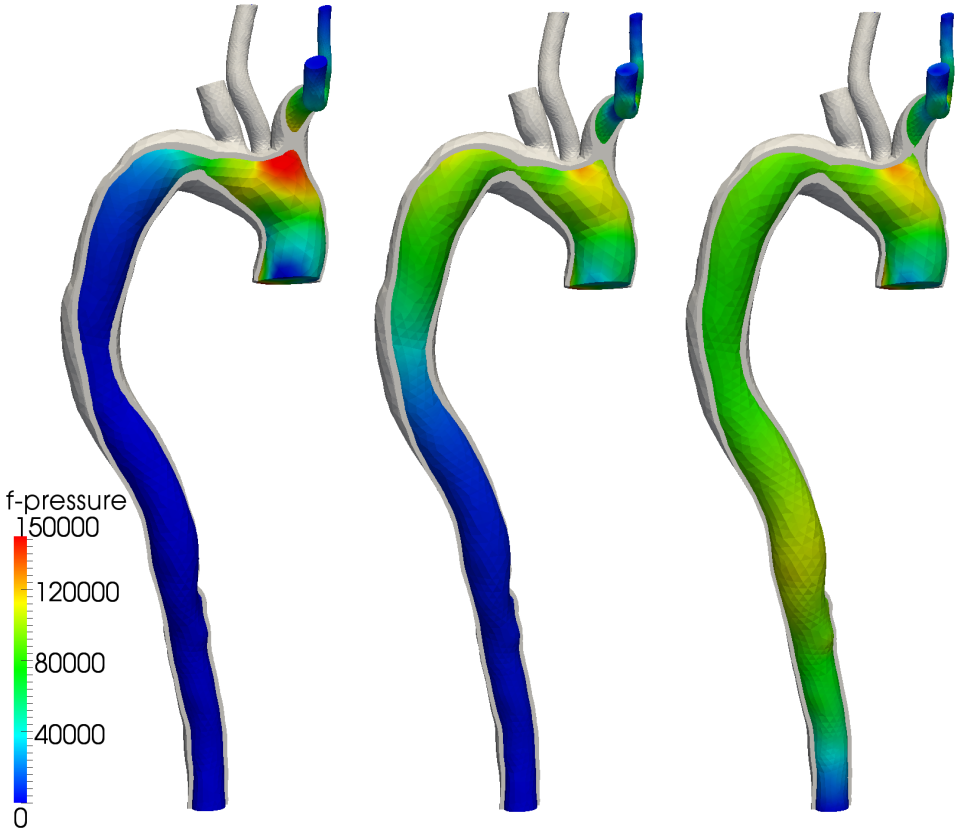

FIG. 7.7. *Blood pressure (in dyne/cm^2) and the deformation of an aorta at $t = 0.015s$, $t = 0.030s$, and $t = 0.075s$.*

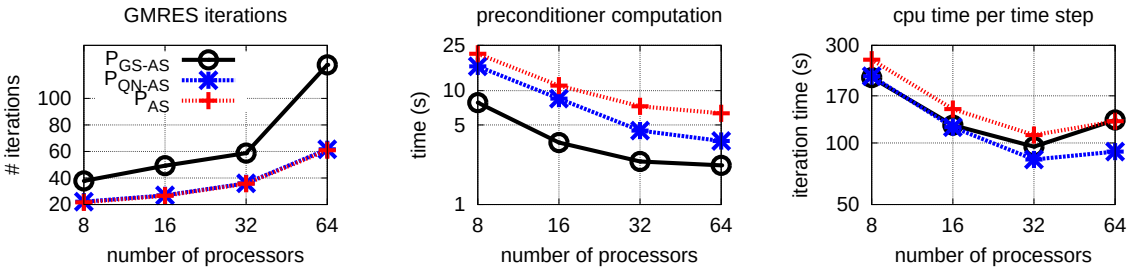

FIG. 7.8. *CE — Simulations on the aorta on Figure 7.7, coarser mesh.*

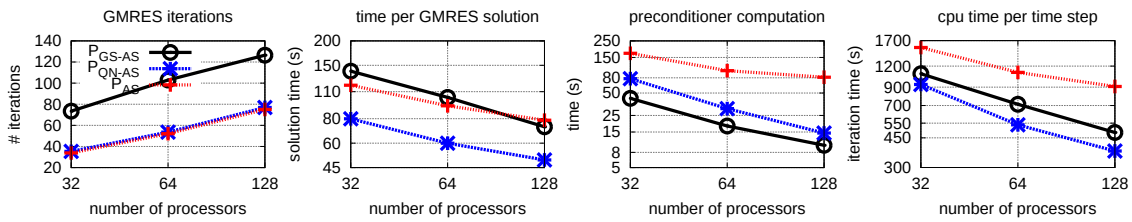

FIG. 7.9. *CE — Simulations on the aorta on Figure 7.7, finer mesh.*

REFERENCES

[1] M. Astorino, F. Chouly, and M. A. Fernández, *Robin based semi-implicit coupling in fluid-structure interaction: Stability analysis and numerics*, To appear, (2009).

[2] Owe Axelsson, *Iterative solution methods*, Cambridge University Press, New York, NY, USA, 1994.

[3] Santiago Badia, Fabio Nobile, and Christian Vergara, *Fluid-structure partitioned procedures based on Robin transmission conditions*, J. Comput. Phys., 227 (2008), pp. 7027–7051.

[4] Santiago Badia, Annalisa Quaini, and Alfio Quarteroni, *Modular vs. non-modular preconditioners for fluid-structure systems with large added-mass effect*, Comput. Methods Appl. Mech. Engrg., 197 (2008), pp. 4216–4232.

[5] ———, *Splitting methods based on algebraic factorization for fluid-structure interaction*, SIAM J. Sci. Comput., 30 (2008), pp. 1778–1805.

[6] A. T. Barker and X. C. Cai, *NKS for fully coupled fluid-structure interaction with application*, Lect. Notes Comput. Sci. Eng., Springer, Berlin.

[7] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang, *Isogeometric fluid-structure interaction: theory, algorithms, and computations*, Comput. Mech., 43 (2008), pp. 3–37.

[8] Erik Burman and Miguel A. Fernández, *Stabilization of explicit coupling in fluid-structure interaction involving fluid incompressibility*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 766–784.

[9] Xiao-Chuan Cai and David E. Keyes, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 24 (2002), pp. 183–200 (electronic).

[10] P. Causin, J.-F. Gerbeau, and F. Nobile, *Added-mass effect in the design of partitioned algorithms for fluid-structure problems*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 4506 – 4527.

[11] Georges-Henri Cottet, Emmanuel Maitre, and Thomas Milcent, *Eulerian formulation and level set models for incompressible fluid-structure interaction*, M2AN Math. Model. Numer. Anal., 42 (2008), pp. 471–492.

[12] Joris Degroote, Klaus-Jrgen Bathe, and Jan Vierendeels, *Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction*, Computers and Structures, 87 (2009), pp. 793 – 801. Fifth MIT Conference on Computational Fluid and Solid Mechanics.

[13] Simone Deparis, *Numerical analysis of axisymmetric flows and methods for fluid-structure interaction arising in blood flow simulation*, PhD thesis, Lausanne, 2004.

[14] Simone Deparis, Marco Discacciati, Gilles Fourestey, and Alfio Quarteroni, *Fluid-structure algorithms based on Steklov-Poincaré operators*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 5797–5812.

[15] W. G. Dettmer and D. Perić, *A fully implicit computational strategy for strongly coupled fluid-solid interaction*, Arch. Comput. Methods Eng., 14 (2007), pp. 205–247.

[16] Howard Elman, Victoria E. Howle, John Shadid, Robert Shuttleworth, and Ray Tuminaro, *Block preconditioners based on approximate commutators*, SIAM J. Sci. Comput, 27 (2006), pp. 1651–1668.

[17] Howard Elman, David Silvester, and Andy Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, 2005, ch. 8. xiv+400 pp. ISBN: 978-0-19-852868-5; 0-19-852868-X.

[18] M.A. Fernández and M. Moubachir, *A Newton method using exact jacobians for solving fluid-structure coupling*, Computers and Structures, 83 (2005), pp. 127–142.

[19] M. A. Fernández, J.-F. Gerbeau, and C. Grandmont, *A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid*, Internat. J. Numer. Methods Engrg., 69 (2007), pp. 794–821.

[20] Luca Formaggia, Alfio Quarteroni, and Alessandro Veneziani, eds., *Cardiovascular Mathematics: Modeling and Simulation of the Circulatory System*, vol. 1 of Modeling, Simulation and Applications, Springer, Milan, 2009.

[21] J.F. Gerbeau and M. Vidrascu, *A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows*, M2AN, 37 (2003), pp. 663–680.

[22] Jean-Frédéric Gerbeau, *A quasi-Newton method for a fluid-structure problem arising in blood flows*, in Proceedings of the second M.I.T. Conference on Computational Fluid and Solid Mechanics, K.J. Bathe, ed., Elsevier, 2003, pp. 1355–1357.

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

[24] Matthias Heil, Andrew L. Hazel, and Jonathan Boyle, *Solvers for large-displacement fluidstructure interaction problems: segregated versus monolithic approaches*, Computational Mechanics, 43 (2008), pp. 91–101.

[25] Björn Hübner, Elmar Walhorn, and Dieter Dinkler, *A monolithic approach to fluid-structure interaction using space-time finite elements*, Computer Methods in Applied Mechanics and Engineering, 193 (2004), pp. 2087 – 2104.

[26] Axel Klawonn, *Block-triangular preconditioners for saddle point problems with a penalty term*, SIAM J. Sci. Comput., 19 (1998), pp. 172–184.

[27] U. Küttler, M. Gee, Ch Förster, A. Comerford, and W. A. Wall, *Coupling strategies for biomedical fluid–structure interaction problems*, Commun. Numer. Meth. Engng., (2009).

[28] Ulrich Küttler and Wolfgang A. Wall, *The dilemma of domain decomposition approaches in fluid-structure interactions with fully enclosed incompressible fluids*, in Domain decomposition methods in

science and engineering XVII, vol. 60 of Lect. Notes Comput. Sci. Eng., Springer, Berlin, 2008, pp. 575–582.

[29] U. Küttler and W. A. Wall, *Fixed-point fluid structure interaction solvers with dynamic relaxation*, Computational Mechanics, 43 (2008), pp. 61–72.

[30] Hermann G. Matthies, Rainer Niekamp, and Jan Steindorf, *Algorithms for strong coupling procedures*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 2028–2049.

[31] F. Nobile, *Numerical Approximation of Fluid-Structure Interaction Problems with Application to Haemodynamics*, PhD thesis, École Polytechnique Fédérale de Lausanne, 2001.

[32] Nicola Parolini and Erik Burman, *Subgrid edge stabilization for transport equations*, tech. report, 2005. EPFL-IACS report 09.2005.

[33] Micol Pennacchio and Valeria Simoncini, *Algebraic multigrid preconditioners for the bidomain reaction–diffusion system*, Submitted, (2009).

[34] A. Quaini and A. Quarteroni, *A semi-implicit approach for fluid-structure interaction based on an algebraic fractional step method*, Math. Models Methods Appl. Sci., 17 (2007), pp. 957–983.

[35] A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, Oxford, 1999.

[36] V. Simoncini, *Block triangular preconditioners for symmetric saddle-point problems*, Appl. Numer. Math., 49 (2004), pp. 63–80.

[37] Tayfun E. Tezduyar, Sunil Sathe, and Keith Stein, *Solution techniques for the fully discretized equations in computation of fluid-structure interactions with the space-time formulations*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 5743–5753.

[38] Andrea Toselli and Olof Widlund, *Domain decomposition methods—algorithms and theory*, vol. 34 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2005.

[39] Hongwu Wang, Jack Chessa, Wing K. Liu, and Ted Belytschko, *The immersed/fictitious element method for fluid-structure interaction: volumetric consistency, compressibility and thin members*, Internat. J. Numer. Methods Engrg., 74 (2008), pp. 32–55.