COMPDYN 2015
5th ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, V. Papadopoulos, V. Plevris (eds.)
Crete Island, Greece, 25–27 May 2015

MULTIGRID SOLVER WITH DOMAIN DECOMPOSITION SMOOTHING FOR STEADY-STATE INCOMPRESSIBLE FSI PROBLEMS

Eugenio Aulisa[1], **Simone Bnà**[2], **and Giorgio Bornia**[1]

[1]Department of Mathematics and Statistics, Texas Tech University
Lubbock, TX 79409
e-mail: {eugenio.aulisa,giorgio.bornia}@ttu.edu

[2] SCAI-CINECA (Super Computing Applications and Innovation)
Casalecchio di Reno, BO, Italy
e-mail: simone.bna@cineca.it

Keywords: Fluid-Structure Interaction, Finite Element Methods, Multigrid, Domain Decomposition

Abstract. *In this paper we investigate the numerical performance of a monolithic Newton-multigrid solver with domain decomposition smoothers for the solution of a class of stationary incompressible FSI problems. The physics of the problem is described using a monolithic approach, where mass continuity and stress balance are automatically satisfied across the fluid-solid interface. The deformation of the fluid domain is taken into account within the nonlinear Newton iterations according to an Arbitrary Lagrangian Eulerian (ALE) scheme. Due to the complexity and variety of the operators, the implementation of the Jacobian matrix in the nonlinear iterations is not a trivial task. To this purpose, we make use of automatic differentiation tools for an exact computation of the Jacobian matrix. The numerical solution of steady-state problems is particularly challenging, due to the ill-conditioning of the induced stiffness matrix. Moreover, the enforcement of the incompressibility condition calls for the use of incompressible solvers either of mixed or segregated type. At each nonlinear outer iteration the resulting linearized system is solved with a geometric multigrid solver. We consider a GMRES smoother preconditioned by an Additive Schwarz Method (ASM). The domain decomposition of the preconditioner is driven by the natural splitting between fluid and solid domain. The numerical results of some benchmark tests for steady-state cases show agreement with the literature and an increased robustness with our choice of smoothers with respect to standard ones.*

1 INTRODUCTION

Fluid-Structure Interaction problems (FSI) are of great interest because of a large number of applications arising in various fields such as aerodynamics, elasticity, civil engineering, biomechanics and hemodynamics. Examples of interesting problems are the fluttering of wings and bridges, mechanical vibrations or water hammer effects in pipe networks, blood flow in large arteries, parachute modeling and oil flow in porous media.

Nowadays, many challenging questions are still open in the FSI community ranging from mathematical modeling and analysis to numerical discretization and computational issues. The fluid and the structure influence each other in a nonlinear way. Indeed, the structure changes its shape under the action of the fluid force and, conversely, the fluid flow is altered by the structure deformation. The numerical solution of FSI problems is a very challenging task. Several choices are possible for the description of the solid and fluid parts, the determination of the unkwnown variables (velocity, displacement, pressure in fluid and in solid), the treatment of the moving domain (ALE, interface tracking/capturing), the linearization procedure (linearize then discretize or viceversa; fixed point, Newton), the choice of the linear solver.

In this work we intend to address two main difficulties concerning FSI simulations: stationary cases and incompressible materials, both for the fluid and for the solid part. Our intent is to highlight the performance of multigrid methods with domain decomposition smoothers in the case of steady-state saddle-point problems. Pseudo-time stepping schemes, in which a stationary solution is reached as a limit of a time sequence, are typically considered due to a better conditioning of the linear systems to be solved. We do not consider here the use of pseudo-time steppers and we treat the fluid and solid as purely incompressible, without introducing slightly-compressible stabilization terms.

Concerning the coupling between the solid and fluid unknowns, we consider a monolithic approach. Monolithic algorithms solve simultaneously for the fluid and the structure unknowns in a unique solver, so that the solid and fluid regions are treated as a single continuum and the boundary conditions at the interface are automatically satisfied ([9, 12]). Monolithic algorithms are CPU-time expensive. This issue is enhanced in the case of incompressible formulation by the saddle-point character of the system. In this work we consider a monolithic strongly coupled strategy. This approach is the most robust and stable one among the strong coupling approaches, but it requires the development of strategies to reduce the computational time.

We consider the case of large deformations, thus obtaining a nonlinear system of equations for the displacement, velocity and pressure variables. The implementation of the Fréchet derivative of the nonlinear FSI operator needed for building the Jacobian matrix is not a trivial task. We have chosen to compute the exact Jacobian matrix by using Automatic Differentiation (AD) tools provided by the Adept software package [10]. Other variants have been proposed in literature for the linearization. In [11] quasi-Newton outer iterations with line search are performed and the Jacobian matrix is computed by a divided difference approach. A Newton method with the analytical computation of the Jacobian using shape-derivative calculus is considered in [6]. A quasi-Newton method in which the variation of the fluid domain in the fluid equations is neglected is proposed in [5, 4].

Inspired by the Arbitrary Lagrangian Eulerian (ALE) formulation for solving time-dependent PDEs in moving domains, we adopt this approach for the movement of the domain within the nonlinear iterations. An appropriate arbitrary mapping is chosen from the fluid fixed reference configuration to the current domain. The ALE mapping is constructed as a harmonic extension of the trace of the structure displacement on the fluid-solid interface.

The solution of the linear system in each linearization step is the most time-consuming part. Multigrid solvers are taken into account for the solution of large sparse linear systems due to their optimal computational complexity, at least for elliptic problems. In this paper we propose to use a geometric Newton-multigrid solver, in which Newton linearization is performed as an outer iteration and the resulting Jacobian system is solved using a geometric multigrid solver. Multigrid F-cycle schemes are considered with a GMRES smoother preconditioned by an additive Schwarz method (ASM). In this work we observed that the use of domain decomposition methods in the smoothing step guarantees convergence where standard smoothers fail. Another advantage of these methods is to allow for an effective parallel implementation. While the coarse grid solution is obtained by inverting the corresponding matrix with a direct sparse solver, several local subproblems are solved on finer levels over subdomain patches.

Both multigrid and domain decomposition methods draw a lot of attention within the FSI community. In [12] a geometric multigrid solver with a Multilevel Pressure Schur Complement (MPSC) Vanka-like smoother is considered, with applications to nonstationary FSI problems in biomechanics. Applications of this scheme to hemodynamics are also addressed in [22, 14]. Monolithic Newton-Krylov algorithms are studied in [8, 26]. In [8] the inner Krylov iterations are preconditioned with algebraic multigrid methods, while an overlapping additive Schwarz preconditioner is considered in [26] with application to parallel three-dimensional blood flow simulations. A partitioned method in which multigrid is used either within the fluid and solid solves or as an outer iteration is addressed in [16, 18]. Other numerical studies are available in the literature on the use of domain decomposition Vanka-type smoothers for multigrid both in Computational Fluid Dynamics (CFD) and in Computational Solid Mechanics (CSM) [1, 19, 24, 25, 14, 2].

The paper is organized as follows. In Section 2 we present the strong and weak formulations of the stationary incompressible FSI problem under investigation. We describe the linearization procedure by means of automatic differentiation, and we illustrate the features of the multigrid linear solver with domain decomposition smoothing. Numerical results of benchmark problems are presented in Section 3. Finally, we draw our conclusions.

2 FORMULATION OF THE FSI PROBLEM

2.1 Strong formulation

Let $\widehat{\Omega} \subset \mathbb{R}^3$ be a reference undeformed configuration for a given domain. Let $\Omega \subset \mathbb{R}^3$ be the deformed steady-state configuration. Let assume that there is a one-to-one, uniquely invertible, sufficiently smooth mapping $\mathcal{X}$ of the reference configuration $\widehat{\Omega}$ to the deformed configuration

$$\mathcal{X} : \widehat{\Omega} \rightarrow \Omega \,.$$

It is clear that the mapping $\mathcal{X}$ depends on the choice of the reference configuration $\widehat{\Omega}$. All the reference configurations are in principle equivalent. For instance, we can set $\widehat{\Omega}$ as the initial (stress-free) configuration. Let us denote an arbitrary material point in the reference configuration $\widehat{\Omega}$ by $\widehat{\mathbf{x}}$. The position of this point in the deformed configuration Ω is given by

$$\mathbf{x} = \mathcal{X}(\widehat{\mathbf{x}}), \quad \text{for all} \quad \widehat{\mathbf{x}} \in \widehat{\Omega}\,.$$

Let Ω^f and Ω^s be the regions occupied by the fluid and the solid at the final deformed configuration, respectively. In the stress-free configuration the fluid and solid region are defined by $\widehat{\Omega}^f$ and $\widehat{\Omega}^s$. Let $\Gamma^i = \Omega^f \cap \Omega^s$ and $\widehat{\Gamma}^i = \widehat{\Omega}^f \cap \widehat{\Omega}^s$ be the interface between solid and fluid in the final and undeformed configuration, respectively. The deformation of the domain $\Omega = \Omega^f \cup \Omega^s$ can be described by considering the two applications

$$\mathcal{X}^s : \widehat{\Omega}^s \to \mathbb{R}^3\,,$$
$$\mathcal{X}^f : \widehat{\Omega}^f \to \mathbb{R}^3\,,$$

such that $\mathrm{Im}(\mathcal{X}^s(\cdot)) = \Omega^s$, $\mathrm{Im}(\mathcal{X}^f(\cdot)) = \Omega^f$. The application $\mathcal{X}$ maps the position of any material point $\widehat{\mathbf{x}}$ from the given fixed reference configuration $\widehat{\Omega}$ to the deformed configuration Ω. The displacement field $\mathbf{d} : \widehat{\Omega} \to \mathbb{R}^3$ is defined as

$$\mathbf{d}(\widehat{\mathbf{x}}) = \mathcal{X}(\widehat{\mathbf{x}}) - \widehat{\mathbf{x}} = \mathbf{x} - \widehat{\mathbf{x}}\,, \tag{1}$$

$$\mathbf{d} = \begin{cases} \mathbf{d}^s & \text{on} \quad \widehat{\Omega}^s \\ \mathbf{d}^f & \text{on} \quad \widehat{\Omega}^f \end{cases}. \tag{2}$$

In the solid domain $\mathbf{d}^s$ is the actual solid displacement. In the fluid domain $\mathbf{d}^f$ is an artificial displacement with no physical meaning that is used to map the deformation of the fluid domain which occurs because of the deformation of the solid domain. The displacement $\mathbf{d}$ is an unknown of the FSI problem, and its knowledge also defines the unknown steady state configuration $\Omega = \Omega(\mathbf{d})$, where $\mathbf{x} \in \Omega$ is given by

$$\mathbf{x} = \widehat{\mathbf{x}} + \mathbf{d}(\widehat{\mathbf{x}}). \tag{3}$$

The velocity field $u : \widehat{\Omega} \to \mathbb{R}^3$

$$\mathbf{u} = \begin{cases} \mathbf{u}^s & \text{on} \quad \Omega^s \\ \mathbf{u}^f & \text{on} \quad \Omega^f \end{cases} \tag{4}$$

represents the fluid flow velocity $\mathbf{u}^f$ in the fluid domain and the solid velocity $\mathbf{u}^s$ in the solid domain. Since we only consider steady-state solutions we assume $\mathbf{u}^s = \mathbf{0}$ everywhere in the solid domain, including on the solid-fluid interface. Both the displacement $\mathbf{d}$ and the velocity $\mathbf{u}$ are continuous across the interface and are treated as monolithic fields on the whole domain.

In the following we will use the notations $\widehat{\nabla}$ or ∇ to refer to the del operator in the reference or in the deformed configuration, respectively.

The steady state momentum and mass balance equations for the solid and the fluid can be written as

$$\nabla \cdot \boldsymbol{\sigma}^s + \mathbf{f}^s = 0 \qquad \text{in } \Omega^s(\mathbf{d})\,, \tag{5}$$
$$-\rho(\mathbf{u} \cdot \nabla \mathbf{u}) + \nabla \cdot \boldsymbol{\sigma}^f + \mathbf{f}^f = 0 \qquad \text{in } \Omega^f(\mathbf{d})\,, \tag{6}$$
$$J - 1 = 0 \qquad \text{in } \widehat{\Omega}^s\,, \tag{7}$$
$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega^f(\mathbf{d})\,. \tag{8}$$

Here, ρ is the fluid density, $\mathbf{f}^s$ and $\mathbf{f}^f$ are the body forces in the solid and fluid domain, respectively. In Eq.(7)

$$J = \det \mathbf{F},$$

where $\mathbf{F}$ is the deformation gradient tensor, $\mathbf{F} = I + \widehat{\nabla}\mathbf{d}$. Notice that the gradient is taken with respect to the reference configuration. The symbol $\boldsymbol{\sigma}$ denotes the stress tensor (or Cauchy stress tensor in the solid). For incompressible fluid flows the stress tensor is given by

$$\boldsymbol{\sigma}^f(\mathbf{u}, p) = -p\boldsymbol{I} + \mu \nabla \mathbf{u} \ \text{ in } \ \Omega^f, \tag{9}$$

where μ is the fluid viscosity. For the solid we consider either the case of an incompressible Neo-Hookean material

$$\boldsymbol{\sigma}^s(\mathbf{d}, p) = -p\mathbf{I} + 2C_1\mathbf{B} \ \text{ in } \ \Omega^s\,, \tag{10}$$

or the case of an incompressible Mooney-Rivlin material

$$\boldsymbol{\sigma}^s(\mathbf{d}, p) = -p\mathbf{I} + 2C_1\mathbf{B} - 2C_2\mathbf{B}^{-1} \ \text{ in } \ \Omega^s\,, \tag{11}$$

where the constants C_1 and C_2 depend on the mechanical properties of the material and

$$\mathbf{B} = \mathbf{F}\mathbf{F}^T$$

is the left Cauchy-Green deformation tensor. The Neo-Hookean and Mooney-Rivlin hyperelastic models can be used for large strains. In Eqs. (6), (10) and (11) p is the pressure field and has to be computed. In the fluid domain p^f is the actual fluid pressure, while in the solid domain p^s does not have a clear physical meaning and can be regarded mathematically as the Lagrange multiplier associated to the incompressibility constraint (7). Similar mathematical interpretation can also be given to the fluid pressure p^f with respect to the incompressible constraint (8). Although we use a monolithic formulation also for the pressure p, we do not require pressure continuity across the solid-fluid interface.

We remark that the solid and fluid momentum equations are written in the final deformed configuration Ω, as well as the fluid divergence-free condition. The mass conservation for the solid is instead written in the stress-free configuration.

On the solid fluid interface we also enforce the stress continuity condition

$$\boldsymbol{\sigma}^f \cdot \mathbf{n} = \boldsymbol{\sigma}^s \cdot \mathbf{n} \qquad\qquad \text{on } \Gamma^i(\mathbf{d}), \tag{12}$$

where $\mathbf{n}$ is the unit normal vector on the deformed interface $\Gamma^i(\mathbf{d})$ whose direction has been fixed either from the fluid to the solid or from the solid to the fluid. No other interface condition is needed for the velocity and the displacement since, as we already pointed out, these are monolithic fields and their continuity is automatically satisfied.

Since these equations are posed on an unknown domain, their linearization will involve a domain update. The kinematic conditions for the movement of the domain are chosen to satisfy

$$\mathbf{0} = \mathbf{u} \qquad\qquad \text{in } \widehat{\Omega}^s\,, \tag{13}$$

$$\mathbf{0} = \widehat{\nabla} \cdot (k(\widehat{\mathbf{x}})\widehat{\nabla}\mathbf{d}) \qquad\qquad \text{in } \widehat{\Omega}^f\,, \tag{14}$$

where the role of the function $k(\widehat{\mathbf{x}})$ is described below. The mesh deformation is dealt based on the sizes of the elements. Following the work in [15] and [17] we want smaller elements to be stiffer than larger ones. In regions where the mesh would undergo large distortions (e.g. the region near the fluid-solid interface) we refine and build smaller and stiffer elements, in order not to degrade the mesh quality. This is easily achieved setting the function $k(\widehat{\mathbf{x}})$ to be a piecewise-constant function discontinuous across the element boundary and whose value is given by

$$k(\widehat{\mathbf{x}}) = \frac{1}{\mathrm{V}_{el}},$$

where V_{el} is the volume of the mesh element that contains the $\widehat{\mathbf{x}}$ coordinate.

2.2 Weak formulation

In this part of the paper we focus on describing how the weak formulation of the FSI problem is achieved using a monolithic approach. In this section we do not put too much emphasis on the external boundary conditions, rather we assume that they are given, are physically meaningful and compatible with all our assumptions. Different types of boundary conditions require us to modify the Sobolev spaces on which the solution and the shape functions are defined. Later in the numerical example part we will be more specific in choosing and describing the boundary conditions. For the sake of simplicity we assume here physically compatible Dirichlet boundary data everywhere.

Let the domain be divided as $\Omega = \Omega_s \cup \Omega_{sf} \cup \Omega_f$. Here, Ω_s and Ω_f denote the solid and fluid parts respectively, while Ω_{sf} is the union of the supports of the shape functions associated to the interface degrees of freedom. We will also define $\Omega_{sf} = \Omega_{s,i} \cup \Omega_{f,i}$, with $\Omega_{s,i} \cap \Omega_{f,i} = \emptyset$. Namely, we assume the interface Γ^i to be on the boundary of Ω_s and Ω_f and contained inside Ω_{sf}.

Let $\mathbf{d}, \mathbf{u}$ and p denote the displacement, velocity and pressure, respectively. After multiplication by appropriate test functions and integration by parts, the weak problem of the coupled FSI reduces in finding $\mathbf{d}, \mathbf{u}, p$ in $\boldsymbol{H}^1(\Omega) \times \boldsymbol{H}^1(\Omega) \times L^2(\Omega)$ such that

$$\int_{\Omega_s(\mathbf{d})} \sigma(\mathbf{d},p) : \nabla \boldsymbol{\phi}^{\mathbf{d}} - \mathbf{f}^s \cdot \boldsymbol{\phi}^{\mathbf{d}} = 0, \qquad \forall\, \boldsymbol{\phi}^{\mathbf{d}} \in \boldsymbol{H}_0^1(\Omega_s(\mathbf{d})), \tag{15}$$

$$\int_{\Omega_{s,i}(\mathbf{d})} \sigma(\mathbf{d},p) : \nabla \boldsymbol{\phi}^{\mathbf{d}} - \mathbf{f}^s \cdot \boldsymbol{\phi}^{\mathbf{d}} + \int_{\Omega_{f,i}(\mathbf{d})} \rho(\mathbf{u}\cdot\nabla\mathbf{u}) \cdot \boldsymbol{\phi}^{\mathbf{d}} + \sigma(\mathbf{u},p) : \nabla \boldsymbol{\phi}^{\mathbf{d}} - \mathbf{f}^f \cdot \boldsymbol{\phi}^{\mathbf{d}} = 0, \quad \forall\, \boldsymbol{\phi}^{\mathbf{d}} \in \boldsymbol{H}_0^1(\Omega_{sf}(\mathbf{d})), \tag{16}$$

$$\int_{\widehat{\Omega}_f} k(x) \nabla \mathbf{d} \cdot \nabla \boldsymbol{\phi}^{\mathbf{d}} = 0, \qquad \forall\, \boldsymbol{\phi}^{\mathbf{d}} \in \boldsymbol{H}_0^1(\widehat{\Omega}_f), \tag{17}$$

$$\int_{\widehat{\Omega}_s} \mathbf{u} \cdot \boldsymbol{\phi}^{\mathbf{u}} = 0, \qquad \forall\, \boldsymbol{\phi}^{\mathbf{u}} \in \boldsymbol{H}^1(\widehat{\Omega}_s), \tag{18}$$

$$\int_{\Omega_f(\mathbf{d})} \rho(\mathbf{u}\cdot\nabla\mathbf{u}) \cdot \boldsymbol{\phi}^{\mathbf{u}} + \sigma(\mathbf{u},p) : \nabla \boldsymbol{\phi}^{\mathbf{u}} - \mathbf{f}^f \cdot \boldsymbol{\phi}^{\mathbf{u}} = 0, \qquad \forall\, \boldsymbol{\phi}^{\mathbf{u}} \in \boldsymbol{H}_0^1(\Omega_f(\mathbf{d})), \tag{19}$$

$$\int_{\widehat{\Omega}_s} (J-1)\phi^p = 0, \qquad \forall\, \phi^p \in L^2(\widehat{\Omega}_s), \tag{20}$$

$$\int_{\Omega_f(\mathbf{d})} \phi^p \nabla \cdot \mathbf{u} = 0, \qquad \forall\, \phi^p \in L^2(\Omega_f(\mathbf{d})), \tag{21}$$

with appropriate boundary conditions on Ω. Notice that Eq. (15) represents the momentum balance inside the solid domain, while equation Eq. (19) represents the momentum balance inside the fluid domain. Eq. (16) satisfies the momentum balance on the interface region between the fluid and the solid domain. The test functions in Eq. (15) and Eq. (19) are on $\boldsymbol{H}_0^1(\Omega_s(\mathbf{d}))$ and $\boldsymbol{H}_0^1(\Omega_f(\mathbf{d}))$, respectively, thus the two sets of equations are defined in two disjoint subdomains. The momentum exchange between these regions occurs through Eq. (16) where the test functions are defined on $\boldsymbol{H}_0^1(\Omega_{sf}(\mathbf{d}))$, thus on a domain which overlaps with both the solid and fluid parts. It is worth noting that, by the coupling conditions (12), the two boundary terms that result from integration by parts at the solid-fluid interface Γ^i cancel out. This assures that stresses at the interface are correctly balanced.

Eq. (17) is the auxiliary displacement equation inside the fluid domain. Notice that the test functions are in $\boldsymbol{H}_0^1(\widehat{\Omega}_f)$. In other words the solution of this equation yields a smooth displacement field inside the fluid domain for given values on the boundary of $\widehat{\Omega}_f$. Thus, this equation does not affect directly the value of the displacement on the solid-fluid interface, which is an unknown of the problem that is evaluated by solving the other parts of the system.

Eq. (18) constrains the solid velocity to be equal to zero both inside the solid region and on its boundary, including the solid-fluid interface $\widehat{\Gamma}^i$, since the test functions are taken in $\boldsymbol{H}^1(\widehat{\Omega}_s)$.

Eqs. (20) and (21) represent the mass continuity constraints in the solid and fluid domains, respectively.

An idea of the variable dependence pattern for the equations in the previous system is given in Figure 2.2. Notice that in the figure we shaded in black the variables that appear explicitly in the integrands and in gray the variables that appear indirectly because the integrals are evaluated on the deformed domain $\Omega(\mathbf{d})$. For the sake of clarity we split the solution variables in the solid and fluid parts by adding the corresponding superscripts ^{s} and ^{f}, but we emphasize that the formulation is monolithic and the solution variables are defined on the whole domain Ω.

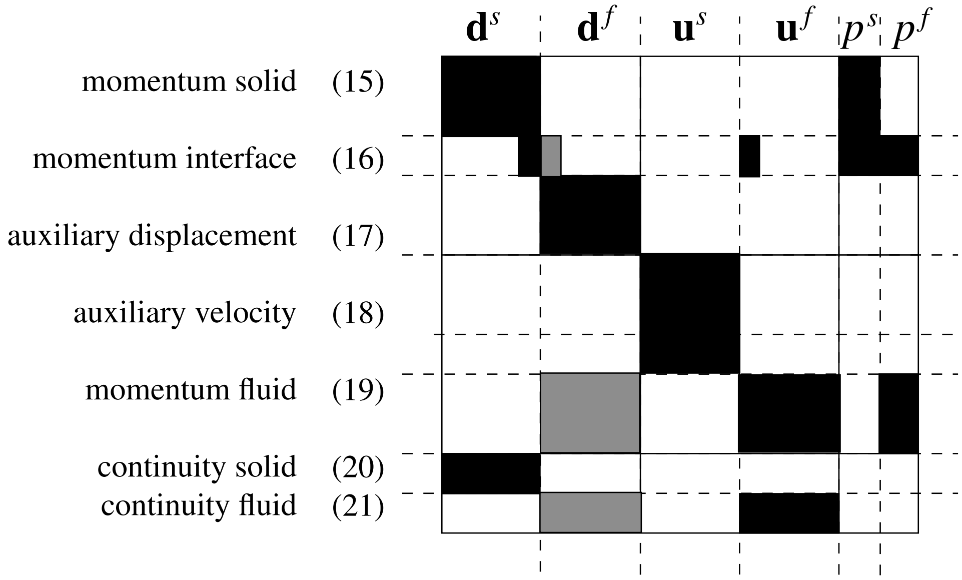

Figure 1: Variable dependence pattern associated to system (15)-(21).

It is worth nothing that the Jacobian matrix associated to equation system (15)-(21) has exactly the same sparsity pattern as in Figure 2.2. The accurate choice of the equation ordering (and corresponding shape function choice) optimizes the diagonal structure of the Jacobian matrix and is crucial for achieving convergence of the proposed algorithm.

2.3 Automatic differentiation for the evaluation of the tangent operator

The nonlinear system (15)-(21) is solved by using a Newton scheme. Let $\mathbf{v} = (\mathbf{d}, \mathbf{u}, p)$ and rewrite the equation system (15)-(21) in compact notation as

$$\mathcal{F}(\mathbf{v}) = 0. \tag{22}$$

Let $\mathbf{v}^0$ be an initial guess. Then a single Newton iteration is given by

$$\mathbf{v}^i = \mathbf{v}^{i-1} - \mathcal{J}^{-1}(\mathbf{v}^{i-1})\mathcal{F}(\mathbf{v}^{i-1}) \quad \text{for} \quad i \geq 1, \tag{23}$$

and it is repeated till the difference in norm between two successive solutions $\|\mathbf{v}^i - \mathbf{v}^{i-1}\|$ drops below a given tolerance value. Here, $\mathcal{J}(\mathbf{v})$ is the tangent operator (or Jacobian) of $\mathcal{F}$ in $\mathbf{v}$ and it is evaluated as

$$\mathcal{J}(\mathbf{v}^i) = \frac{\partial \mathcal{F}}{\partial \mathbf{v}}(\mathbf{v}^i).$$

This basic iteration exhibits quadratic convergence provided that the initial guess is sufficiently close to the solution. As already pointed out the Jacobian exhibits the same dependence variable pattern as the original system $(15) - (21)$ as described in Figure 2.2.

For the considered problem the symbolic differentiation of $\mathcal{J}$ is cumbersome, due to the presence of the integrals evaluated on the moving domain $\Omega(\mathbf{d})$. An approximate evaluation of $\mathcal{J}$ can be done either by neglecting some of the dependencies or by using approximate differentiation perturbing each component of $\mathbf{v}$ in $\mathcal{F}(\mathbf{v})$. In the current work the authors have instead computed the exact Jacobian using fast reverse automatic differentiation as described in [10]. Automatic differentiation is a set of techniques that numerically evaluate the derivative of a function specified in a computer program. It exploits the fact that every computer program executes a sequence of elementary arithmetic operations (addition, subtraction, multiplication, division, etc.) and elementary functions (exp, log, sin, cos, etc.). By applying the chain rule repeatedly to these operations derivatives of arbitrary order can be evaluated. Among the several libraries available we have chosen Adept, a software library that enables algorithms written in C and C++ to be automatically differentiated using operator overloading strategy.

It is worth noticing that in almost all the other works available in literature the authors prefer to rewrite all the equations in (15)-(21) in the fixed domain $\widehat{\Omega}$ using Kirchhoff-Piola tensor multiplication and Jacobian transformation, in order to avoid differentiation in moving domains. This approach greatly complicates the formulation of the problem, but it is preferred because once the equations are written in the fixed domain then the exact Jacobian $\mathcal{J}$ can be evaluated using symbolic differentiation (however, it still remains a challenging task in both derivation and numerical implementation). In our algorithm such an approach is not needed at all, and we choose to rewrite each equation in the domain configuration that is most convenient and to use automatic differentiation for the evaluation of the Jacobian. This approach results in high clarity for the formulation, as well as exactness and ease of use in the numerical implementation.

2.4 Multigrid discretization and inter-grid operators

We hereby describe the general ideas that lie behind the multigrid algorithm used to discretize and solve the FSI system for the variables $\mathbf{d}$, $\mathbf{u}$, and p. By starting at the coarse level $l = 0$, we discretize the entire domain $\widehat{\Omega}$ into a collection of finite elements, with the constraint that any point on the solid-fluid interface $\widehat{\Gamma}^i$ is located on the boundary between two elements. A geometrically conforming coarse triangulation $\mathcal{T}_h^0$ is then generated not only on the entire domain $\widehat{\Omega}$, but also on the subdomains $\widehat{\Omega}_s$, $\widehat{\Omega}_f$, and $\widehat{\Omega}_{sf}^0$. Recall that we defined $\widehat{\Omega}_{sf}$ to be the union of the supports of all the test functions associated to the interface degrees of freedom. Therefore its extension depends on the triangulation and is given by the collection of all the finite elements adjacent to the interface $\widehat{\Gamma}^i$. For this reason we added the apex notation, $\widehat{\Omega}_{sf}^0$, indicating the level mesh where it is defined.

Based on a simple element midpoint refinement, successive level meshes $\mathcal{T}_h^l$ are built recursively up to the top level $l = n$. Every triangulation $\mathcal{T}_h^l$ is geometrically conforming within all domains $\widehat{\Omega}$, $\widehat{\Omega}_s$, $\widehat{\Omega}_f$ and $\widehat{\Omega}_{sf}^l$. The nodes of the coarser mesh are always included in the nodes of the finer mesh. This is the key point for this kind of discretization and it is always true for different level meshes generated by using midpoint refinement.

Let $\mathbf{V}^l(\widehat{\Omega}) \subset \mathbf{H}^1(\widehat{\Omega})$ be a finite-element space built on the triangulation $\mathcal{T}_h^l$, and let $\mathbf{V}^l(\widehat{\Omega}_s)$, $\mathbf{V}^l(\widehat{\Omega}_f)$, and $\mathbf{V}^l(\widehat{\Omega}_{sf}^l)$ be the corresponding spaces obtained by the restriction of $\mathbf{V}^l(\widehat{\Omega})$ to the subdomains $\widehat{\Omega}_s$, $\widehat{\Omega}_f$, and $\widehat{\Omega}_{sf}^l$, respectively. Then the following nested relations between the spaces hold

$$\mathbf{V}^0(\widehat{\Omega}) \subset \mathbf{V}^1(\widehat{\Omega}) \subset \cdots \subset \mathbf{V}^n(\widehat{\Omega}), \tag{24}$$

$$\mathbf{V}^0(\widehat{\Omega}_s) \subset \mathbf{V}^1(\widehat{\Omega}_s) \subset \cdots \subset \mathbf{V}^n(\widehat{\Omega}_s), \tag{25}$$

$$\mathbf{V}^0(\widehat{\Omega}_f) \subset \mathbf{V}^1(\widehat{\Omega}_f) \subset \cdots \subset \mathbf{V}^n(\widehat{\Omega}_f), \tag{26}$$

but

$$\mathbf{V}^0(\widehat{\Omega}_{sf}^0) \not\subset \mathbf{V}^1(\widehat{\Omega}_{sf}^1) \not\subset \cdots \not\subset \mathbf{V}^n(\widehat{\Omega}_{sf}^n), \tag{27}$$

since the domain Ω_{sf}^l reduces in size as l increases.

Similarly, let $X^l(\widehat{\Omega}) \subset L^2(\widehat{\Omega})$ be a finite-element space built on the triangulation $\mathcal{T}_h^l$, and let $X^l(\widehat{\Omega}_s)$ and $X^l(\widehat{\Omega}_f)$, be the corresponding spaces obtained by restriction of $X^l(\widehat{\Omega})$ to the subdomains $\widehat{\Omega}_s$, and $\widehat{\Omega}_f$, respectively. Then the following nested relations between the spaces hold

$$X^0(\widehat{\Omega}) \subset X^1(\widehat{\Omega}) \subset \cdots \subset X^n(\widehat{\Omega}), \tag{28}$$

$$X^0(\widehat{\Omega}_s) \subset X^1(\widehat{\Omega}_s) \subset \cdots \subset X^n(\widehat{\Omega}_s), \tag{29}$$

$$X^0(\widehat{\Omega}_f) \subset X^1(\widehat{\Omega}_f) \subset \cdots \subset X^n(\widehat{\Omega}_f). \tag{30}$$

We solve for the FSI problem on the finest level spaces $\mathbf{V} = \mathbf{V}^n$ and $X = X^n$. The solution triplet $(\mathbf{d}, \mathbf{u}, p)$ is sought in the finite dimensional space $\mathbf{V} \times \mathbf{V} \times X$. We recall that the pair of spaces $\mathbf{V} \times X$, chosen for coupling either the solution variables $(\mathbf{d}, p)$ or $(\mathbf{u}, p)$ have to be taken appropriately in such a way that the discrete inf-sup condition is satisfied. This guarantees the stability of the approximation for both the incompressible hyperelastic material and the incompressible Navier-Stokes equation. We have chosen for $\mathbf{V}$ the space of biquadratic Lagrangian polynomials and for X the space of linear discontinuous polynomials, where the discontinuity occurs across the element boundaries. This last choice is well suited for the pressure field p since we do not enforce any continuity between the fluid and the solid pressure across the interface.

The coarser spaces are used to define and solve a Full Newton Geometric Multigrid Algorithm scheme, however extra attention should be given to define the inter level transfer operators between levels. The coarse-to-fine (prolongation) operator is classical and is taken to be the natural injection on the whole domain $\widehat{\Omega}$

$$I_{l-1}^l : \mathbf{V}^{l-1} \times \mathbf{V}^{l-1} \times X^{l-1}(\widehat{\Omega}) \longrightarrow \mathbf{V}^l \times \mathbf{V}^l \times X^l(\widehat{\Omega}), \tag{31}$$

given by

$$I_{l-1}^l \mathbf{v} = \mathbf{v}\,, \quad \text{for all} \quad \mathbf{v} \in \mathbf{V}^{l-1} \times \mathbf{V}^{l-1} \times X^{l-1}(\widehat{\Omega}). \tag{32}$$

The fine-to-coarse (restriction) operator is non classical, due both to the choice of the test function support and to the special ordering of the equations in the equation system (15)-(21). Rather than defining it in a formal mathematical setting (that will just over-complicate this discussion) we prefer to emphasize the major differences between a classical restriction operator and the current one.

When the multigrid algorithm is built on nested finite element spaces, the prolongation operator $\mathcal{I}_{l-1}^{l}$ is generally taken to be the natural injection between the two nested level spaces V^{l-1} and V^{l}, with $V^{l-1} \subset V^{l}$, as

$$\mathcal{I}_{l-1}^{l} : V^{l-1} \to V^{l} \ . \tag{33}$$

$$\mathcal{I}_{l-1}^{l} v = v, \qquad \text{for all } v \in V^{l-1}. \tag{34}$$

Then, the corresponding restriction operator is defined to be the transpose of $\mathcal{I}_{l-1}^{l}$ with respect to the inner product $(.,.)$. In other words

$$\mathcal{I}_{l}^{l-1} : V^{l} \to V^{l-1} \ , \tag{35}$$

$$(\mathcal{I}_{l}^{l-1} v, w) = (v, \mathcal{I}_{l-1}^{l} w) \qquad \text{for all } v \in V^{l} \text{ and } w \in V^{l-1}. \tag{36}$$

This is done under the underlying assumption that the set of partial differential equations associated to a particular test function is defined on the whole domain. In the equation system (15)-(21) this assumption is violated, since the equations have been fragmented, shuffled and defined piecewise on part of the domain. Moreover, in the interface momentum equation 16, the test functions in the target set have support Ω_{sf}^{l-1} larger than the support Ω_{sf}^{l} of the test functions in the original set. Therefore contributions from the other momentum equations are needed. To overcome these obstacles we build the restriction operator for each equation set, with special attention to the momentum interface equations. With reference to Figure 2.4, we have the following:

- the coarse solid momentum equations are obtained by restricting only the fine solid momentum equations;
- the coarse interface momentum equations are obtained by contributions not only from the interface but also from the solid and fluid momentum equations;
- the coarse auxiliary displacement equations are obtained by restricting only the fine auxiliary displacement equations on the fluid domain;
- the coarse auxiliary velocity equations are obtained by restricting the fine auxiliary velocity equations both on the solid domain and on the interface;
- the coarse fluid momentum equations are obtained by restricting only the fine fluid momentum equations;
- mass continuity equations are restricted in a standard way since the test functions associated to them are discontinuous and no interaction occurs between the two subdomains.

2.5 Domain decomposition for the multigrid algorithm

In the Newton-multigrid solver at each level we use a GMRES smoothing algorithm, where the tangent matrix is preconditioned by means of a domain decomposition strategy. In particular, we use an Additive Schwarz Method (ASM) with overlapping subdomains, where first we split the whole domain in the fluid and solid subregions, and then we further divide each subregion into smaller blocks. In each block the equations to be solved are taken following a Vanka-type strategy. The Vanka-type class of smoothers can be seen as a block Gauss-Seidel method (for details see [13, 1, 19, 24]). In each step several small linear systems are solved,

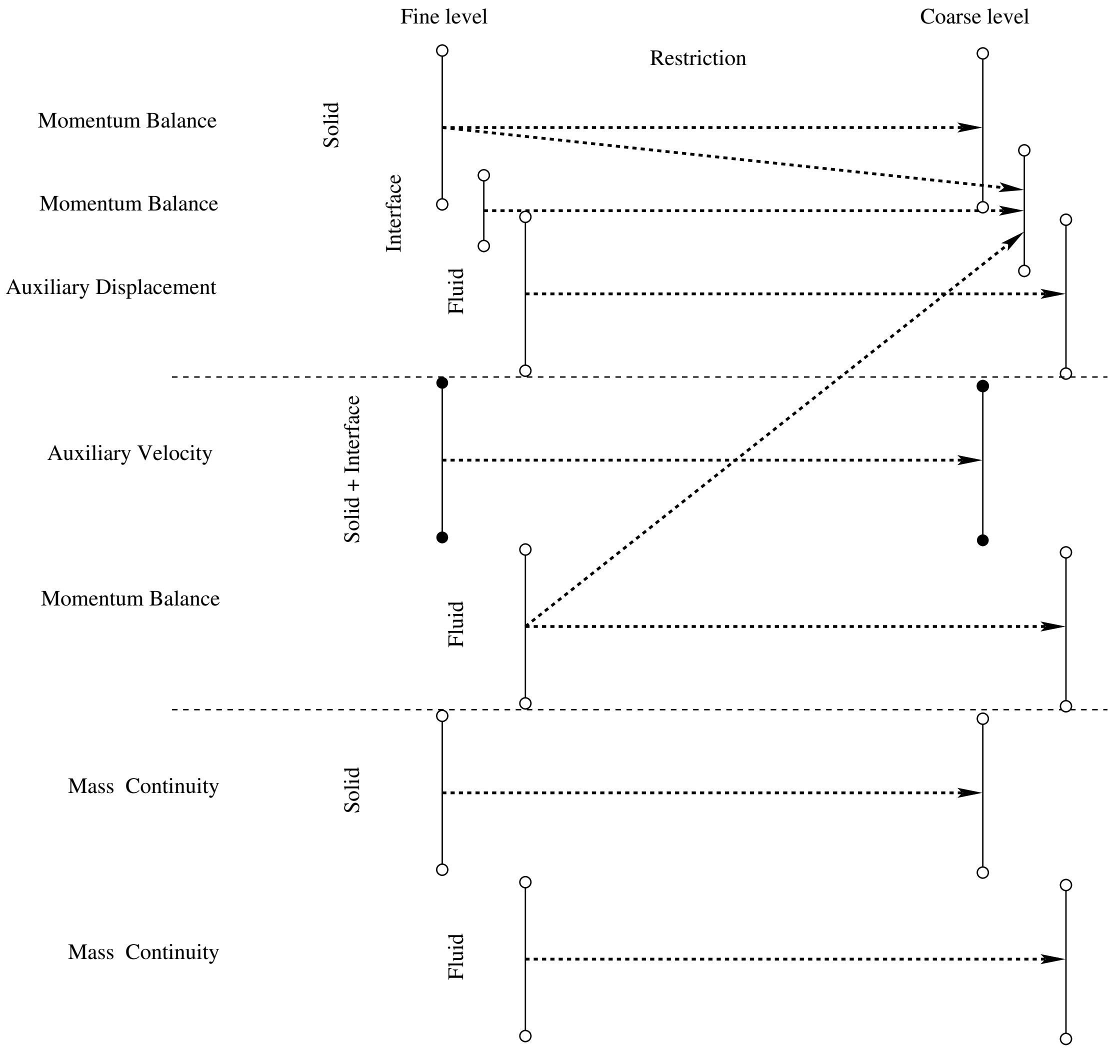

Figure 2: Restriction Scheme

each of which corresponds to all the degrees of freedom (DOFs) associated within the block of elements. Each subsystem is then solved using a direct or iterative solver. Notice that the DOFs associated to an element consists in displacement, velocity and pressure. While the support of the test functions associated to the pressure DOFs is limited to the element itself, the support of the test function associated to the displacement and velocity DOFs extends to the neighboring elements. This extension is responsible for the overlapping and the exchange of information between blocks.

3 NUMERICAL RESULTS

We seek to validate and evaluate the accuracy and performance of the proposed Newton-multigrid solver for a set of FSI benchmark configurations that can be found in the literature (see [3, 7, 23]).

3.1 Steady-state two-dimensional Hron-Turek FSI1 test

In [20] the authors proposed a benchmark for testing and comparing different numerical methods and code implementations for fluid-structure interaction problems. This benchmark is based on the older successful *flow around cylinder* setting developed in [21] for incompressible laminar flow. The overall setup of this interaction problem consists of an elastic solid object attached to a infinite rigid cylinder in a laminar channel flow. The fluid is supposed to be incompressible and the structure is allowed to be compressible or incompressible. The magnitude of the average inlet velocity is chosen in order to induce a steady flow and stationary deformations.

The computational domain is based on the 2D version of the well-known *flow around cylinder* benchmark and it is shown in Figure 3. The parameters which define the geometry are given as follows:

- the domain dimensions are: length $L = 2.5m$, height $H = 0.41m$;
- the circle center is positioned at $C = (0.2m, 0.2m)$ (measured from the left bottom corner of the channel) and the radius is $r = 0.05m$;
- the elastic structure bar has length $l = 0.35m$ and height $h = 0.02m$; the right bottom corner is positioned at $(0.6m, 0.19m)$, and the left end is fully attached to the fixed and rigid cylinder;
- the control point is A on the tip of the structure, with undeformed coordinates $A_0 = (0.6m, 0.2m)$.

The thickness and the length of the beam are chosen in order to reduce the bending stiffness without introducing additional numerical complications connected with high aspect ratios in the geometry. As reported in [21], the setting is intentionally non-symmetric to prevent the dependence of the onset of any possible oscillation on the precision of the computation.

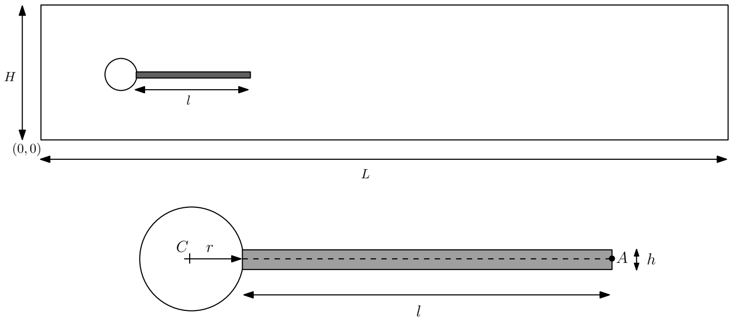

Figure 3: Computational domain and detail of the beam.

The FSI problem at hand needs to prescribe some boundary conditions.

- A parabolic velocity profile is prescribed in the left channel inflow section

$$\mathbf{v}^f(0, y) = 1.5U_m \frac{y(H-y)}{\left(\frac{H}{2}\right)^2} = 1.5U_m \frac{4.0}{0.1681} y(0.41 - y)\,. \tag{37}$$

- As outflow boundary condition the reference pressure at the outflow section is set to have zero mean value. This value has no influence on the solid deformation since fluid and solid are both incompressible. This is not true if the structure is supposed to be compressible.
- The *no-slip* condition is prescribed for the fluid on the other boundary edges. i.e. top and bottom wall, circle and fluid-structure interface.

We point out that the initialization of the nonlinear algorithm is performed by prescribing zero velocity in the fluid and no deformation in the structure.

The results of the benchmark computations are summarized in Table 2: $d_1(A)$ and $d_2(A)$ denote the displacements in $x-$ and $y-$direction of the control point A. The column "ndof" refers to the sum of unknowns for all velocity components, pressure, and displacement components.

Table 1: Parameter settings for the Hron-Turek FSI1 benchmark.

Parameter	symbol	measure unit	FSI1
Fluid density	ρ^f	$[10^3 \frac{Kg}{m^3}]$	1
Fluid viscosity	ν^f	$[10^{-3} \frac{m^2}{s}]$	1
Solid density	ρ^s	$[10^3 \frac{Kg}{m^3}]$	1
Poisson coefficient	ν^s	-	0.4
Shear modulus	μ^s	$[10^6 \frac{Kg}{ms^2}]$	0.5
Density ratio	β	-	1
Dimensionless shear modulus	Ae	-	3.5 $\times 10^4$
Average inlet velocity	U_m	$[\frac{m}{s}]$	0.2
Reynolds number	Re	-	20

All simulations have been performed with a fully implicit monolithic ALE-FEM method with a fully coupled multigrid solver as described in the previous chapter. For the validation of the employed fluid and solid solvers, we performed computations for different levels of spatial discretization (see Table 2).

The FSI test has been performed for an inflow speed resulting in a steady state solution. The parameter values of the test are given in the Table 1. This configuration of the benchmark has been named FSI1 by the authors who proposed it.

Table 2: Results for FSI1.

# of Levels	nel	ndof	$d_1(A)[\times 10^{-3}]$	$d_2(A)[\times 10^{-3}]$	drag	lift
1	792	15,640	0.02277155	0.81681181	14.29122	0.77623
2	3,168	61,376	0.02274210	0.81849791	14.28967	0.77512
3	12,672	243,136	0.02271325	0.81848982	14.28475	0.77456
4	50,688	967,808	0.02269713	0.81855286	14.28432	0.77424
ref.			0.0227	0.818	14.284	0.774

3.2 Steady-state vertical beam test

The second test example is based on a 2D classical benchmark proposed by COMSOL Multiphysics and consists of a narrow vertical structure attached to the bottom wall of a horizontal channel which bends under the force due to viscous drag and fluid pressure. As in the previous benchmark the fluid and solid are both incompressible.

The model geometry consists of a horizontal flow channel in the middle of which an obstacle, a narrow vertical structure, is placed. The fluid flows from left to right, except where the obstacle forces it to flow into a narrow path in the upper part of the channel, and it exerts a force onto the structure wall resulting from the viscous drag and fluid pressure. The deformable structure bends under the applied force thus modifying the path followed by the fluid flow. The parameters which define the geometry are shown in Figure 4:

- the channel dimensions are: length $L = 300\mu m$, height $H = 100\mu m$;
- the elastic vertical structure has length $l = 35\mu m$ and height $h = 5.5\mu m$; the left side is positioned at $d = 100\mu m$ away from the channel's left boundary and the bottom end is fully attached to the bottom wall of the channel; the top end of the beam is semicircular which radius is $r = 5\mu m$;
- the control point is A, attached to the top of the vertical narrow structure, with undeformed coordinates $A_0 = (105\mu m, 55\mu m)$.

As for the previous benchmark, the thickness and the length of the vertical structure are chosen in order to get a significant excursion of the beam without introducing additional numerical complications connected with high aspect ratios in the geometry.

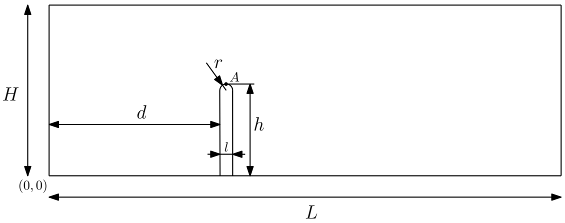

Figure 4: Computational domain.

The fluid flow in the channel is described by the incompressible Navier-Stokes equations for the velocity field $\mathbf{v}$ and the pressure p in the ALE (deformed) moving coordinate system. We assume that no gravitation or other volume forces affect the fluid motion. An incompressible water-like substance with a density $\rho^f = 1000Kg/m^3$ and dynamic viscosity $\mu^f = 0.001Pa \cdot s$ has been used as a fluid.

The structural deformations are solved by using a neo-Hookean hyperelastic material model and a nonlinear geometry formulation to allow large deformations. The structure consists of a narrow vertical flexible material with a density of $\rho^s = 7850Kg/m^3$, Young's modulus $E = 200kPa$ and Poisson coefficient $\nu^s = 0.50$.

At the channel entrance on the left, the flow has fully developed laminar with a parabolic velocity profile. At the outflow (right boundary), a do-nothing boundary condition is prescribed. On the solid (non-deforming) walls, no-slip conditions are imposed, $\mathbf{v} = \mathbf{0}$, while on the deforming interface the continuity of the velocity between fluid and solid is prescribed.

The obstacle is fixed to the bottom of the fluid channel. All other beam boundaries experience a load from the fluid, given by

$$\boldsymbol{\sigma}^s_{|\gamma_i} = (-p^f\mathbf{I} + \mu^f(\nabla\mathbf{v} + (\nabla\mathbf{v})^T)) \cdot \mathbf{n}\,,$$

where $\mathbf{n}$ is the normal vector to the boundary. This load represents the sum of pressure and viscous drag forces.

In this FSI benchmark we are interested in the computation of the displacement of a beam subject to a bending force in order to highlight the potentiality of the multigrid algorithm that has been presented in the previous chapter. The results of the benchmark computations are collected in Table 3. We report the number of refinement, the number of elements, the number of DOFs, the x and y displacements and the drag and lift forces.

The reference values are compared with the results obtained by solving the model using COMSOL Multiphysics.

Table 3: Results for COMSOL benchmark.

# of Levels	nel	ndof	$d_1(\mathbf{A})$	$d_2(\mathbf{A})$	drag	lift
2	1,088	21,188	$1.080789e^{-05}$	$-1.357274e^{-06}$	0.0094275	-0.395467
3	4,352	83,716	$1.081792e^{-05}$	$-1.359843e^{-06}$	0.0083106	-0.384255
4	17,408	332,804	$1.082601e^{-05}$	$-1.361546e^{-06}$	0.0050521	-0.384331
ref.			$1.0826e^{-05}$	$-1.361e^{-06}$	0.005	-0.384

4 CONCLUSIONS

In this work we presented a monolithic Newton-multigrid solver with domain decomposition smoothing for the solution of stationary incompressible FSI problems. This class of problems presents several challenging features due to the nonlinearity of the operators, the ill-conditioning of the matrices for steady-state cases and the enforcement of the incompressibility constraint. The Jacobian matrix in the Newton linearization of the FSI operator has been computed exactly by means of automatic differentiation tools. The numerical results are in agreement with benchmark solutions. Robust computations of FSI steady-state solutions can be performed with the proposed multigrid algorithm. No pseudo-time stepping needs to be introduced, which brings to a drastic reduction in computational time.

REFERENCES

[1] E. Aulisa, S. Manservisi, and P. Seshaiyer. A computational multilevel approach for solving two-dimensional Navier–Stokes equations over non-matching grids. *Computer methods in applied mechanics and engineering*, 195(33):4604–4616, 2006.

[2] E. Aulisa, S. Manservisi, and P. Seshaiyer. A multilevel domain decomposition approach to solving coupled applications in computational fluid dynamics. *Int. J. Numer. Meth. Fluids*, 56(8):1139–1145, 2008.

[3] Klaus-Jürgen Bathe and Gustavo A Ledezma. Benchmark problems for incompressible fluid flows with structural interactions. *Computers & structures*, 85(11):628–644, 2007.

[4] S. Bnà. *Multilevel Domain Decomposition Algorithms for Monolithic Fluid-Structure Interaction Problems with Application to Haemodynamics*. PhD thesis, Alma Mater Studiorum Università di Bologna, 2014.

[5] S. Bnà, E. Aulisa, and S. Manservisi. A Multilevel Domain Decomposition Solver for Monolithic Fluid-Structure Interaction Problems. In *11th International Conference of Numerical Analysis and Applied Mathematics 2013: ICNAAM 2013*, volume 1558, pages 871–874. AIP, 2013.

[6] M.Á. Fernández and M. Moubachir. A Newton method using exact Jacobians for solving fluid-structure coupling. *Computers and Structures*, 83:127–142, 2005.

[7] Miguel Ángel Fernández and Marwan Moubachir. A newton method using exact jacobians for solving fluid–structure coupling. *Computers & Structures*, 83(2):127–142, 2005.

[8] MW Gee, U Küttler, and WA Wall. Truly monolithic algebraic multigrid for fluid–structure interaction. *International Journal for Numerical Methods in Engineering*, 85(8):987–1016, 2011.

[9] M. Heil. An Efficient Solver to the Fully Coupled Solution of Large-Displacement Fluid-Structure Interaction Problems. *Comput. Meth. Appl. Mech. Eng.*, 193:1–23, 2004.

[10] Robin J Hogan. Adept fast automatic differentiation library for c++: User guide.

[11] J. Hron and S. Turek. A monolithic fem/multigrid solver for ale formulation of fluid structure interaction with application in biomechanics. In H.-J. Bungartz and M. Schäfer, editors, *Fluid-Structure Interaction: Modeling, Simulation, Optimization*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 146–170. Springer, 2006.

[12] Jaroslav Hron and Stefan Turek. *A monolithic FEM/multigrid solver for an ALE formulation of fluid-structure interaction with applications in biomechanics*. Springer, 2006.

[13] S. Manservisi. Numerical analysis of vanka-type solvers for steady Stokes and Navier–Stokes flows. *SIAM J. Numer. Anal.*, 44(5):2025–2056, September 2006.

[14] M Razzaq, H Damanik, J Hron, A Ouazzi, and S Turek. Fem multigrid techniques for fluid–structure interaction with application to hemodynamics. *Applied Numerical Mathematics*, 62(9):1156–1170, 2012.

[15] PA Sackinger, PR Schunk, and RR Rao. A newton–raphson pseudo-solid domain mapping technique for free and moving boundary problems: a finite element implementation. *Journal of Computational Physics*, 125(1):83–103, 1996.

[16] Michael Schäfer, Marcus Heck, and Saim Yigit. An implicit partitioned method for the numerical simulation of fluid-structure interaction. In *Fluid-structure interaction*, pages 171–194. Springer, 2006.

[17] K Stein, T Tezduyar, and R Benney. Mesh moving techniques for fluid-structure interactions with large displacements. *Journal of Applied Mechanics*, 70(1):58–63, 2003.

[18] DC Sternel, M Schäfer, M Heck, and S Yigit. Efficiency and accuracy of fluid-structure interaction simulations using an implicit partitioned approach. *Computational mechanics*, 43(1):103–113, 2008.

[19] S. Turek. *Efficient solvers for incompressible flow problems: an algorithmic and computational approach*, volume 6. Springer, 1999.

[20] S. Turek and J. Hron. Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. In H.-J. Bungartz and M. Schäfer, editors, *Fluid-Structure Interaction: Modeling, Simulation, Optimization*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 371–385. Springer, 2006.

[21] S. Turek and M. Schäfer. Benchmark computations of laminar flow around a cylinder. In E. H. Hirschel, editor, *Flow Simulation with High-Performance Computers II*, volume 52 of *Notes on Numerical Fluid Mechanics*, pages 547–566. Vieweg, 1996.

[22] Stefan Turek, J Hron, M Madlik, M Razzaq, H Wobker, and JF Acker. *Numerical simulation and benchmarking of a monolithic multigrid solver for fluid-structure interaction problems with application to hemodynamics*. Springer, 2010.

[23] Stefan Turek and Jaroslav Hron. *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow*. Springer, 2006.

[24] S P Vanka. Block-implicit multigrid calculation of two-dimensional recirculating flows. *Comput. Methods Appl. Mech. Eng.*, 59(1):29–48, November 1986.

[25] Hilmar Wobker and Stefan Turek. Numerical studies of vanka-type smoothers in computational solid mechanics. *Advances in Applied Mathematics and Mechanics*, 1(1):29–55, 2009.

[26] Yuqi Wu and Xiao-Chuan Cai. A parallel monolithic domain decomposition method for blood flow simulations in 3d. In *Domain Decomposition Methods in Science and Engineering XX*, pages 671–678. Springer, 2013.