HIGH ORDER FLUID STRUCTURE INTERACTION IN 2D AND 3D APPLICATION TO BLOOD FLOW IN ARTERIES

V. CHABANNES, G. PENA, AND C. PRUD'HOMME

ABSTRACT. This paper addresses the numerical approximation of Fluid Structure Interaction (FSI) problems through the Arbitrary Lagrangian Eulerian (ALE) framework, high order methods and a Dirichlet-Newmann approach for the coupling. The paper is divided in two main parts. The first part concerns the discretization method for the FSI problem. We introduce an improved ALE map, capable of handling curved geometries in 2D and 3D in a unified manner, that is based on a local differential operator. We also propose a minimal Continuous Interior Penalty (CIP) stabilization term for the fluid discretization that accounts for a smaller computational effort, while stabilizing the flow regime. The second part is dedicated to validate our numerical strategy through a benchmark and some applications to blood flow in arteries.

1. INTRODUCTION

Over the last few years, we have been working on building a mathematical and computational framework for arbitrary order fluid-structure interaction, see [1, 2, 3, 4, 5, 6], in 2D and 3D including using simplicial meshes with a wide range of applications and in particular in bio-mechanics, e.g. blood flows in arteries. In this paper we present the progress made since our last publications [2, 1] as well as a brief overview of the framework we have built so far.

Our computational framework builds upon FEEL++ [6, 7] which allows for arbitrary order continuous and discontinuous Galerkin methods (finite element, spectral elements, ...) in 1D, 2D and 3D on simplices and hypercubes. The computational domain can also be high order, that is to say, the elements are curved and in our case described by a geometrical transformation — from a straight element — which is a polynomial of degree greater than one. These high order meshes can be generated by GMSH [8] — up to order five in 2D and order four in 3D. — High order approximations come at a cost both in terms of implementation and computational points of view. The former is addressed by a very generic framework based on modern C++ programming (meta-programming, expression templates, ...) and a language mimicing the mathematical language. The latter is addressed by a careful implementation and optimisation, see [6].

We propose to solve the Fluid-Structure Interaction (FSI) problem in the Arbitrary Lagrangian Eulerian (ALE) framework detailed in section 2 in equations (3)-(7). In [1] the authors have proposed a framework for high order (in space and time) fluid structure interaction in 2D using an efficient high order ALE map construction which is described in [2]. In this paper, we follow the same ALE framework. A fundamental ingredient in the ALE framework is the discrete ALE transformation that maps the reference configuration onto the computational domain at each timestep. The discrete ALE map proposed in [2] allows for an accurate description of the boundary of the computational domain, while inducing straight edges in the interior elements of the computational domain's mesh. However, this construction is based on the harmonic extension operator and Gordon-Hall transformations, see Gordon-Hall [9, 10]. The use of the harmonic extension can, if the mesh deformation is large enough, introduce invalid meshes in the computations, see Figure 3. As to the Gordon-Hall transformations, they make the extension of this ALE map intrincate to three dimensional domains. To overcome these difficulties, we propose in this paper to replace the stage based on Gordon-Hall transformations, by the solution of a local differential problem in each element in contact with the curved boundary and the harmonic extension by the Winslow smoother, see [11, 24]. The construction of this map shall be addressed in detail in section 3.

In the framework presented in [2], the fluid flow is stabilized with the Continuous Interior Penalty (see the references therein for more detail on this stabilization procedure) method (CIP) in the case of dominant convection terms. The CIP stabilization term aims at penalizing the jump of the gradient across the faces of the triangulation. In the 2D and 3D blood flow applications presented in the paper, see section 6, even though stabilization is not necessary due to the presence of dominant convective terms,

it is required to control the velocity at the inlet of the domain where Neumann boundary conditions are imposed. In fact we propose to add minimal stabilization contributions, see section 4.

The paper is organized in the following way: first, in section 2, we introduce some definitions and notations needed to present the construction of the ALE map and the FSI framework addressed later in sections 3 and 4 respectively. Section 5 is devoted to validating the numerical method presented with the benchmark [12]. Finally, we apply the methodology developed to two blood flow problems under realistic parameters, both in 2D and 3D, and using high order approximations.

2. Definitions and Notations

Let $\Omega \subset \mathbb{R}^d$, $d \geq 1$, denote a bounded connected domain and δ a discretisation parameter. We define $\hat{K} \subset \mathbb{R}^d$ ($d = 1, 2, 3$) a reference elementary convex, e.g. a simplex or a hypercube. We denote by $\mathcal{T}_\delta$ a finite collection of nonempty, disjoint open simplices or hypercubes $\mathcal{T}_\delta \equiv \mathcal{T}_{(h,k)} = \{K = \boldsymbol{\varphi}^{\text{geo}}_{K,k}(\hat{K})\}$ forming a partition of Ω such that $h = \max_{K\in\mathcal{T}_\delta} h_K$, with h_K denoting the diameter of the element $K \in \mathcal{T}_\delta$ and $\boldsymbol{\varphi}^{\text{geo}}_{K,k}$ is the polynomial of degree k that maps $\hat{K}$ to K which also called the geometric transformation. The partition $\mathcal{T}_\delta$ induces a discretization of Ω, Ω_δ, defined as the union of the closure of all elements in this partition. Note that if Ω is a polyhedral domain then $\Omega_\delta = \Omega$. We say that a hyperplanar closed subset F of $\overline{\Omega_\delta}$ is a mesh face if it has positive $(d-1)$-dimensional measure and if either there exist K_1, $K_2 \in \mathcal{T}_\delta$ such that $F = \partial K_1 \cap \partial K_2$ (and F is called an *internal face*) or there exists $K \in \mathcal{T}_\delta$ such that $F = \partial K \cap \partial\Omega_\delta$ (and F is called a *boundary face*). Internal faces are collected in the set $\mathcal{F}^i_\delta$, boundary faces in $\mathcal{F}^b_\delta$ and we let $\mathcal{F}_\delta := \mathcal{F}^i_\delta \cup \mathcal{F}^b_\delta$. For all $F \in \mathcal{F}_\delta$, we define $\mathcal{T}_F := \{K \in \mathcal{T}_\delta \mid F \subset \partial K\}$. For every interface $F \in \mathcal{F}^i_\delta$ we introduce two associated normals to the elements in $\mathcal{T}_F$ and we have $\mathrm{n}_{K_1,F} = -\mathrm{n}_{K_2,F}$, where $\mathrm{n}_{K_i,F}$, $i \in \{1, 2\}$, denotes the unit normal to F pointing out of $K_i \in \mathcal{T}_F$. On a boundary face $F \in \mathcal{F}^b_\delta$, $\mathrm{n}_F = \mathrm{n}_{K,F}$ denotes the unit normal pointing out of Ω_δ.

Without loss of generality we suppose from now on that we work with simplicial elements. Given a positive integer N, we denote by $\mathbb{P}^N(\hat{K})$ and $\mathbb{P}^N(K)$ the spaces of polynomials of total degree less or equal than N defined in $\hat{K}$ and K respectively. We shall refer to Ω_δ as the *computational* domain and we introduce Ω^*_δ, which will be designated as the *reference* domain. We assume that the reference domain has a straight edge/face mesh associated i.e. $\mathcal{T}^*_\delta \equiv \mathcal{T}^*_{(h,1)}$. Furthermore we admit that the mesh $\mathcal{T}^*_\delta$ covers exactly the domain Ω^*_δ, *i.e.*, $\Omega^*_\delta = \bigcup_{K^*\in\mathcal{T}^*_\delta} \overline{K^*}$. We denote $\mathcal{A}_\delta \equiv \mathcal{A}_{(h,k)}$ a transformation that maps Ω^*_δ to Ω_δ which we later call the ALE discrete map.

We define $P^N_c(\Omega^*_\delta \equiv \Omega^*_{(h,1)})$ and $[P^N_c(\Omega^*_\delta \equiv \Omega^*_{(h,1)})]^d$ the spaces of continuous scalar and vectorial functions respectively on Ω^*_δ as follows:

$$P^N_c(\Omega^*_\delta) = \{v \in C^0(\Omega^*_\delta) \mid v \circ \boldsymbol{\varphi}^{\text{geo}}_{K,1} \in \mathbb{P}^N(\hat{K})\ \forall K \in \mathcal{T}^*_\delta\}, \quad [P^N_c(\Omega^*_\delta)]^d = \prod_1^d P^N_c(\Omega^*_\delta). \tag{1}$$

We define similarly $P^N_c(\Omega_\delta \equiv \Omega_{(h,k)})$ and $[P^N_c(\Omega_\delta \equiv \Omega_{(h,k)})]^d$ with $k \geqslant 1$:

$$P^N_c(\Omega_\delta) = \{v \in C^0(\Omega_\delta) \mid v \circ \boldsymbol{\varphi}^{\text{geo}}_{K,k} \in \mathbb{P}^N(\hat{K})\ \forall K \in \mathcal{T}_\delta\}, \quad [P^N_c(\Omega_\delta)]^d = \prod_1^d P^N_c(\Omega_\delta). \tag{2}$$

Finally let us denote by $\boldsymbol{\eta} : \partial\Omega^*_\delta \longrightarrow \partial\Omega_\delta$ a displacement function. Through $\boldsymbol{\eta}$, we classify three subsets of the boundary: *(i)* Γ^*_M, the portion of the boundary that moves according to the displacement $\boldsymbol{\eta}$, *(ii)* Γ^*_F, the portion of the boundary that stays fixed (ie, $\boldsymbol{\eta}(\boldsymbol{s}) = \boldsymbol{s}, \forall \boldsymbol{s} \in \Gamma^*_F$) and *(iii)* Γ^*_N, the part of the boundary on which we do not prescribe a displacement. The image of each subset, Γ^*_M, Γ^*_F and Γ^*_N by $\boldsymbol{\eta}$ is denoted by Γ_M, Γ_F and Γ_N, respectively. These three sets do not overlap and they verify $\partial\Omega^*_\delta = \overline{\Gamma^*_M} \cup \overline{\Gamma^*_F} \cup \overline{\Gamma^*_N}$. Denote $\mathcal{T}^{*,b}_\delta = \{K^* \in \mathcal{T}^*_\delta : \partial K^* \cap \Gamma^*_M \neq \emptyset\}$ the set of elements K^* sharing a face with the boundary of Ω^*_δ.

Bearing in mind these notations, let us introduce the Fluid-Structure Interaction (FSI) problem we aim to solve, in the Arbitrary Lagrangian Eulerian (ALE) framework. It reads as: find $(\mathcal{A}_t, \boldsymbol{u}_f, p_f, \boldsymbol{\eta}_s)$

such that

$$\rho_f \left.\frac{\partial \mathbf{u}_f}{\partial t}\right|_{\mathbf{x}^*} - \mathbf{div}_{\mathbf{x}}(2\mu_f \mathbf{D}_{\mathbf{x}}(\mathbf{u}_f)) + \rho_f((\mathbf{u}_f - \mathbf{w}_f)\cdot \boldsymbol{\nabla}_{\mathbf{x}})\mathbf{u}_f + \nabla_{\mathbf{x}} p_f = 0, \quad \text{in } \Omega_f^t \times I \tag{3}$$

$$\mathrm{div}_{\mathbf{x}}(\mathbf{u}_f) = 0, \quad \text{in } \Omega_f^t \times I \tag{4}$$

$$\rho_s \frac{\partial^2 \boldsymbol{\eta}_s}{\partial t^2} - \nabla \cdot (\boldsymbol{F}_s \boldsymbol{\Sigma}_s) = 0, \quad \text{in } \Omega_s \times I \tag{5}$$

$$\mathbf{u}_f - \frac{\partial \boldsymbol{\eta_s}}{\partial t} \circ \mathcal{A}_t^{-1} = 0, \quad \text{on } \Gamma_{fsi}^t \tag{6}$$

$$\boldsymbol{F}_s \boldsymbol{\Sigma}_s \hat{\mathrm{n}}_s + J_{\mathcal{A}_t} \boldsymbol{F}_{\mathcal{A}_t}^{-T} \hat{\boldsymbol{\sigma}}_f \hat{\mathrm{n}}_f = 0, \quad \text{on } \Gamma_{fsi}^0 \tag{7}$$

where $I = (0, T]$, $\Omega_f^t = \mathcal{A}_t(\Omega_f^*)$ and Ω_s are the domains occupied by the fluid and structure, respectively, Ω_f^* is the ALE reference domain for the fluid, $\mathbf{x}^*$ and $\mathbf{x}$ denote the coordinate systems in Ω_f^* and Ω_f^t respectively, $\mathbf{u}_f$ and p_f the velocity and pressure of the fluid, ρ_f and ρ_s are the densities of the fluid and structure, respectively, μ_f is the viscosity of the fluid, $\left.\frac{\partial \mathbf{u}_f}{\partial t}\right|_{\mathbf{x}^*}$ is the ALE time derivative, $\mathbf{D}_{\mathbf{x}}(\mathbf{u}_f)$ the fluid deformation tensor, $\boldsymbol{\eta}_s$ the displacement of the structure, $\boldsymbol{F}_s = \mathbf{I} + \nabla \boldsymbol{\eta}_s$ is the structure deformation tensor, $\mathbf{I}$ is the identity tensor, $\boldsymbol{E}_s = \frac{1}{2}\left(\boldsymbol{F}_s \boldsymbol{F}_s^T - \mathbf{I}\right)$ the Green-Lagrange tensor, $\boldsymbol{\Sigma}_s = \lambda_s \left(\mathrm{tr}\, \boldsymbol{E}_s\right) \mathbf{I} + 2\mu_s \boldsymbol{E}_s$ represents the second Piola-Kirchoff stress tensor, Γ_{fsi}^t the interface boundary between the fluid and structure and $\mathcal{A}_t$ is the continuous ALE map. These notations allow to define the stress tensor $\hat{\boldsymbol{\sigma}}_f = \boldsymbol{\sigma}_f \circ \mathcal{A}_t$ and the normal vectors $\hat{\mathrm{n}}_f = \mathrm{n}_f \circ \mathcal{A}_t$ and $\hat{\mathrm{n}}_s = \mathrm{n}_s \circ \mathcal{A}_t$ in the reference domain, where n_f and n_s denote, respectively, the normal vector associated with the fluid and structure computational domains.

We define also $\boldsymbol{F}_{\mathcal{A}_t} = \nabla \mathcal{A}_t$ and $J_{\mathcal{A}_t} = \det\left(\boldsymbol{F}_{\mathcal{A}_t}\right)$. The problem is complemented with boundary conditions at the inflow Γ_{in} and outflow Γ_{out}. Since our motivation is driven by blood flow in arteries, we present in Figures 1 and 2 two simplified geometries that exemplify the domains and boundaries for the problem described above.

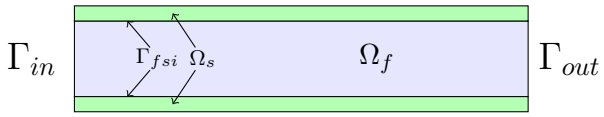


FIGURE 1. Two dimensional geometry

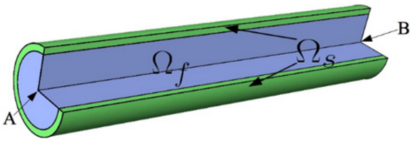
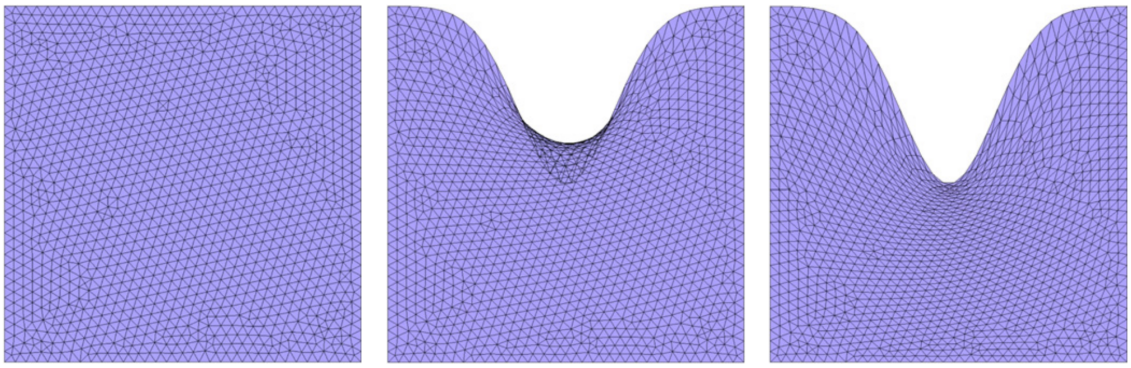


FIGURE 2. Three dimensional geometry

3. HIGH ORDER ALE TRANSFORMATION

In both papers [2, 1], the piecewise linear map is created is calculated by performing first a harmonic extension (or modified harmonic extension) of the boundary data. However, if the displacement is too large, these operators can induce meshes that are not valid due to, for instance, *mesh folding*. A way to circumvent this problem, that stems from the structure of the proposed ALE map construction, is to replace the harmonic extension by a more suitable and flexible operator that avoids these issues or improves the mesh quality. An example of such an operator is the Winslow smoother [11, 24]. From a continuous point of view, the Winslow smoother enforces that the inverse of the ALE map is harmonic, not the map itself. This accounts for solving a quasi-linear system of PDEs, which can be done using a fixed point method. Figure 3 displays the effect of the harmonic extension and the Winslow smoother for the same imposed displacement at the top of the unit square, see [18].

We suppose from now on that only the elements touching the boundary are curved while the elements in the interior are straight — the geometric transformation associated is affine. — The construction of the ALE map is then done in two steps: the first step is to use the Winslow smoother to extend the displacement $\boldsymbol{\eta}$ to the interior of the reference domain using piecewise linear polynomial functions and obtain the corresponding ALE transformation, $\mathcal{A}_{(h,1)}$. The second step is a correction performed in each element that touches the curved boundary in order to build a high order approximation. In each element



(a) Reference mesh (b) Harmonic ALE map (c) Winslow smoother ALE map

FIGURE 3. Comparison of first order meshes in 2D generated by the harmonic extension (invalid) and Winslow smoother (valid) operators respectively

$K^* \in \mathcal{T}^{*,b}_{(h,1)}$ we look for $\mathcal{A}_{(h_{K^*},N)} \in [\mathbb{P}^N(K^*)]^d$ such that

$$\begin{cases} \displaystyle\int_{K^*} \nabla \mathcal{A}_{(h_{K^*},N)} : \nabla \mathbf{z}\, dx = 0, & \forall \mathbf{z} \in [\mathbb{P}^N(K^*)]^d \\ \mathcal{A}_{(h_{K^*},N)}(\boldsymbol{x}^*) = \boldsymbol{\eta}(\boldsymbol{x}^*) + \boldsymbol{x}^* - \mathcal{A}_{(h,1)}(\boldsymbol{x}^*), & \forall \boldsymbol{x}^* \in \partial K^* \cap \Gamma^*_M \\ \mathcal{A}_{(h_{K^*},N)} = \mathbf{0}, & \text{elsewhere on } \partial K^*. \end{cases}$$

where Γ^*_M is the portion of boundary in the reference domain that is curved in the computational domain. The final ALE map, $\mathcal{A}_\delta \equiv \mathcal{A}_{(h,N)}$ is obtained by adding to $\mathcal{A}_{(h,1)}$ the correction $\mathcal{A}_{(h_{K^*},N)}$ on each element of $\mathcal{T}^{*,b}_{(h,1)}$

$$\mathcal{A}_{(h,N)}(\mathbf{x}^*) = \mathcal{A}_{(h,1)}(\mathbf{x}^*) + \sum_{K^* \in \mathcal{T}^{*,b}_{(h,1)}} \mathcal{A}_{(h_{K^*},N)}(\mathbf{x}^*).$$

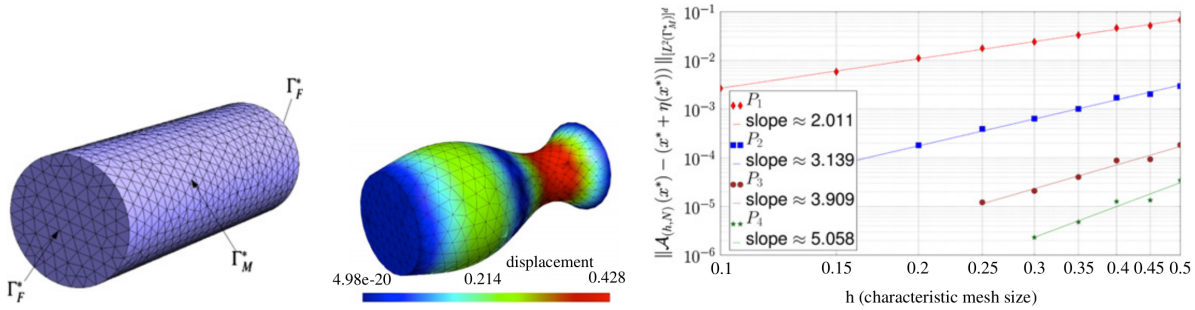
Proposition 1 (Properties of $\mathcal{A}_{(h,N)}$)**.** *Under the previous assumptions and if all reference domains coincide for all δ then $\mathcal{A}_{(h,N)} \in [P^N_c(\mathcal{T}^{*,b}_{(h,1)})]^d$ enjoys optimal approximation properties* (i.e. *if $\boldsymbol{\eta} \in \boldsymbol{H}^m(\partial\Omega^*_\delta)$ then the boundary approximation is $\mathcal{O}\left(h^{\min(N+1,m)}\right)$ in the L^2-norm) and $\mathcal{A}_{(h,N)} \in [P^1_c(\mathcal{T}^*_{(h,1)} \backslash \mathcal{T}^{*,b}_{(h,1)})]^d$.*

The proof of this Proposition can be found in [22] for $d = 2, 3$. The idea of the proof is to notice that the boundary approximation error can be written as a nodal interpolation error of the displacement function. Then, using a standard scaling argument, an error estimate can be obtained which will depend on the polynomial order N and the regularity of $\boldsymbol{\eta}$. The more general case in which the reference domains do not coincide is still open though we expect that a similar result holds. Indeed, in the following, we present some numerical experiments that point in that direction. Let us consider the reference domain depicted in Figure 4(a) defined by

$$\Omega^{*,cy} = \left\{ (x^*, y^*, z^*) \in \mathbb{R}^3 : \; x^* \in [0,5]\,, y^{*2} + z^{*2} \leq 0.5^2 \right\}$$

and the associated displacement of its boundary $\boldsymbol{\eta}^{cy}(\boldsymbol{x}^*) = 0.2 \exp\left(\dfrac{x^*}{5}\right) \sin\left(\dfrac{\pi x^*}{2.5}\right) \mathrm{n}^*$. Figure 4(b) displays the computational domains colored by the corresponding ALE map.

Finally, Figure 4(c) displays the convergence rate of the quantity $\|\mathcal{A}_{(h,N)}(\boldsymbol{x}^*) - (\boldsymbol{x}^* + \boldsymbol{\eta}(\boldsymbol{x}^*))\|_{[L^2(\Gamma^*_M)]^d}$ which confirms our belief that Proposition 1 is also valid in this case with respect to geometric order N.



(a) $\Omega^{*,cy}$ (b) $\Omega^{cy} = \mathcal{A}_{(h,4)}(\Omega^{*,cy})$ (c) Convergence rates

FIGURE 4. Reference (left) and computational (center) meshes of order 4 displayed using GMSH colored by the displacement's magnitude. Convergence rate plots of $\mathcal{A}_{(h,N)}$, $N = 1, 2, 3, 4$ with respect to h (right) .

4. DISCRETIZATION METHOD

We introduce now the numerical method used to solve the FSI problem. We use a Dirichlet-Newmann approach for the coupled FSI problem. This accounts for using a fixed point method, iterating between solves of the fluid equations and solves of the structure equations with appropriate Dirichlet and Newmann boundary conditions respectively. For the fluid solver, we use the same numerical methods as in [2] which we review very briefly in the next section as well as the structure solver afterwards.

4.1. Fluid discretization.

We start by introducing some notations. Let $\Omega^*_{f,(h,1)}$ denote the reference domain, constructed from the high order mesh (with geometrical transformations of degree k) of the domain occupied by the fluid at time t_0 by straightening all interior edges and faces (note that as a consequence, this domain is a polyhedron). We define $\Omega^t_{f,\delta} := \mathcal{A}^t_\delta(\Omega^*_{f,(h,1)})$ which is partitioned by a high order mesh. With these notations, the fluid problem reads as: for each $n \geqslant 1$, we look for the solution $(\mathbf{u}^{n+1}_\delta, p^{n+1}_\delta) \in [P^N_c(\Omega^t_{f,\delta})]^d \times P^{N-1}_c(\Omega^t_{f,\delta})$, with $\mathbf{u}^0_\delta = \mathbf{u}_{0,\delta}$ in $\Omega^{t_0}_{f,\delta}$, such that

$$
\begin{aligned}
\rho_f \frac{\beta_{-1}}{\Delta t}\left(\mathbf{u}^{n+1}_\delta, \mathbf{v}\right) + a\left(\mathbf{u}^{n+1}_\delta, \mathbf{v}\right) + s\left(\mathbf{u}^{n+1}_\delta, \mathbf{v}; \mathbf{u}^*_\delta\right) & \\
-b\left(\mathbf{v}, p^{n+1}_\delta\right) + c\left(\mathbf{u}^{n+1}_\delta, \mathbf{v}; \mathbf{u}^*_\delta - \mathbf{w}^{n+1}_\delta\right) &= \left(\tilde{\mathbf{f}}^{n+1}_\delta, \mathbf{v}\right), \quad \forall \mathbf{v} \in [P^N_c(\Omega^{t_{n+1}}_{f,\delta})]^d \\
b\left(\mathbf{u}^{n+1}_\delta, q\right) &= 0, \qquad \forall q \in P^{N-1}_c(\Omega^{t_{n+1}}_{f,\delta})
\end{aligned} \tag{8}
$$

where

$$
a\left(\mathbf{u}^n_\delta, \mathbf{v}\right) = \int_{\Omega^{t_n}_{f,\delta}} \mathbf{D}_{\mathrm{x}}(\mathbf{u}^n_\delta) : \boldsymbol{\nabla}\mathbf{v} dx, \quad b\left(\mathbf{v}, p^n_\delta\right) = \int_{\Omega^{t_n}_{f,\delta}} \mathrm{div}_{\mathbf{x}}(\mathbf{v})\, p^n_\delta \, dx
$$

$$
\left(\mathbf{u}^n_\delta, \mathbf{v}\right) = \int_{\Omega^{t_n}_{f,\delta}} \mathbf{u}^n_\delta \cdot \mathbf{v} dx, \quad c\left(\mathbf{u}^n_\delta, \mathbf{v}; \boldsymbol{\beta}\right) = \rho_f \int_{\Omega^{t_n}_{f,\delta}} \left[\boldsymbol{\beta} \cdot \boldsymbol{\nabla}_{\mathbf{x}}\right] \mathbf{u}^{\mathbf{n}}_\delta \cdot \mathbf{v} dx
$$

$$
s\left(\mathbf{u}^n_\delta, \mathbf{v}; \boldsymbol{\beta}\right) = \gamma \sum_{F \in \mathcal{F}} \int_F |\boldsymbol{\beta} \cdot \mathrm{n}| \frac{h^2_F}{N^{3.5}} [\![\boldsymbol{\nabla}\mathbf{u}^n_\delta]\!]_F \cdot [\![\boldsymbol{\nabla}\mathbf{v}]\!]_F \, ds,
$$

β_{-1} and $\tilde{\mathbf{f}}^{n+1}_\delta$ are determined according to the corresponding Backward Differentiation Formula (BDF), $\mathcal{F}$ denotes a set of faces and $[\![\cdot]\!]_F$ denotes the jump across the face F, $\mathbf{u}^*_\delta$ is an extrapolated velocity and $\mathbf{w}^{n+1}_\delta$ is the mesh velocity. See [2] for a complete description of all variables. At each iteration of the fixed point method, this problem is subject to Dirichlet boundary conditions in the moving interface of the domain and Neumann boundary conditions in the inlet and outlet (these are incorporated through the force term $\tilde{\mathbf{f}}^{n+1}_\delta$).

In [2] the authors proposed $\mathcal{F} = \mathcal{F}^i_h$, i.e., the use of all internal faces in the calculation of $s\left(\mathbf{u}^n_\delta, \mathbf{v}; \boldsymbol{\beta}\right)$. We propose two alternatives which reduce considerably the computational effort, while still stabilizing the flow regime in blood flow simulations, see section 6. They are (i) $\mathcal{F} = \Gamma_{in} \cap \mathcal{F}^b_h$ for only the faces at the inlet and (ii) $\mathcal{F} = \mathcal{F}^i_h \cap \left\{\mathbf{x} \in \mathbb{R}^d : (\mathbf{x}_{in} - \mathbf{x}, \mathrm{n}_{in}) \leqslant 0\right\}$, for a set of faces close to the inlet, where $\mathbf{x}_{in}$ is a suitable point inside the computational domain and n_{in} is the outward normal to Γ_{in}.

4.2. **Structure discretization.** We start this section by introducing the notation for the reference domain for the structure, $\Omega_{s,\delta}$, as well as the discrete space

$$V_{s,\delta}^N \equiv V_{s,(h,k)}^N = \left\{ \boldsymbol{\eta}_s \in [P_c^N(\Omega_{s,\delta} \equiv \Omega_{s,(h,k)})]^d, \quad \boldsymbol{\eta}_s = \mathbf{0} \text{ on } \Gamma_{in} \text{ and } \Gamma_{out} \right\}. \tag{9}$$

where $N \geqslant 1$ and $k \geqslant 1$. Given a displacement function $\boldsymbol{\eta}_s \in V_{s,\delta}^N$, we introduce the structure configuration at time t as $\Omega_s^t = (\mathbf{I} + \boldsymbol{\eta}_s)(\Omega_{s,\delta})$.

Using a Newmark scheme for the time discretisation at each time step t_n, we define the displacement $\boldsymbol{\eta}_{s,\delta}^n$ and its time derivatives $\dot{\boldsymbol{\eta}}_{s,\delta}^n$ and $\ddot{\boldsymbol{\eta}}_{s,\delta}^n$, all belonging to space $V_{s,\delta}^N$. Moreover, we enforce that the velocity and acceleration discretisations are functions of the previous time steps

$$\ddot{\boldsymbol{\eta}}_{s,\delta}^{n+1} = \frac{1}{\beta \Delta t^2}\left(\boldsymbol{\eta}_{s,\delta}^{n+1} - \boldsymbol{\eta}_{s,\delta}^{n}\right) - \frac{1}{\beta \Delta t}\dot{\boldsymbol{\eta}}_{s,\delta}^{n} - \left(\frac{1}{2\beta} - 1\right)\ddot{\boldsymbol{\eta}}_{s,h}^{n} \tag{10}$$

$$\dot{\boldsymbol{\eta}}_{s,\delta}^{n+1} = \frac{\gamma}{\beta \Delta t}\left(\boldsymbol{\eta}_{s,\delta}^{n+1} - \boldsymbol{\eta}_{s,\delta}^{n}\right) - \left(\frac{\gamma}{\beta} - 1\right)\dot{\boldsymbol{\eta}}_{s,\delta}^{n} - \Delta t\left(\frac{\gamma}{2\beta} - 1\right)\ddot{\boldsymbol{\eta}}_{s,\delta}^{n} \tag{11}$$

The two parameters γ and β are taken to be $\gamma = 0.5$ and $\beta = 0.25$ to ensure unconditional stability of the Newmark scheme, see [23] for more details. The discrete variational formulation leads to finding $\boldsymbol{\eta}_{s,\delta}^{n+1} \in V_{s,\delta}^N$ for all time step t_{n+1} such as:

$$\int_{\Omega_{s,\delta}} \rho_s \ddot{\boldsymbol{\eta}}_{s,\delta}^{n+1} \cdot \boldsymbol{\xi}_s dx + \int_{\Omega_{s,\delta}} \left(\boldsymbol{F}_{s,\delta}^{n+1}\boldsymbol{\Sigma}_{s,\delta}^{n+1}\right) : \nabla \boldsymbol{\xi}_s dx = \int_{\Gamma_{fsi}^0} \boldsymbol{g}_s \cdot \boldsymbol{\xi}_s ds, \quad \forall \boldsymbol{\xi}_s \in V_{s,\delta}^N \tag{12}$$

where $\boldsymbol{g}_s = J_{\mathcal{A},\delta}\boldsymbol{F}_{\mathcal{A},\delta}^{-T}\hat{\boldsymbol{\sigma}}_{f,\delta}\hat{\mathrm{n}}_f$ is the stress induced by the fluid.

4.3. **Solution strategy.** The algebraic systems arising from the discretization proposed in the previous section are solved using a Newton or quasi-Newton algorithm with a cubic line search method. At each step, the linear solver applies the GMRES method with a LU preconditioner. The preconditioner is typically built only once throughout the nonlinear iterations unless the nonlinearity is too stiff and the preconditioner needs to be recalculated during the non-linear iterations. In the quasi-Newton instance, the Jacobian can be rebuilt once in a while during the nonlinear iterations or just once, which is often preferred when simulating time-dependent problems. The underlying framework for the linear and nonlinear solvers is PETSc [19].

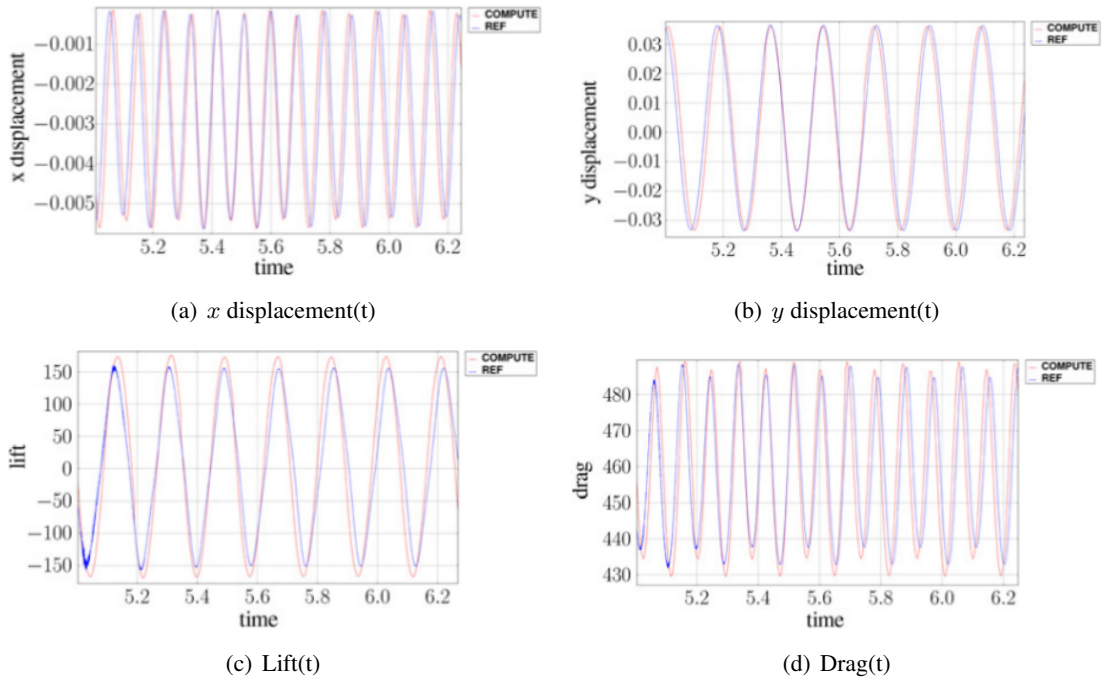
5. VALIDATION

We now present a FSI benchmark presented in [12] to validate our numerical strategy. This benchmark is a simulation of a flow in a rectangular domain with a rigid obstacle, a cylinder, and an elastic beam clamped to the cylinder. The benchmark configuration is taken from [12], which is a well documented test for FSI calculations. We monitor the lift and drag around the cylinder and the flag in Figures 5(c)-5(d) as well as the displacement of a point on the flag in Figures 5(a)-5(b). We compare our results with the ones provided in [12, 13, 14, 15, 16, 17]. In Figure 5, we denote **REF** the values computed by [14].

	x $[\times 10^{-3}]$	y $[\times 10^{-3}]$	Drag	Lift
[12]	$-2.69 \pm 2.53\,[10.9]$	$1.48 \pm 34.38\,[5.3]$	$457.3 \pm 22.66\,[10.9]$	$2.22 \pm 149.78\,[5.3]$
[13]			464.5 ± 40.50	$6.00 \pm 166.00\,[5.5]$
[14]	$-2.88 \pm 2.72\,[10.9]$	$1.47 \pm 34.99\,[5.5]$	$460.5 \pm 27.74\,[10.9]$	$2.50 \pm 153.91\,[5.5]$
[15]	$-4.54 \pm 4.34\,[10.1]$	$1.50 \pm 42.50\,[5.1]$	$467.5 \pm 39.50\,[10.1]$	$16.2 \pm 188.70\,[5.1]$
[16]			474.9 ± 28.10	$3.90 \pm 165.90\,[5.5]$
[17]	$-2.83 \pm 2.78\,[10.8]$	$1.35 \pm 34.75\,[5.4]$	$458.5 \pm 24.00\,[10.8]$	$2.50 \pm 147.50\,[5.4]$
(1)	$-2.86 \pm 2.74\,[10.9]$	$1.31 \pm 34.71\,[5.4]$	$459.7 \pm 29.97\,[10.9]$	$4.46 \pm 172.53\,[5.4]$
(2)	$-2.85 \pm 2.72\,[10.9]$	$1.35 \pm 34.62\,[5.4]$	$459.2 \pm 29.62\,[10.9]$	$3.53 \pm 172.73\,[5.4]$
(3)	$-2.88 \pm 2.75\,[10.9]$	$1.35 \pm 34.72\,[5.4]$	$459.3 \pm 29.84\,[10.9]$	$3.19 \pm 171.20\,[5.4]$

TABLE 1. Results for FSI3

The tests we present where chosen to stress that the use of high order approximations allow to reduce the number of degrees of freedom in the calculations, while obtaining results which are in accordance with reference values (see, for example, the configuration in [14], Table 2 and Figure 6). Only the lift



(a) x displacement(t)

(b) y displacement(t)

(c) Lift(t)

(d) Drag(t)

FIGURE 5. Results for **FSI3** with the configuration **(3)** in Table 2

is not in very good agreement: some little perturbations appear when we decrease the time step and it seems to be caused by the Aitken relaxation, more details in [13].

Finally, with the Figures 7(a) and 7(b), we show two screenshots of this benchmark.

	N_{elt}	N_{dof}	$[P_c^N(\Omega_{f,\delta})]^2 \times P_c^{N-1}(\Omega_{f,\delta}) \times V_{s,\delta}^N$	Δt
[14]	15872	304128		0.00025
(1)	1284	27400	$[P_c^4(\Omega_{f,(h,3)})]^2 \times P_c^3(\Omega_{f,(h,3)}) \times V_{s,(h,3)}^3$	0.005
(2)	2117	44834	$[P_c^4(\Omega_{f,(h,3)})]^2 \times P_c^3(\Omega_{f,(h,3)}) \times V_{s,(h,3)}^3$	0.005
(3)	4549	95427	$[P_c^4(\Omega_{f,(h,3)})]^2 \times P_c^3(\Omega_{f,(h,3)}) \times V_{s,(h,3)}^3$	0.005

TABLE 2. Benchmark configurations for **FSI3**

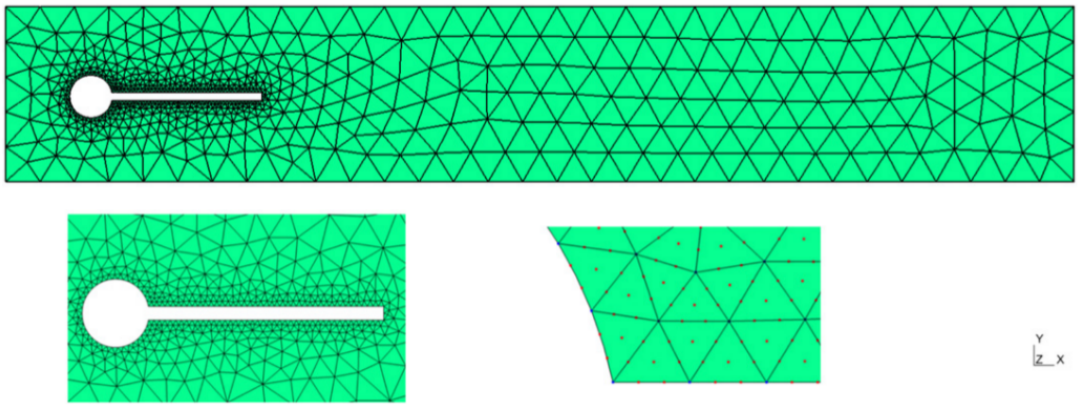
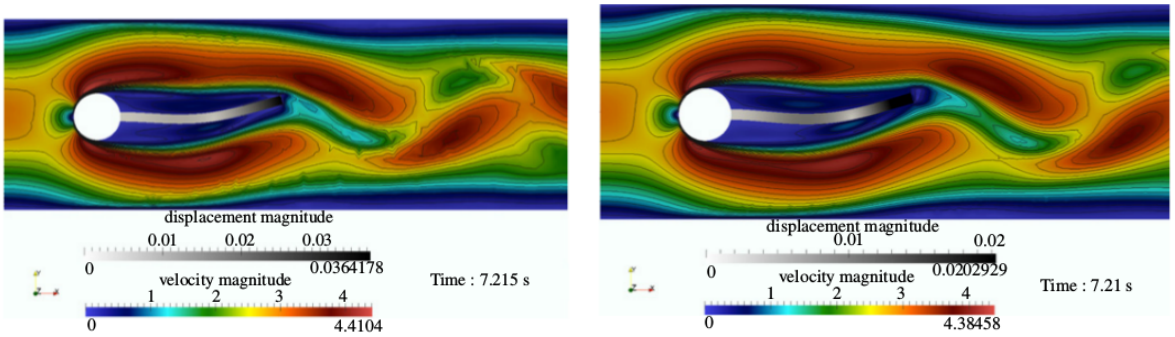


FIGURE 6. Reference mesh used for **(1)** in table 2



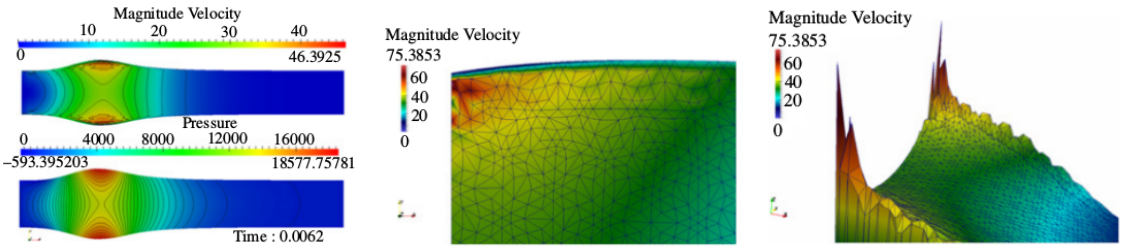
(a) Configuration **(1)** (b) Configuration **(3)**

FIGURE 7. Results for **FSI3** with the configuration **(1)** and configuration **(3)**

6. BLOOD FLOW APPLICATIONS

We now describe a 2D and 3D benchmark in the blood flow context which are simulating the propagation of a pressure pulse through a pipe. In both cases the minimal CIP stabilization strategy is used to control the velocity at the inlet. For the 2D case, we have chosen $\mathcal{F} = \mathcal{F}_h^i \cap \{\mathbf{x} \in \mathbb{R}^2 : (\mathbf{x}_{in} - \mathbf{x}, \mathrm{n}_{in}) \leqslant 0\}$, with $\mathbf{x}_{in} = (0.3, 0)$. For the 3D case, we stabilize only on the faces of the inlet, i.e. $\mathcal{F} = \Gamma_{in} \cap \mathcal{F}_h^b$.

6.1. **2D Benchmark.** The geometry is a rectangle of 6 cm long by 1cm. In this particular case, we use a 1D reduced model for the structure, the *generalized string model*, using the discretization as in [5]. The purpose of this simulation is to show that under realistic parameters, the minimal CIP stabilization, as well as the high order approximation for both fluid and structure provide satisfactory results.



(a) Velocity (top) and pressure (bottom) using minimal CIP stabilisation (b) Velocity without stabilisation (t=0.0027s) (c) Velocity without stabilisation (t=0.0027s)

FIGURE 8. Simulation using a $P_c^4(\Omega_{f,(h,3)}) \times P_c^3(\Omega_{f,(h,3)})$ fluid discretisation, $V_{s,(h,1)}^3$ for the structure and timestep $\Delta t = 10^{-4}$. All others parameters are taken from [20].

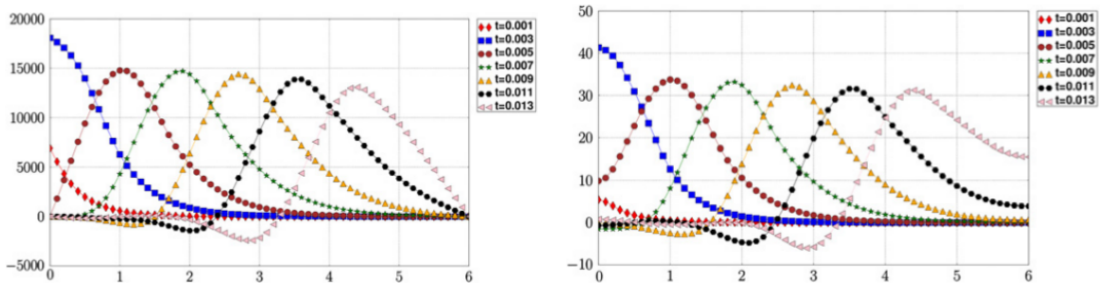
We plot the flow rate and average pressure at various cross sections of the flow for several timesteps in Figure 9. The results are in good agreement with [20].

6.2. **3D Benchmark.** Finally we present a blood flow application in large arteries, as presented in [21]. The geometry is a straight pipe of length $L = 6$ with an axis $(1, 0, 0)$ and a radius 0.5. A pressure pulse $1.3332 \times 10^4 g/\left(cm\ s^2\right)$ has been imposed at the inlet boundary during $0.003s$. The thin elastic vessel $(0.1cm)$ is clamped at the inlet and outlet. Figure 10 displays the radial displacement at various points of the FSI interface while figure 11 shows the pressure wave propagation for different time steps. We have used a $P_c^3(\Omega_{f,(h,2)}) \times P_c^2(\Omega_{f,(h,2)})$ space for the fluid and $V_{2,(h,2)}$ for the structure. The time scheme for the fluid is a BDF scheme of second order.

Again the results are in agreement with [21] however a more thorough benchmarking is needed.

7. CONCLUSION

We now have a complete high order fluid-structure interaction framework in 2D and 3D. However much remains to be done in various areas. Indeed we need to make a thorough study of the presented



(a) Average pressure (b) Flow rate

FIGURE 9. Average pressure and flow rate at various cross-sections of the flow and various time steps.

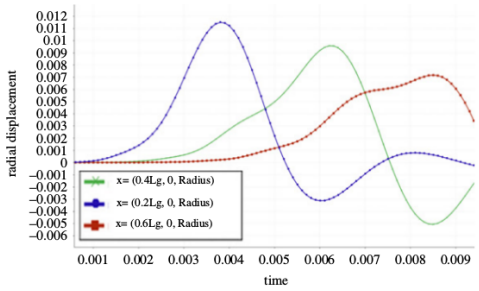
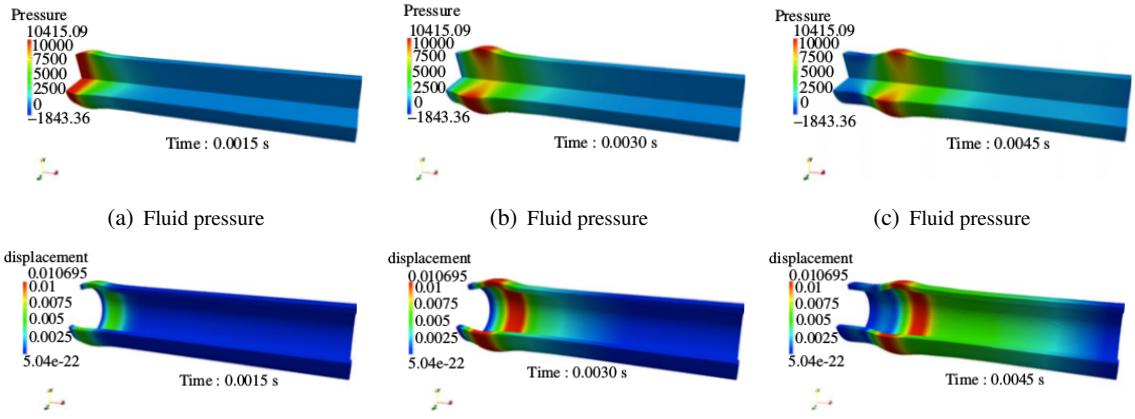


FIGURE 10. Radial displacement versus time for several points on the fsi interface



(a) Fluid pressure (b) Fluid pressure (c) Fluid pressure

(d) Structure displacement (e) Structure displacement (f) Structure displacement

FIGURE 11. Pressure wave in a straight pipe. We show the fluid pressure and the fluid displacement of the pipe is magnified 10 times.

framework for solving the Navier-Stokes equations in moving domains in terms of approximations — the Arbitrary Lagrangian Eulerian framework in particular — as well the underlying linear algebra solvers updating the results of our previous paper [2]. The FSI framework needs also to be more thoroughly benchmarked especially in 3D however there are no available benchmarks to our knowledge. Also, in the three dimensional case, a more robust strategy for the first step of the ALE map is under investigation.

Acknowledgments. The authors would like to thank the Région Rhône-Alpes (ISLE/CHPID project) as well as the french National Research Agency (MOSICOB and Cosinus-HAMM projects) and the Calouste Gulbenkian Foundation (*Estímulo à Investigação* project) for their financial support. This work was also partially funded by the Centre for Mathematics of the University of Coimbra and *Fundação para a Ciência e Tecnologia*, through European program COMPETE/FEDER.

REFERENCES

[1] G. Pena and C. Prud'homme, Construction of a high order fluid-structure interaction solver, J. Comput. Appl. Math. 234 (7) (2010) 2358–2365.

[2] G. Pena, C. Prudhomme and A. Quarteroni, High order methods for the approximation of the incompressible Navier-Stokes equations in a moving domain, Computer Methods in Applied Mechanics and Engineering 209-212 (0) (2012) 197 – 211. doi:10.1016/j.cma.2011.09.016.

[3] C. Prud'homme, Life: Overview of a unified C++ implementation of the finite and spectral element methods in 1D, 2D and 3D, 2007.

[4] C. Prud'homme, A domain specific embedded language in C++ for automatic differentiation, projection, integration and variational formulations.

[5] G. Pena, Spectral element approximation of the incompressible Navier-Stokes equations evolving in a moving domain and applications, Ph.D. thesis, École Polytechnique Fédérale de Lausanne, n°4529 (November 2009).

[6] C. Prud'Homme, V. Chabannes, V. Doyeux, M. Ismail, A. Samake and G. Pena, Feel++: A Computational Framework for Galerkin Methods and Advanced Numerical Methods, submitted to ESAIM Proc. (Jan. 2012).
URL http://hal.archives-ouvertes.fr/hal-00662868

[7] C. Prud'homme, V. Chabannes and G. Pena, Feel++: Finite Element Embedded Language in C++, Free Software available at http://www.feelpp.org, contributions from A. Samake, V. Doyeux, M. Ismail and S. Veys.

[8] C. Geuzaine and J.-F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre-and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (11) (2009) 1309–1331.

[9] W. J. Gordon and C. A. Hall, Construction of curvilinear coordinate systems and their applications, Int. J. Numer. Meth. Eng. 7 (1973) 461–477.

[10] W. J. Gordon and C. A. Hall, Transfinite element methods: blending-function interpolation over arbitrary curved element domains, Int. J. Numer. Meth. Eng. 21 (1973) 109–129.

[11] A. Winslow, Numerical solution of the quasilinear poisson equations in a nonuniform triangle mesh, J. Comp. Phys. 2 (1967) 149–172.

[12] S. Turek and J. Hron, Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow, Fluid-Structure Interaction (2006) 371–385.

[13] M. Breuer, G. De Nayer, M. Münsch and T. Gallinger and R. Wüchner, Fluid-structure interaction using a partitioned semi-implicit predictor-corrector coupling scheme for the application of large-eddy simulation, Journal of Fluids and Structures 29 (2012) 107–130.

[14] Turek, S. and Hron, J. and Mádlík, M. and Razzaq, M. and Wobker, H. and Acker, J. F., Numerical Simulation and Benchmarking of a Monolithic Multigrid Solver for Fluid-Structure Interaction Problems with Application to Hemodynamics, Springer Berlin Heidelberg, Lecture Notes in Computational Science and Engineering 73 (2010) 193–220.

[15] Münsch, M. and Breuer, M., Numerical Simulation of Fluid-Structure Interaction Using Eddy-Resolving Schemes, Springer Berlin Heidelberg, Lecture Notes in Computational Science and Engineering 73 (2010) 221–253

[16] Gallinger, T., 2010. Effiziente Algorithmen zur partitionierten Lösung stark gekoppelter Probleme der Fluid-Struktur-Wechselwirkung. Ph.D. Thesis. Lehrstuhl fur Statik, Technische Universitat Munchen, Germany.

[17] Sandboge, R., Fluid-structure interaction with OpenFSITM and MD NastranTM structural solver, Ann Arbor 1001 (2010)

[18] J. Hermansson and P. Hansbo, A variable diffusion method for mesh smoothing, Communications in Numerical Methods in Engineering 19 (2003) 897–908.

[19] S. Balay, W. D. Gropp, L. C. McInnes and B. F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202.

[20] F. Nobile, Numerical approximation of fluid-structure interaction problems with application to haemodynamics, PhD, Ecole Polytechnique Féderale de Lausanne, Switzerland.

[21] L. Formaggia, A. Quarteroni and A. Veneziani, Cardiovascular Mathematics: Modeling and simulation of the circulatory system, Springer Verlag, 2009.

[22] V. Chabannes, Towards blood flow simulation, PhD, Université de Grenoble, Grenoble (end of 2012).

[23] Valdés Vázquez, J.G., Fluid-Structure Interaction Analysis ,VDM Verlag Dr. Mueller e.K., 2008.

[24] Liseĭkin, VD, Grid generation methods, Springer Verlag (1999)

V. CHABANNES, UNIVERSITÉ GRENOBLE 1 / CNRS, LABORATOIRE JEAN KUNTZMAN / UMR 5224. GRENOBLE, F-38041, FRANCE

E-mail address: vincent.chabannes@imag.fr

G. PENA, CMUC, UNIVERSITY OF COIMBRA, LARGO D. DINIS, APARTADO 3008, 3001-454 COIMBRA, PORTUGAL

E-mail address: gpena@mat.uc.pt

C. PRUD'HOMME, UNIVERSITÉ GRENOBLE 1 / CNRS, LABORATOIRE JEAN KUNTZMAN / UMR 5224. GRENOBLE, F-38041, FRANCE, UNIVERSITÉ DE STRASBOURG / CNRS, IRMA / UMR 7501. STRASBOURG, F-67000, FRANCE

E-mail address: christophe.prudhomme@imag.fr