



High order fluid structure interaction in 2D and 3D. Application to blood flow in arteries

Vincent Chabannes, Gonalo Pena, Christophe Prud'Homme

► To cite this version:

Vincent Chabannes, Gonalo Pena, Christophe Prud'Homme. High order fluid structure interaction in 2D and 3D. Application to blood flow in arteries. 2012. hal-00657622v1

HAL Id: hal-00657622

<https://hal.science/hal-00657622v1>

Preprint submitted on 7 Jan 2012 (v1), last revised 2 Feb 2012 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

High order fluid structure interaction in 2D and 3D

Application to blood flow in arteries

Vincent Chabannes^a, Gonalo Pena^{b,**}, Christophe Prud’homme^{a,c,*}

^aUniversité Grenoble 1 / CNRS, Laboratoire Jean Kuntzman / UMR 5224. Grenoble, F-38041, France

^bCMUC, University of Coimbra, Largo D. Dinis, Apartado 3008, 3001-454 Coimbra, Portugal

^cUniversité de Strasbourg / CNRS, IRMA / UMR 7501. Strasbourg, F-67000, France

Abstract

Accuracy is critical if we are to trust the simulation’s predictions. In settings such as fluid-structure interaction it is all the more important to obtain reliable results to understand, for example, the impact of pathologies on blood flows in the cardiovascular system. In [1] we 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 the first part of the paper, we propose a new high order ALE construction which allows for any type — not only vertical as in [2] — of not too large displacements that applies to 2D as well as 3D. This construction relies on a first order approximation and a correction step that recovers the high order accuracy. In the second part of the paper we present an update of our fluid-structure interaction framework.

Keywords: High order methods, Arbitrary Lagrangian Eulerian transformation, Fluid-Structure Interaction

1. Introduction

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

The paper is organized in the following way: first, we introduce some notations and present a brief overview of the status of [6] in section 1.2 (which will be described in another publication). In section 2 we state our latest advances in one of our central ingredients to achieve high order fluid-structure interaction, namely the Arbitrary Lagrangian-Eulerian framework. Next, the fluid and structure models are presented, associated with their respective discretisation and solution methods. These methods are

*Corresponding author

**Principal corresponding author

Email addresses: vincent.chabannes@imag.fr (Vincent Chabannes), gpena@mat.uc.pt (Gonalo Pena), christophe.prudhomme@imag.fr (Christophe Prud’homme)

Preprint submitted to Journal of Computational and Applied Mathematics

January 7, 2012

validated based on the benchmark [7]. In the last section, we propose a fluid-structure interaction solver, combination of the tools presented throughout the paper. The solver is validated by the benchmark [7] and some 3D preliminary results are presented.

1.1. Notations

Given an elementary simplex domain $K^* \subset \mathbb{R}^d$, $d = 1, 2, 3$, and a positive integer N , let us denote by $\mathbb{P}^N(K^*)$ the space of polynomials of total degree less or equal than N , defined in K^* . We fix a reference element, \hat{K} , and consider a transformation $\varphi_{K^*} : \hat{K} \rightarrow K^*$, called the *geometrical transformation*.

We consider now two domains, Ω^* and $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, which we later refer as the *reference* and the *computational* domains, respectively. We further assume that the reference domain has a straight edge/face mesh associated with it, \mathcal{T}^* , ie, $\varphi_{K^*} \in \mathbb{P}^1(\hat{K})$, $\forall K^* \in \mathcal{T}^*$. Furthermore we admit that the mesh \mathcal{T}^* covers exactly the domain Ω^* , i.e., $\Omega^* = \bigcup_{K^* \in \mathcal{T}^*} \overline{K^*}$.

We denote $P_{c,h}^N(\Omega^*)$, $\mathbf{P}_{c,h}^N(\Omega^*)$ the spaces of piecewise scalar, respectively vectorial, polynomial of total degree N , continuous functions in Ω^* , and $P_{td,h}^N(\Omega^*)$ the space of piecewise polynomial of total degree N , totally discontinuous functions in Ω^* .

Finally let us denote by $\boldsymbol{\eta} : \partial\Omega^* \rightarrow \partial\Omega$ 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, $\boldsymbol{\eta}(s) = s$, $\forall s \in \Gamma_F^*$) 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 $\partial\Omega^* = \overline{\Gamma_M^*} \cup \overline{\Gamma_F^*} \cup \overline{\Gamma_N^*}$. Denote $\mathcal{T}^{*,b} = \{K^* \in \mathcal{T}^* : \partial K^* \cap \Gamma_M^* \neq \emptyset\}$ the set of elements K^* sharing a face with the boundary of Ω^* .

1.2. Computational framework for Galerkin methods

Our computational framework builds upon Feel++ [6] which allows for arbitrary order cG and dG 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 geometrical transformation of each element K of the mesh 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. One of the optimisations that allows to have a huge gain in computational effort is to straighten all the high order elements except for the boundary faces of the computational mesh. This is achieved by moving all the nodes associated to the high order transformation to the position

these nodes would have if a first order geometrical transformation were applied. This procedure can be formalized in the following operator

$$\eta_K^{\text{straightening}}(\varphi_K^N(\mathbf{x}^*)) = (\varphi_K^1(\mathbf{x}^*) - \varphi_K^N(\mathbf{x}^*)) - (\varphi_{K \cap \Gamma}^1(\mathbf{x}^*) - \varphi_{K \cap \Gamma}^N(\mathbf{x}^*)) \quad (1)$$

where \mathbf{x}^* is any point in K^* and $\varphi_K^1(\mathbf{x}^*)$ and $\varphi_K^N(\mathbf{x}^*)$ its images by the geometrical transformation of order one and order N , respectively. On one hand, the first two terms ensure that for all K not intersecting Γ , the order one and N transformations produce the same image. On the other hand, the last two terms are 0 unless the image of \mathbf{x}^* is on Γ and, in this case, we don't move the high order image of \mathbf{x}^* . This allows to have straight internal elements and elements touching the boundary to remain high order. When applying numerical integration, specific quadratures are considered when dealing with internal elements or elements sharing a face with the boundary. The performances, thanks to this transformation, are similar to the ones obtained with first order meshes. However, it needs to be used with care as it can generate folded meshes.

2. High order ALE

We now turn to our high order Arbitrary Lagrangian-Eulerian (ALE) framework. A fundamental piece in performing simulations in the ALE framework is the transformation that maps the reference configuration onto the computational domain, at each timestep. This is called *ALE map*. In Pena and Prud'homme [1], the authors propose a high order ALE map that allows for an accurate description of the boundary of the computational domain, while inducing a straight edges in the interior elements of the computational domain's mesh. However, this construction has the disadvantage of relying upon the Gordon-Hall transformations, see Gordon-Hall [9, 10], which makes it (implementation-wise) intricate to extend to three dimensional domains.

To overcome this difficulty, we 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. We review here the construction from [1, 2]. The first step is to perform a modified harmonic extension (according to Masud and Kanchi [11]) of the displacement $\boldsymbol{\eta}$ to the interior of the reference domain using piecewise linear polynomial functions. The corresponding ALE transformation, \mathcal{A}^1 , satisfies a discrete element-weighted Laplace equation.

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

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

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}^N is obtained by adding to \mathcal{A}^1 the correction $\mathcal{A}_{K^*}^N$ on each element of $\mathcal{T}^{*,b}$

$$\mathcal{A}^N(\mathbf{x}^*) = \mathcal{A}_1(\mathbf{x}^*) + \sum_{K^* \in \mathcal{T}^{*,b}} \mathcal{A}_{K^*}^N(\mathbf{x}^*) + \mathbf{x}^*$$

Proposition 1 (Properties of \mathcal{A}^N). *Under the previous assumptions and by construction, $\mathcal{A}^N \in \mathbf{P}_{c,h}^N(\mathcal{T}^{*,b})$ — enjoys optimal approximation properties i.e. the boundary approximation is $\mathcal{O}(h^{N+1})$ in the L_2 -norm — and $\mathcal{A}^N \in \mathbf{P}_{c,h}^1(\mathcal{T}^* \setminus \mathcal{T}^{*,b})$.*

2.1. The Harmonic extension and Winslow smoother

In both papers [1, 2], the piecewise linear map created in the first step is calculated by performing 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 [12]. 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 PDE's, which can be done using fixed point iterations. In Figure 1 we show the effect of the modified harmonic extension and the Winslow smoother for the same testcase.

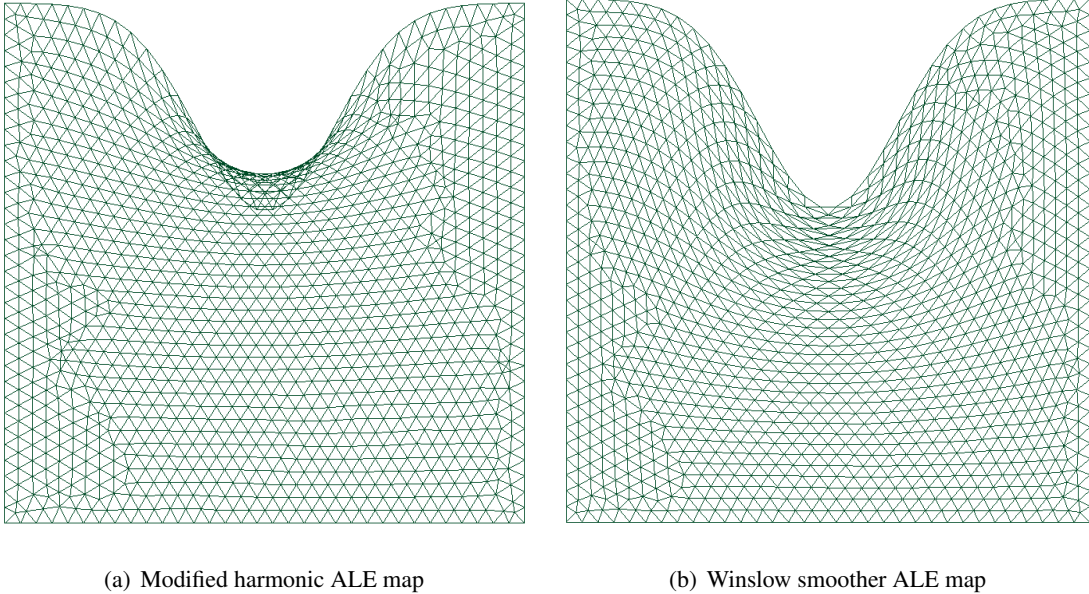


Figure 1: Comparison of first order meshes generated by the harmonic extension (left) and Winslow smoother (right) operators respectively

2.2. Numerical verification of Proposition 1

We present now some numerical experiments to verify Proposition 1. In order to conduct these experiments we use (i) Gmsh [8] to build the initial high order meshes in 2D and 3D and visualize the meshes computed by our methods and (ii) Feel++ [3, 6] which provides the framework for arbitrary order Galerkin methods to solve the partial differential equations and handle the computational meshes. Note that the mesh \mathcal{T}^* used to solve equation (1) is built automatically from the initial high order mesh through the straightening process.

We consider the reference domain depicted in Figure 2(a) defined by

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

and the associated displacement of its boundary $\boldsymbol{\eta}^{cy}(\mathbf{x}^*) = 0.2 \exp(\frac{x^*}{5}) \sin\left(\frac{\pi x^*}{2.5}\right) \mathbf{n}^*$. Figure 2(b) displays the computational domains colored by the corresponding ALE map.

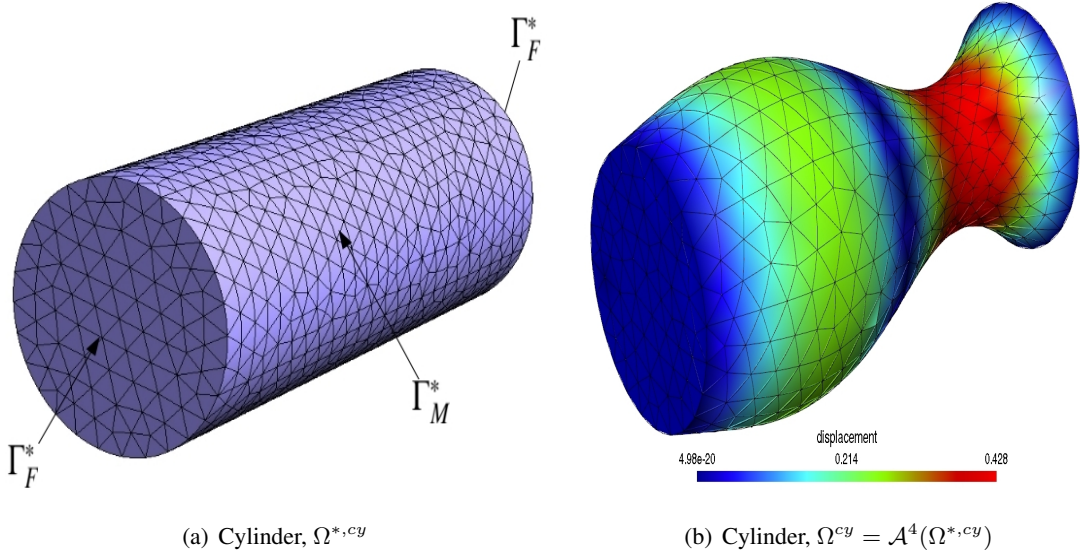
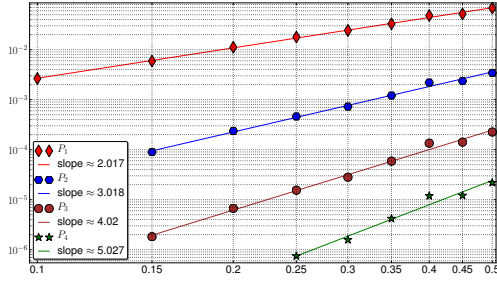
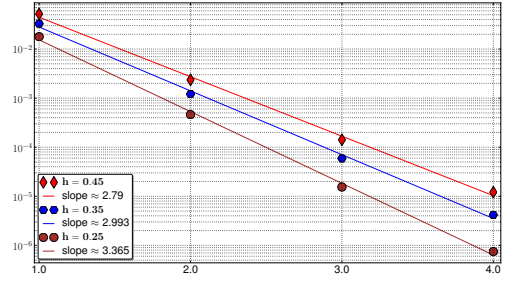


Figure 2: Reference (left) and computational (right) meshes of order 4 displayed using Gmsh colored by the displacement 2-norm

Finally, Figure 3(a) displays the convergence rate of the quantity $\|\mathcal{A}^N(\mathbf{x}^*) - (\mathbf{x}^* + \boldsymbol{\eta}(\mathbf{x}^*))\|_{[L^2(\Gamma_M^*)]^d}$ which confirms the result of Proposition 1. Figure 3(b) represents the convergence rate with respect to geometric order N .



(a) Cylinder



(b) Cylinder

Figure 3: Convergence rate plots of $\mathcal{A}^{N,cy}$ with respect to h (left) and convergence rate plots of $\mathcal{A}^{N,cy}$ with respect to N (right).

3. Models and their discretizations methods

We now turn to the structure and fluid models, their associated discretization and solution methods. In each case, we display some of the results of our strategy applied to the benchmark [7]. Note that our fluid solver has also been benchmarked for bi-fluid simulations [13] following the proposal [14].

Two benchmarks presented in [7] are performed to verify our code. The first one corresponds to the oscillation of an elastic beam only subjected to gravity. We monitor the coordinates of the tip of the beam. The second is the simulation of a flow in rectangular domain with a rigid obstacle: a flag clamped to a cylinder. We monitor the lift and drag.

The algebraic systems arising from the discretization proposed in the next sections 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 stiff and the preconditioner needs to be recalculated. 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 [15].

3.1. Structure

3.1.1. Models and discretizations

We first introduce the deformation gradient tensor, which allows to measure the solid deformation from the displacement η_s , $F_s = I + \nabla \eta_s$.

Other useful tensors are the right Cauchy-Green tensor C_s and the Green-Lagrange tensor E_s which can be expressed by $C_s = F_s^T F_s$, $E_s = \frac{1}{2} (C_s - I)$ where E_s has the property to be divided in two terms, ϵ_s (linear) and γ_s (quadratic)

$$E_s = \underbrace{\frac{1}{2} (\nabla \eta_s + (\nabla \eta_s)^T)}_{\epsilon_s} + \underbrace{\frac{1}{2} ((\nabla \eta_s)^T \nabla \eta_s)}_{\gamma_s}. \quad (3)$$

The simplest model of this type is the **elastic linear model**, which is valid for small displacements and deformations. It reads as

$$\rho_s \frac{\partial^2 \boldsymbol{\eta}_s}{\partial t^2} - \nabla \cdot (\boldsymbol{\Sigma}_s) = \mathbf{f}_s, \quad \boldsymbol{\Sigma}_s = \lambda_s (tr \boldsymbol{\epsilon}_s) \mathbf{I} + 2\mu_s \boldsymbol{\epsilon}_s \quad (4)$$

with λ_s and μ_s being the Lamé coefficients.

The next model is the **hyper-elastic and compressible model** which is valid for large deformations. The generic balance equation of hyper-elasticity in Lagrangian description is given by

$$\rho_s \frac{\partial^2 \boldsymbol{\eta}_s}{\partial t^2} - \nabla \cdot (\mathbf{F}_s \boldsymbol{\Sigma}_s) = \mathbf{f}_s, \quad \boldsymbol{\Sigma}_s = \lambda_s (tr \mathbf{E}_s) \mathbf{I} + 2\mu_s \mathbf{E}_s. \quad (5)$$

where $\boldsymbol{\Sigma}_s$ represents the second Piola-Kirchhoff stress tensor.

To take into account the material incompressibility, we use the **Hyper-elastic and incompressible model**. The pressure p_s acts as a lagrange multiplier to enforce incompressibility. The model reads as

$$\rho_s \frac{\partial^2 \boldsymbol{\eta}_s}{\partial t^2} - \nabla \cdot (\mathbf{F}_s \boldsymbol{\Sigma}_s) = \mathbf{f}_s, \quad det \mathbf{F}_s = 1, \quad \boldsymbol{\Sigma}_s = -p_s (det \mathbf{F}_s) \mathbf{C}_s^{-1} + \lambda_s (tr \mathbf{E}_s) \mathbf{I} + 2\mu_s \mathbf{E}_s. \quad (6)$$

Finally we also make use of a **1D reduced model** for thin structures like shells, also known as generalized strings, see [5, 16].

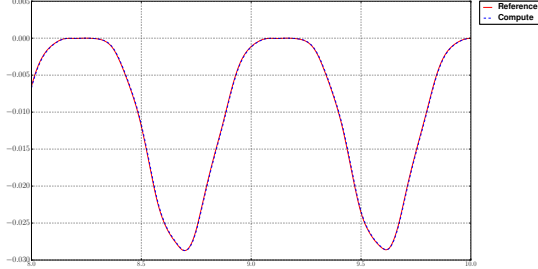
For the numerical results presented in the next section we considered a hyper-elastic model for the structure. We discretized the equations using $[\mathbb{P}^N]^d$ elements and a continuous approximation in space while we used the Newmark method to get an order 2 discretization in time.

3.1.2. Benchmark

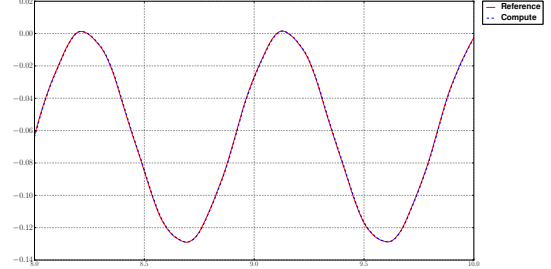
The results of the CSM3 benchmark from [7] are displayed in Table 1 and the x, y coordinates of the tip of the beam on the Figure 4. The results are in accordance with the reference values **REF**.

$N_{element}$	N_{dof}	$x [\times 10^{-3}]$	$y [\times 10^{-3}]$
REF		$-14.305 \pm 14.305 [1.0995]$	$-63.607 \pm 65.160 [1.0995]$
4199	17536(P_2)	$-14.585 \pm 14.59 [1.0953]$	$-63.981 \pm 65.521 [1.093]$
4199	38900(P_3)	$-14.589 \pm 14.594 [1.0953]$	$-63.998 \pm 65.522 [1.093]$
1043	17422(P_4)	$-14.591 \pm 14.596 [1.0953]$	$-64.009 \pm 65.521 [1.093]$
4199	68662(P_4)	$-14.59 \pm 14.595 [1.0953]$	$-64.003 \pm 65.522 [1.093]$
4199	17536(P_2)	$-14.636 \pm 14.64 [1.0969]$	$-63.937 \pm 65.761 [1.0945]$
4199	38900(P_3)	$-14.642 \pm 14.646 [1.0969]$	$-63.949 \pm 65.771 [1.0945]$
1043	17422(P_4)	$-14.645 \pm 14.649 [1.0961]$	$-63.955 \pm 65.778 [1.0945]$
4199	68662(P_4)	$-14.627 \pm 14.629 [1.0947]$	$-63.916 \pm 65.739 [1.0947]$
4199	17536(P_2)	$-14.645 \pm 14.645 [1.0966]$	$-64.083 \pm 65.623 [1.0951]$
4199	38900(P_3)	$-14.649 \pm 14.65 [1.0966]$	$-64.092 \pm 65.637 [1.0951]$
1043	17422(P_4)	$-14.652 \pm 14.653 [1.0966]$	$-64.099 \pm 65.645 [1.0943]$

Table 1: Results for **CSM3** with $\Delta t = 0.02, 0.01, 0.005$. Line **REF** displays the reference values for [7]



(a) x displacement



(b) y displacement

Figure 4: Results for **CSM3** with $\Delta t = 0.005$. x and y displacement of the point A depending on time

3.2. Fluid

3.2.1. Models and discretization

We mainly use a Newtonian fluid model which neglects shear-thinning and viscoelastic effects. The govern equations are the classical Navier-Stokes equations which read as

$$\rho_f \frac{\partial \mathbf{u}_f}{\partial t} + \rho_f (\mathbf{u}_f \cdot \nabla) \mathbf{u}_f - \nabla \cdot \boldsymbol{\sigma}_f = \mathbf{f}_f \quad (7)$$

$$\nabla \cdot \mathbf{u}_f = 0 \quad (8)$$

where \mathbf{u}_f is the fluid velocity and ρ_f its density and

$$\boldsymbol{\sigma}_f = -p_f \mathbf{I} + \boldsymbol{\tau}_f, \quad \boldsymbol{\tau}_f = 2\mu_f \mathbf{D}_f, \quad \mathbf{D}_f = \frac{1}{2} \left(\nabla \mathbf{u}_f + (\nabla \mathbf{u}_f)^T \right) \quad (9)$$

where p_f is the pressure and μ_f the fluid viscosity.

The previous equations are discretized using the standard Taylor-Hood element $[\mathbb{P}^N]^d \times \mathbb{P}^{N-1}$ in space and the BDF of order 2 or 3 discretizations in time. In 2D the geometry is discretized using order 1 to 5 geometric transformations while in 3D from order 1 to 4, see [8].

3.2.2. Benchmark

The results of the CFD3 benchmark from [7] are displayed in Table 3.2.2 for various geometrical (N_{geo}) and velocity/pressure $N/N-1$ approximations. The results in Table 3.2.2 for a BDF2 scheme and time step 5×10^{-3} are in accordance with the **REF** results. However, we made some extra computations using a smaller time step $\Delta t = 0.002$ and BDF2/BDF3 schemes and we observe a significant shift in the Drag and Lift coefficients. This shift is already present with BDF3 and $\Delta t = 0.005$. Several computations were made using various meshes and discretization for the velocity, pressure and geometry. BDF2 and BDF3 for $\Delta t = 0.002$ are very much in accordance with the reference results except perhaps for the mean of the Lift which tells us that we are probably resolving properly the time and spatial scales while $\Delta t = 0.005$ using BDF2 or BDF3 is not sufficient.

N_{geo}	$N_{element}$	$N_{dof} (N, N-1)$	N_{bdf}	Drag	Lift
REF				$439.45 \pm 5.6183 [4.3956]$	$-11.893 \pm 437.81 [4.3956]$
P_1	8042	$37514(P_2/P_1)$	2	$437.47 \pm 5.3750 [4.3457]$	$-9.7865 \pm 437.54 [4.3457]$
P_2	2334	$26706(P_3/P_2)$	2	$439.27 \pm 5.1620 [4.3457]$	$-8.887 \pm 429.06 [4.3457]$
P_2	7970	$89790(P_3/P_2)$	2	$439.56 \pm 5.2335 [4.3457]$	$-11.719 \pm 425.81 [4.3457]$
P_1	3509	$39843(P_3/P_2)$	2	$438.24 \pm 5.5375 [4.3945]$	$-11.024 \pm 433.90 [4.3945]$
P_1	8042	$90582(P_3/P_2)$	2	$439.25 \pm 5.6130 [4.3945]$	$-10.988 \pm 437.70 [4.3945]$
P_2	2334	$26706(P_3/P_2)$	2	$439.49 \pm 5.5985 [4.3945]$	$-10.534 \pm 441.02 [4.3945]$
P_2	7970	$89790(P_3/P_2)$	2	$439.71 \pm 5.6410 [4.3945]$	$-11.375 \pm 438.37 [4.3945]$
P_3	3499	$73440(P_4/P_3)$	3	$439.93 \pm 5.8072 [4.4921]$	$-14.511 \pm 440.96 [4.3945]$
P_4	2314	$78168(P_5/P_4)$	2	$439.66 \pm 5.6412 [4.3945]$	$-11.329 \pm 438.93 [4.3945]$
P_3	2340	$49389(P_4/P_3)$	2	$440.03 \pm 5.7321 [4.3945]$	$-13.25 \pm 439.64 [4.3945]$
P_3	2334	$49266(P_4/P_3)$	3	$440.06 \pm 5.7773 [4.3945]$	$-14.092 \pm 440.07 [4.3945]$

Table 2: Results for **CFD3** with $\Delta t = 0.01, 0.005, 0.002$. Line **REF** displays the reference values for [7]

4. Fluid structure interaction

In the fluid-structure interaction context, we chose to write the fluid dynamics equations in the *Arbitrary Lagrangian Eulerian* (ALE) framework. It allows to take into account the deformation of

the fluid domain. We need also to introduce into our model the domain's velocity of deformation \mathbf{w}_f , see e.g. [1, 2], the fluid equation set now reads in the moving domain Ω_t over the time interval I :

$$\rho_f \frac{\partial \mathbf{u}_f}{\partial t} \Big|_{\mathbf{x}^*} - \mathbf{div}_{\mathbf{x}}(2\mu_f \mathbf{D}_{\mathbf{x}}(\mathbf{u}_f)) + \rho_f((\mathbf{u}_f - \mathbf{w}_f) \cdot \nabla_{\mathbf{x}}) \mathbf{u}_f + \nabla_{\mathbf{x}} p_f = \mathbf{f}, \quad \text{in } \Omega_t \times I \quad (10)$$

$$\mathbf{div}_{\mathbf{x}}(\mathbf{u}_f) = 0, \quad \text{in } \Omega_t \times I \quad (11)$$

where all differential operators are defined w.r.t. the Eulerian coordinate system, except the ALE time derivative.

The fluid and the structure are coupled through a partitioned method with an implicit or semi-implicit scheme, see [1, 5] for more details.

4.1. 2D Benchmark

This final benchmark is a mix of the two previous tests. We remove the gravity and the flag part (not the cylinder) is now allowed to move. The results are in accordance with the reference values even though we used a really coarse mesh except for the drag. A complete study using high order approximation like for the fluid shall be available at the time of Acomen'11.

	$x [\times 10^{-3}]$	$y [\times 10^{-3}]$	Drag	Lift
REF	$-2.69 \pm 2.53 [10.9]$	$1.48 \pm 34.38 [5.3]$	$457.3 \pm 22.66 [10.9]$	$2.22 \pm 149.78 [5.3]$
	$-2.78 \pm 2.69 [10.7]$	$1.55 \pm 33.87 [5.3]$	$459.8 \pm 31.06 [10.7]$	$-2.19 \pm 174.45 [5.3]$

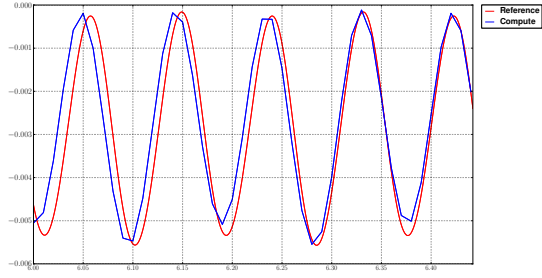
Table 3: Results for **FSI3** with $\Delta t = 0.01$. There are 1226 elements in Ω_f and 26211 dofs associated (P_4/P_3) , 260 elements in Ω_s and 2612 dofs associated (P_3) . The geometry is first order and the BDF scheme for the fluid is order 2. **REF** values are found in [7]

4.2. 3D Benchmark

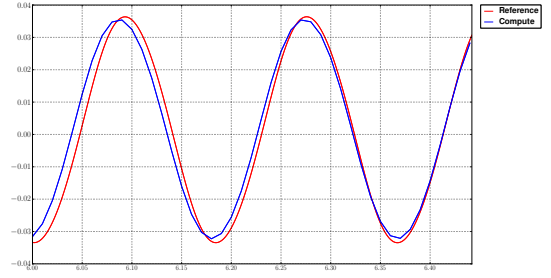
Finally we present a blood flow application in large arteries that has been proposed in [16]. The geometry is a straight pipe and a pressure pulse of $1.3332 \times 10^4 g / (cm s^2)$ has been imposed at the inlet boundary during $0.003s$. The thin elastic vessel ($0.1cm$) is clamped at the inlet and outlet. Figure 6 shows the pressure wave propagation for different time steps. We have used a (P_2/P_1) space for the fluid and P_1 for the structure. The geometry for the fluid and structure is order one. The time scheme for the fluid is also order 1.

5. Conclusion

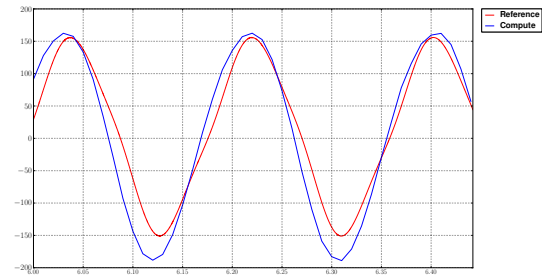
We have now a complete high order fluid-structure interaction framework in 2D and 3D. However much remains to be in various areas. Indeed we need to make a thorough study of our Navier-Stokes in moving domains framework in terms of approximations — the Arbitrary Lagrangian Eulerian in



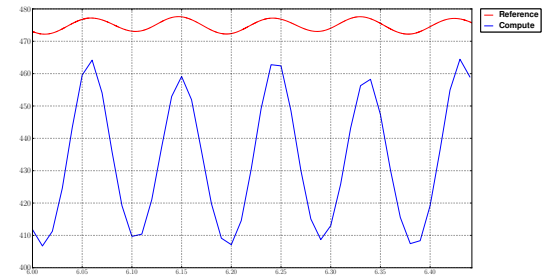
(a) x displacement(t)



(b) y displacement(t)



(c) Lift(t)



(d) Drag(t)

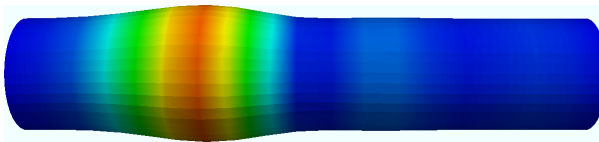
Figure 5: Results for **FSI3** with $\Delta t = 0.01$.



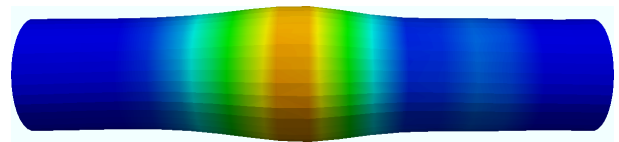
(a) $t=0.0015$



(b) $t=0.0030$



(c) $t=0.0045$



(d) $t=0.0060$

Figure 6: Pressure wave in a straight pipe. We show the fluid pressure and the fluid displacement of the pipe is magnified 15 times.

particular — as well the underlying linear algebra solvers updating the results of our previous paper [2]. We also have to compare various fluid-structure strategies as well as the wide range of space time and geometry approximations at our disposal to provide a complete overview.

Acknowledgments

This work is funded by the Région Rhône-Alpes (ISLE/CHPID project), the ANR (MOSICOB and MN-HAMM project) and Calouste Gulbenkian Foundation (*Estímulo à Investigação* project).

References

- [1] G. Pena, 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, A. Quarteroni, High order methods for the approximation of the incompressible navierstokes equations in a moving domain, *Computer Methods in Applied Mechanics and Engineering* 209212 (0) (2012) 197 – 211. doi:10.1016/j.cma.2011.09.016.
URL <http://www.sciencedirect.com/science/article/pii/S0045782511003124>
- [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).
URL <http://library.epfl.ch/theses/?nr=4529>
- [6] C. Prud'homme, V. Chabannes, 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.
- [7] S. Turek, J. Hron, Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow, *Fluid-Structure Interaction* (2006) 371–385.
- [8] C. Geuzaine, 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, C. A. Hall, Construction of curvilinear coordinate systems and their applications, *Int. J. Numer. Meth. Eng.* 7 (1973) 461–477.
- [10] W. J. Gordon, C. A. Hall, Transfinite element methods: blending-function interpolation over arbitrary curved element domains, *Int. J. Numer. Meth. Eng.* 21 (1973) 109–129.
- [11] H. Kanchi, A. Masud, A 3D adaptive mesh moving scheme, *International Journal for Numerical Methods in Fluids* 54 (6-8) (2007) 923–944.
- [12] A. Winslow, Numerical solution of the quasilinear poisson equations in a nonuniform triangle mesh, *J. Comp. Phys.* 2 (1967) 149–172.
- [13] V. Doyeux, Y. Guyot, V. Chabannes, C. Prud'homme, M. Ismail, Simulation of two phase flow using a level set method. application to vesicle dynamics, in: *Fifth International Conference on Advanced Computational Methods in Engineering (ACOMEN 2011)*, 2011.
- [14] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, L. Tobiska, Quantitative benchmark computations of two-dimensional bubble dynamics, *International Journal for Numerical Methods in Fluids* 60 (11) (2009) 1259–1288.

doi:10.1002/fld.1934.

URL <http://dx.doi.org/10.1002/fld.1934>

- [15] S. Balay, W. D. Gropp, L. C. McInnes, 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.
- [16] F. Nobile, Numerical approximation of fluid-structure interaction problems with application to haemodynamics, PhD, Ecole Polytechnique Fédérale de Lausanne, Switzerland.