

Practical Contributions on the Fictitious Domain Method for a Fluid–Structure Interaction Problem



Guillaume Delay and Michel Fournié

Abstract The present study deals with the numerical simulation of a fluid–structure interaction problem. The fluid is represented by the incompressible Navier–Stokes equations and the structure is described by an ODE depending on two degrees of freedom. A recent fictitious domain method on a fixed mesh is considered. For that choice, we provide several tricks to meet the difficulties arising from the fluid–structure interaction. All developed tools can be applied to very general geometries and deformations of the structure. Finally, numerical simulations are conducted in a realistic aeronautics configuration.

1 Introduction

The interaction between a fluid and a deformable structure, which appears in a very large field of industrial problems, has recently received increasing attention from the scientific community. Numerical simulations of such problems remain a challenging task.

A first difficulty comes from the fact that the whole system is the assembling of two subsystems of different natures. We can then either consider the full system as a whole part and write a variational formulation that comprises the fluid and the structure equations or we can use appropriate solvers for each of these subsystems. The first approach is called monolithic approach [22]. In the sequel, we considered the second approach called partitioned approach [12]. We solve separately the

G. Delay (✉)

Institut de Mathématiques de Toulouse, Université Paul Sabatier III & CNRS, Toulouse Cedex, France

e-mail: guillaume.delay@math.univ-toulouse.fr; guillaume.delay@enpc.fr

M. Fournié

Institut de Mathématiques de Toulouse, Toulouse, France

e-mail: michel.fournie@math.univ-toulouse.fr

© Springer Nature Switzerland AG 2020

G. R. Barrenechea, J. Mackenzie (eds.), *Boundary and Interior Layers, Computational and Asymptotic Methods BAIL 2018*, Lecture Notes in Computational Science and Engineering 135, https://doi.org/10.1007/978-3-030-41800-7_3

structure and the fluid equations at each time step. A possibility would have been, inside a time step, to iterate between the structure and the fluid solvers to reach an equilibrium between those states. This would have given a strong coupling at an increased computational cost. However, for small enough time steps, a weak coupling is sufficient [10, 15].

Another challenge is to handle the fluid domain that changes over time. This is indeed a difficulty as most of the numerical methods use fitted meshes, i.e. meshes such that the physical boundary is composed of cell faces. Hence, if we want to use such a fitted mesh, we need to change it at every time step to fit the moving boundary. Some algorithms such as the Arbitrary Lagrangian Eulerian (ALE) approach make it possible with small computation costs for small deformations of the fluid domain [13]. However, for large deformations of the domain, a complete remeshing is needed, which is highly expensive.

In order to avoid any change of the mesh during the computation, we consider a fictitious domain approach, i.e. the boundary of the physical domain can arbitrary cut the mesh. Several methods are included in this framework, for instance the immersed boundary method [8, 19, 20] or the penalization method [2, 16, 18].

This kind of method has already been used for fluid–structure interaction in [1, 14, 17]. A recent review can be found in [4].

In the present work, we use a XFEM type method that can also be found under the name of cutFEM. This method has been developed in the context of crack propagation in fatigue mechanics [9]. The main characteristic of this method is the use of a level-set function to locate an interface in the domain and an enrichment of the finite element basis with functions depending on the position of the interface. In the fluid–structure context, the interface will be the boundary between the fluid and the structure. We adapt and investigate a recent method of that type that is theoretically analyzed for the Stokes problem only in [11]. We focus our attention on determinant implementation stages that are non-standard.

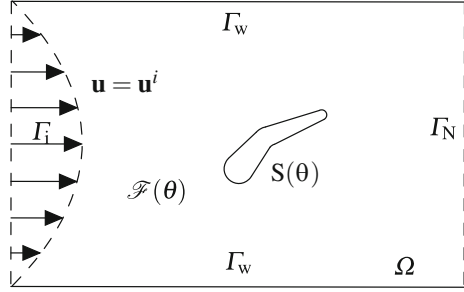
We present the considered model in Sect. 2, the partitioned process in Sect. 3, the XFEM in Sect. 4 and the handling of the moving domain in Sect. 5. Numerical simulations are given in Sect. 6. Finally, Sect. 7 is a summary of the article.

2 The Problem

The system we are interested in corresponds to the 2D interaction of a fluid and a deformable structure that can be assimilated to a steering gear depending on two parameters that we denote $\theta = (\theta_1, \theta_2) \in \mathbb{R}^2$.

The whole system is enclosed in a box representing a wind tunnel Ω . The boundary of that box can be decomposed as $\partial\Omega = \Gamma_i \cup \Gamma_w \cup \Gamma_N$, where Γ_i , Γ_w and Γ_N are parts of the boundary where inflow Dirichlet boundary conditions, homogeneous Dirichlet boundary conditions and Neumann-like boundary conditions are respectively imposed (see Fig. 1).

Fig. 1 The domain configuration: the whole domain (a wind tunnel) is decomposed as $\Omega = S(\theta) \cup \mathcal{F}(\theta)$ (the structure and the fluid)



We consider the incompressible Navier–Stokes equations for the fluid and a virtual work principle for the structure, see [6, 7]. The structure and fluid domains depend on the parameters θ , we denote them respectively $S(\theta)$ and $\mathcal{F}(\theta)$. We denote $Q_\theta = \cup_{t \in (0, T)} \{t\} \times \mathcal{F}(\theta(t))$, $\Sigma_\theta = \cup_{t \in (0, T)} \{t\} \times \partial S(\theta(t))$, $\Sigma_i = (0, T) \times \Gamma_i$, $\Sigma_w = (0, T) \times \Gamma_w$ and $\Sigma_N = (0, T) \times \Gamma_N$.

The equations considered for the fluid are the following ones

$$\left\{ \begin{array}{ll} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \Delta \mathbf{u} - \nabla p = \mathbf{f}_{\mathcal{F}} & \text{in } Q_\theta, & (1) \\ \operatorname{div} \mathbf{u} = 0 & \text{in } Q_\theta, & (2) \\ \mathbf{u} = \mathbf{u}^i & \text{on } \Sigma_i, & (3) \\ \mathbf{u} = 0 & \text{on } \Sigma_w, & (4) \\ \sigma_F(\mathbf{u}, p) \mathbf{n} = 0 & \text{on } \Sigma_N, & (5) \\ \mathbf{u} = \mathbf{u}_S & \text{on } \Sigma_\theta, & (6) \end{array} \right.$$

this system is completed by the following equation for the structure

$$M_\theta \ddot{\theta} = M_I(\theta, \dot{\theta}) + M_A(\theta, -\sigma_F(\mathbf{u}, p) \mathbf{n}) + \mathbf{f}_S \quad \text{on } (0, T), \quad (7)$$

and we consider the initial conditions

$$\mathbf{u}(0) = \mathbf{u}_0 \quad \text{in } \mathcal{F}(\theta_0), \quad (8)$$

$$\theta(0) = \theta_0 = (\theta_{1,0}, \theta_{2,0}), \quad \dot{\theta}(0) = \omega_0 = (\omega_{1,0}, \omega_{2,0}). \quad (9)$$

In the previous equations, we denoted \mathbf{u} the velocity field of the fluid, p the pressure field, ν the viscosity of the fluid, $\mathbf{f}_{\mathcal{F}}$ a source term acting as a force per unit volume, \mathbf{u}^i a Dirichlet datum on the inflow boundary Γ_i , $\sigma_F(\mathbf{u}, p) = \nu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - pI$ the stress tensor of the fluid, \mathbf{n} the unit outward normal to Ω , \mathbf{u}_S the velocity field of the structure, \mathbf{f}_S a source term on the structure equation. These equations have to be completed with suitable expressions for M_θ , M_I , M_A and \mathbf{u}_S .

The matrix $M_\theta \in \mathbb{R}^{2 \times 2}$, which is invertible, and the vectors $M_I, M_A \in \mathbb{R}^2$ depend respectively on θ ; on $\theta, \dot{\theta}$; on θ and the force exerted by the fluid on the structure, i.e. $\sigma_F(\mathbf{u}, p)\mathbf{n}_\theta$, where \mathbf{n}_θ is the unit normal to $\partial S(\theta)$ pointing inward the structure. For a more general presentation of the numerical scheme, we do not specify those terms. The numerical simulations will be led with the expressions given in Sect. 6.1. For those terms, the well-posedness of Eqs. (1)–(9) has been proven in [7]. The reader can find there further information.

3 The Time-Marching Process

We describe in this section, the time-marching process which is of partitioned type. This means that we solve the structure and the fluid systems separately and one after the other.

The fluid step at time t^{n+1} will be obtained by discretizing with a semi-implicit scheme the variational formulation: Find $(\mathbf{u}^{n+1}, p^{n+1}, \boldsymbol{\lambda}^{n+1}) \in (\mathbf{H}^1(\mathcal{F}(\theta^{n+1}))) \times (L^2(\mathcal{F}(\theta^{n+1}))) \times (\mathbf{H}^{-1/2}(\partial S(\theta^{n+1})) \times \mathbf{H}^{-1/2}(\Gamma_i \cup \Gamma_w))$ such that

$$\left\{ \begin{array}{l} \int_{\mathcal{F}(\theta^{n+1})} \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} \cdot \mathbf{v} + (\mathbf{u}^n \cdot \nabla) \mathbf{u}^{n+1} \cdot \mathbf{v} + \frac{\nu}{2} (\nabla \mathbf{u}^{n+1} + (\nabla \mathbf{u}^{n+1})^T) : (\nabla \mathbf{v} + \nabla \mathbf{v}^T) \\ \quad - p^{n+1} \operatorname{div} \mathbf{v} + \int_{\Gamma_i \cup \Gamma_w \cup \partial S(\theta^{n+1})} \boldsymbol{\lambda}^{n+1} \cdot \mathbf{v} = \int_{\mathcal{F}(\theta^{n+1})} \mathbf{f}_{\mathcal{F}}(t^{n+1}) \cdot \mathbf{v}, \\ \int_{\mathcal{F}(\theta^{n+1})} q \operatorname{div} \mathbf{u}^{n+1} = 0, \\ \int_{\Gamma_i \cup \Gamma_w \cup \partial S(\theta^{n+1})} \mathbf{u}^{n+1} \cdot \boldsymbol{\mu} = \int_{\Gamma_i} \mathbf{u}^i(t^{n+1}) \cdot \boldsymbol{\mu} + \int_{\partial S(\theta^{n+1})} \mathbf{u}_S(t^{n+1}) \cdot \boldsymbol{\mu}, \end{array} \right. \quad (10)$$

$$\forall (\mathbf{v}, q, \boldsymbol{\mu}) \in (\mathbf{H}^1(\mathcal{F}(\theta^{n+1}))) \times (L^2(\mathcal{F}(\theta^{n+1}))) \times (\mathbf{H}^{-1/2}(\partial S(\theta^{n+1})) \times \mathbf{H}^{-1/2}(\Gamma_i \cup \Gamma_w)).$$

A Euler scheme semi-discretizes the partial derivative w.r.t. time. It requires \mathbf{u}^n to be integrated over the domain $\mathcal{F}(\theta^{n+1})$ (while it is defined on $\mathcal{F}(\theta^n)$ only). We then need an extension procedure that we expose in Sect. 4.4. Note that the Dirichlet boundary conditions have been imposed in a weak way by the use of Lagrange multipliers $\boldsymbol{\lambda}^{n+1} = (\boldsymbol{\lambda}_{\partial S(\theta^{n+1})}^{n+1}, \boldsymbol{\lambda}_{\Gamma_i \cup \Gamma_w}^{n+1})$. This induces the computation of the additional variable $\boldsymbol{\lambda}_{\partial S(\theta^{n+1})}^{n+1} = -\sigma_F(\mathbf{u}^{n+1}, p^{n+1})\mathbf{n}_\theta$ which represents the fluid forces acting on the structure at time t^{n+1} and will be approximated by $\boldsymbol{\lambda}_h^{n+1}$. The index $\partial S(\theta^{n+1})$ is dropped in the sequel. The structure evolution is computed by the following finite difference method:

$$\left\{ \begin{array}{l} \theta^{n+1} = 2\theta^n - \theta^{n-1} + (\Delta t_{n+1})^2 M_\theta^{-1} (M_A(\theta^n, \boldsymbol{\lambda}_h^n) + M_I(\theta^n, \omega^n)), \end{array} \right. \quad (11)$$

$$\left\{ \begin{array}{l} \omega^{n+1} = \omega^n + \Delta t_{n+1} M_\theta^{-1} (M_A(\theta^n, \boldsymbol{\lambda}_h^n) + M_I(\theta^n, \omega^n)), \end{array} \right. \quad (12)$$

where we have denoted $\omega^n = (\omega_1^n, \omega_2^n)$ the approximation of $\dot{\theta}(t^n)$. Note that we use λ_h^n in the structure equations.

At each time step, we use the procedure described in Algorithm 1. The main difficulties are to compute the fluid step and to adapt the fluid domain, they are tackled respectively in Sects. 4 and 5.

Algorithm 1 The splitting scheme

Require: $(\mathbf{u}_h^n, p_h^n, \lambda_h^n, \theta^n, \theta^{n-1}, \omega^n, \Delta t_{n+1})$.

1 Compute $(\theta^{n+1}, \omega^{n+1})$ with the structure step (11)–(12).

2 Update $\mathcal{F}(\theta^{n+1})$.

3 Compute $(\mathbf{u}_h^{n+1}, p_h^{n+1}, \lambda_h^{n+1})$ with the fluid step.

4 Compute the next time step Δt_{n+2} with the CFL condition (15) (presented in Sect. 4.3).

4 The Discretization Method for the Fluid Equations

In this section we detail the way we approximate the evolution of the fluid state.

4.1 The Finite Element Method

We define a background mesh \mathbb{T}_h covering the whole domain $\Omega = \mathcal{F}(\theta) \cup \mathcal{S}(\theta)$. This mesh will not be modified during the simulation. The mesh cells can then be cut arbitrarily by the interface $\partial\mathcal{S}(\theta)$. This means that the cells are either entirely contained in the fluid or the structure domain, or are shared between those two domains. For each variable, we define a finite element method on the whole triangulation \mathbb{T}_h . We choose the use of Taylor–Hood elements, i.e. \mathbb{P}_2 elements for \mathbf{u}_h , \mathbb{P}_1 elements for p_h and for λ_h .

Then, the approximated functions are the trace of these polynomials on the physical domain, $\mathcal{F}(\theta)$ for \mathbf{u}_h and p_h , $\partial\mathcal{S}(\theta)$ for λ_h . We depict the functional basis associated to this method in Fig. 2.

Away from the interface, we have the usual finite elements associated to usual degrees of freedom (dof). Near the interface, the elements are cut. Note that some dof of the fluid are located in the structure domain, we call them *fictitious degrees of freedom* in the sequel. They correspond to actual dof of the method, however, their value do not have any physical meaning. This is why we need to handle those dof carefully (see Sect. 4.4). In the structure, away from the interface, all degrees of freedom are discarded.

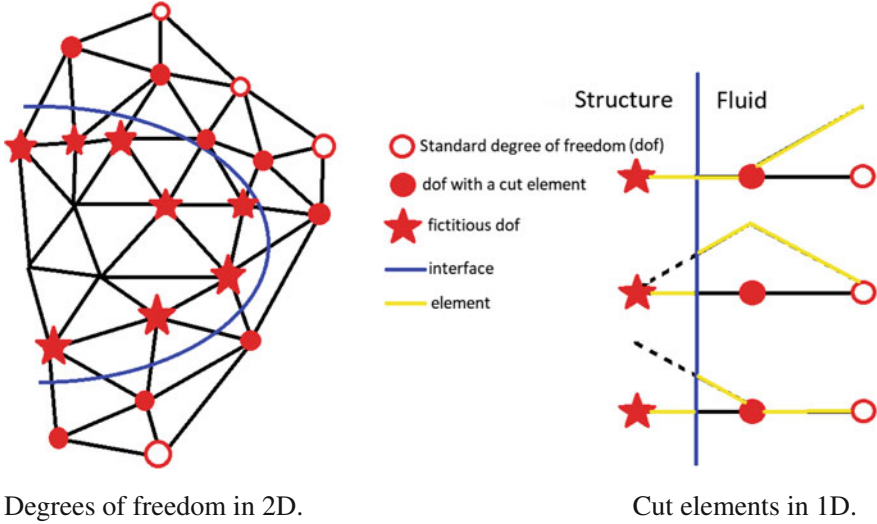


Fig. 2 The degrees of freedom for \mathbb{P}_1 finite elements

4.2 The Stabilization Term

Classical XFEM applied to the Navier–Stokes equations does not ensure optimal convergence rate for the Lagrange multipliers. However, to simulate accurately the dynamics of the structure, we want λ_h to be a good approximation of λ .

In order to recover the optimal convergence rate, we add the following stabilization term to the variational formulation of the fluid problem (10)

$$- \gamma_0 h \int_{\partial S(\theta)} (\lambda_h + \sigma_F(\hat{\mathbf{u}}_h, \hat{p}_h) \mathbf{n}_\theta) \cdot (\boldsymbol{\mu}_h + \sigma_F(\hat{\mathbf{v}}_h, \hat{q}_h) \mathbf{n}_\theta) \, d\mathbf{x}, \quad (13)$$

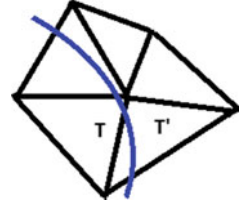
with a mesh-independent constant $\gamma_0 > 0$. It corresponds to an augmented Lagrangian approach [5].

If we choose $\hat{\mathbf{u}}_h = \mathbf{u}_h$, $\hat{p}_h = p_h$, $\hat{\mathbf{v}}_h = \mathbf{v}_h$ and $\hat{q}_h = q_h$, then we get optimal convergence rate for λ_h if the mesh does not have any *bad triangle*. We call bad triangle a triangle that is cut by the interface and has only a tiny part of its surface in the fluid domain. For instance, in Fig. 3, T is a bad triangle.

The coefficients of the matrices are computed by an integration of the basis functions on the fluid domain. Then, such bad elements with only a tiny part in the fluid domain induce tiny coefficients in the matrices that are then ill-conditioned.

Since we consider a fixed mesh that can be arbitrarily cut by the interface, this stabilization is not enough. For such bad triangles, we take the velocity and pressure terms $\hat{\mathbf{u}}_h$, \hat{p}_h , $\hat{\mathbf{v}}_h$ and \hat{q}_h in (13) as the extrapolation of the values of these variables in a good neighbor T' of T . Doing that way, we obtain optimal convergence for λ_h even if some triangles are badly cut in the mesh, see [11].

Fig. 3 A bad element T and a good neighbor T' . The fluid domain $\mathcal{F}(\theta)$ is on the right of the interface represented in blue



4.3 Algebraic Formulation

We denote with U , P and Λ the coordinate vectors of the velocity, the pressure and the multiplier into the \mathbb{P}_2 , \mathbb{P}_1 and \mathbb{P}_1 Finite Element (FE) basis respectively. The FE method can be rewritten under the following matrix form:

$$\frac{1}{\Delta t} \begin{pmatrix} M_{\mathbf{uu}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} U^{n+1} - U^n \\ P^{n+1} - P^n \\ \Lambda^{n+1} - \Lambda^n \end{pmatrix} + \begin{pmatrix} A_{\mathbf{uu}} & A_{\mathbf{up}} & A_{\mathbf{u}\lambda} \\ A_{\mathbf{up}}^T & A_{pp} & A_{p\lambda} \\ A_{\mathbf{u}\lambda}^T & A_{p\lambda}^T & A_{\lambda\lambda} \end{pmatrix} \begin{pmatrix} U^{n+1} \\ P^{n+1} \\ \Lambda^{n+1} \end{pmatrix} = \begin{pmatrix} F^{n+1} \\ 0 \\ G^{n+1} \end{pmatrix}. \tag{14}$$

All the matrices are given by the discretization of the terms induced by (10) and (13). The matrices $A_{\mathbf{uu}}$, $A_{\mathbf{up}}$, $A_{\mathbf{u}\lambda}$, $A_{\mathbf{up}}^T$ and $A_{\mathbf{u}\lambda}^T$ correspond respectively, up to a perturbation induced by the stabilization term, to $\nu \Delta \mathbf{u}$, ∇p , λ , the incompressibility constraint and the Dirichlet boundary conditions, $M_{\mathbf{uu}}$ is the mass matrix. The vectors F^{n+1} and G^{n+1} correspond to the source term and to the Dirichlet data at t^{n+1} . The matrices A_{pp} , $A_{p\lambda}$, $A_{p\lambda}^T$ and $A_{\lambda\lambda}$ would have been null if there were no stabilization terms. Since the fluid domain changes over time, we discard and compute all the matrices at every time step. Some optimizations can be made to update only the coefficients linked to the cut cells.

In order to have a stable scheme, we use the following CFL condition:

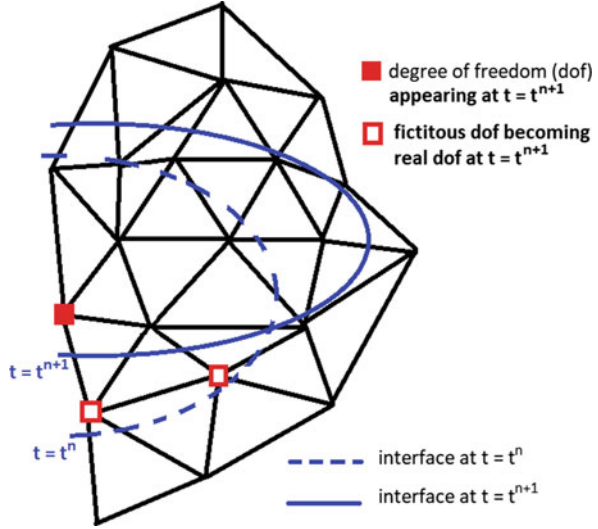
$$\Delta t_{n+1} = \min \left(\text{cfl} \times \frac{h}{V_{\max}^n}, \Delta t_{\max} \right), \tag{15}$$

where $\text{cfl} \in (0, 1)$, V_{\max}^n is the maximum velocity of the fluid in $\mathcal{F}(\theta^n)$ and Δt_{\max} is the maximum time step allowed.

4.4 Initialization of the Fictitious Velocity Values

The time derivative has been discretized in (14) by a finite difference method. This is a natural way of discretizing it. However, it induces some difficulties. Indeed, at every time step, in order to compute \mathbf{u}_h^{n+1} , we need to provide the value of \mathbf{u}_h^n at

Fig. 4 Initialization of the velocity dof. In order to tackle difficulties, all dof that are in $S(\theta^n)$ (the fictitious and the discarded ones) are given by the velocity of the structure at time t^n . Since we consider adherence conditions between the fluid and the structure, the velocity of the structure near the interface is close to the one of the fluid. This justifies the meaning of the value given to the nodes in $S(\theta^n)$



every dof considered for $\mathcal{F}(\theta^{n+1})$. The difficulty is due to the fact that such a value is not available when that degree of freedom was discarded in $\mathcal{F}(\theta^n)$ (see Fig. 4).

Moreover, as exposed above, the values of the fictitious degrees of freedom have to be carefully used since they do not have any physical meaning. Hence, a dof that is fictitious in $\mathcal{F}(\theta^n)$ and becomes real in $\mathcal{F}(\theta^{n+1})$ cannot be straightforwardly used. We then impose the velocity of the structure at t^n to all dof that are in $S(\theta^n)$.

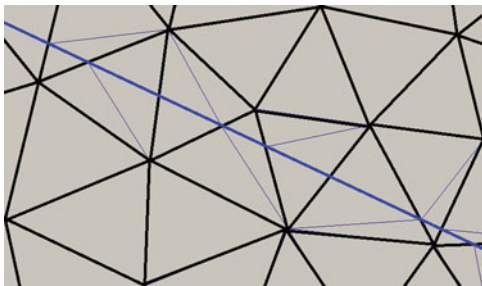
5 The Evolution of the Fluid Domain

In this section, we describe the method used to adapt the fluid domain at every time step. We first describe the integration over the cut cells, then we present the way we locate the interface between the fluid and the structure.

5.1 Integration over the Cut Cells

All matrices in (14) are computed with an integration on the fluid domain only. We then need to lead integrations on the mesh, in particular over cut cells. In order to make these integrations possible, we divide every cut cell in sub-triangles that are taken fitted to the interface. We then only have to integrate over the sub-triangles that are in the fluid domain (see Fig. 5). This step is implemented with the *qhull* library [3]. The method is classical, we recall it here for completeness of the presentation.

Fig. 5 Subdivision of the cut cells. Note that the operation consists in sub-dividing cut cells and not remeshing. Hence, the sub-cells are used only for integration and no new degrees of freedom are defined



This process requires a level-set function to locate the interface (as the null level of this function). We describe in Sect. 5.2 the way that we compute this function.

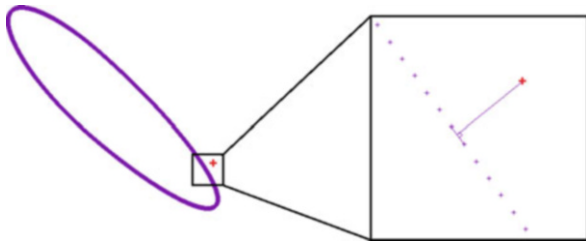
5.2 Position of the Interface

We locate the interface as the zero of a level-set function. In the literature, most of the time, the level-set function is computed as the solution of a PDE representing its evolution [23]. This generates undesirable effects such as numerical diffusion.

In our setting, the structure position and then the interface depends only on the two parameters $\theta = (\theta_1, \theta_2)$. At each time step, we then compute a precise approximation of the level-set function associated to the parameters (θ_1^n, θ_2^n) (see Fig. 6).

We can tune the number of points considered to gain more precision. The drawback of this method is that it is time expensive. In order to mitigate the run time, we compute the level-set function only for the required nodes, i.e. the ones near the interface. A smart function chooses those nodes.

Fig. 6 Computation of the level-set function. The interface is represented by a list of points (computed analytically). The distance is computed with the polygon formed by those points



6 Numerical Simulations

In this section we detail our choices for the numerical simulations and show some results. All simulations are run with the *GetFEM++* library [21].

6.1 The Structure Modelling

To represent the deformations of the structure, we introduce a diffeomorphism $\mathbf{X}(\theta_1, \theta_2, \cdot)$ that transforms a reference configuration corresponding to $\theta_{\text{ref}} = (0, 0)$ into a given configuration corresponding to $\theta = (\theta_1, \theta_2)$ (see Fig. 7).

More precisely, we can define this diffeomorphism by

$$\mathbf{X}(\theta_1, \theta_2, \mathbf{x}) = \begin{pmatrix} \left(g_1(x_1) + x_2 \frac{N_1(x_1)}{|\mathbf{N}(x_1)|} \right) \cos(\theta_1) - \left(g_2(x_1) + x_2 \frac{N_2(x_1)}{|\mathbf{N}(x_1)|} \right) \sin(\theta_1) \\ \left(g_1(x_1) + x_2 \frac{N_1(x_1)}{|\mathbf{N}(x_1)|} \right) \sin(\theta_1) + \left(g_2(x_1) + x_2 \frac{N_2(x_1)}{|\mathbf{N}(x_1)|} \right) \cos(\theta_1) \end{pmatrix},$$

where $\mathbf{N}(x_1) = (N_1(x_1), N_2(x_1)) = (-g'_1(x_1), g'_2(x_1))$,

$$g_1(\ell) = \begin{cases} \ell & \text{if } \ell \leq a, \\ a + (\ell - a) \cos(\theta_2/2) - f(\ell) \sin(\theta_2/2) & \text{if } \ell \in (a, b), \\ x_{B'} + (\ell - b) \cos(\theta_2) & \text{if } \ell \geq b, \end{cases}$$

$$g_2(\ell) = \begin{cases} 0 & \text{if } \ell \leq a, \\ (\ell - b) \sin(\theta_2/2) + f(\ell) \cos(\theta_2/2) & \text{if } \ell \in (a, b), \\ y_{B'} + (\ell - b) \sin(\theta_2) & \text{if } \ell \geq b, \end{cases}$$

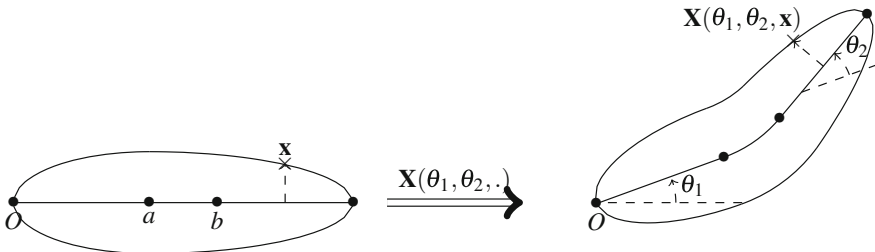


Fig. 7 The diffeomorphism \mathbf{X}

for $x_{B'} = b + (b - a) \cos(\theta_2/2)$, $y_{B'} = (b - a) \sin(\theta_2/2)$ and

$$f(\ell) = \frac{\tan(\theta_2/2)}{b - a} (\ell - (a + b)/2)^2 - \tan(\theta_2/2) \frac{b - a}{4}.$$

We use the values $a = 0.4$ and $b = 0.6$ in the sequel. To complete (1)–(9), we define

$$\mathbf{u}_S(t, \mathbf{x}) = \dot{\theta}_1 \partial_{\theta_1} \mathbf{X}(\theta_1, \theta_2, \mathbf{x}) + \dot{\theta}_2 \partial_{\theta_2} \mathbf{X}(\theta_1, \theta_2, \mathbf{x}), \quad (16)$$

$$(M_\theta)_{ij} = (\partial_{\theta_i} \mathbf{X}(\theta_1, \theta_2, \cdot), \partial_{\theta_j} \mathbf{X}(\theta_1, \theta_2, \cdot))_S, \quad (17)$$

$$(M_I)_j = -\left(\dot{\theta}_1^2 \partial_{\theta_1}^2 \mathbf{X}(\theta_1, \theta_2) + 2\dot{\theta}_1 \dot{\theta}_2 \partial_{\theta_1 \theta_2} \mathbf{X}(\theta_1, \theta_2) + \dot{\theta}_2^2 \partial_{\theta_2}^2 \mathbf{X}(\theta_1, \theta_2), \partial_{\theta_j} \mathbf{X}(\theta_1, \theta_2) \right)_S, \quad (18)$$

$$(M_A(\theta, \mathbf{f}))_j = \int_{\partial S(\theta)} \mathbf{f} \cdot \partial_{\theta_j} \mathbf{X}(\theta_1, \theta_2, \mathbf{Y}(\theta_1, \theta_2, \mathbf{x})), \quad (19)$$

where $(\mathbf{f}, \mathbf{g})_S = \rho \int_{\partial S(0)} \mathbf{f} \cdot \mathbf{g}$ and $\mathbf{Y}(\theta_1, \theta_2, \cdot)$ is the inverse diffeomorphism of $\mathbf{X}(\theta_1, \theta_2, \cdot)$.

We have proven well-posedness of the problem (1)–(9) with (16)–(19) in [7]. The reader will find there further information about this model.

6.2 Numerical Results

The whole domain $\Omega = (-1.0, 8.0) \times (y_{min}, y_{max})$ with $y_{min} = -2.5$ and $y_{max} = 2.1$ is discretized by a triangular mesh of 35,731 cells. It is locally refined near its boundary, near the corners, in the zone where lies the structure and in the wake behind the structure. We do not consider any forces in the fluid $\mathbf{f}_{\mathcal{F}} = 0$. The inflow condition is a perturbed Poiseuille profile $\mathbf{u}^i(t, x_2) = \frac{6U_m}{(y_{max} - y_{min})^2} (-x_2^2 + (y_{max} + y_{min})x_2 - y_{min}y_{max}) - z_p(x_2)e^{-(t-0.5)^2}$, where $U_m = 1.0$ and $z_p(x_2) = 0.8 \sin(2\pi(x_2 + 0.75)/1.5)$ if $x_2 \in [-0.75, 0.75]$ and $z_p(x_2) = 0$ elsewhere is a profile chosen to perturb the stationary configuration.

We use the parameters $\nu = 1/120$ (Reynolds number of 120), $\rho = 5$, $\gamma_0 = 0.05$, $\text{cfl} = 0.8$ and $\Delta t_{max} = 5 \cdot 10^{-4}$.

The initial parameters for the structure are $\theta_0 = (-20^\circ, 0)$ and $\omega_0 = (0, 0)$. The initial velocity profile \mathbf{u}_0 is obtained by solving the stationary Navier–Stokes equations in the initial configuration. The source on the structure \mathbf{f}_S is chosen to compensate the forces of the fluid in the initial configuration and then enforce the initial state to be a stationary state.

The evolution of the system is reported in Figs. 8 and 9. The stationary setting is perturbed by the inflow condition. The perturbation propagates in the fluid and destabilizes the structure. Von Kármán vortex street appears in the wake behind the structure.

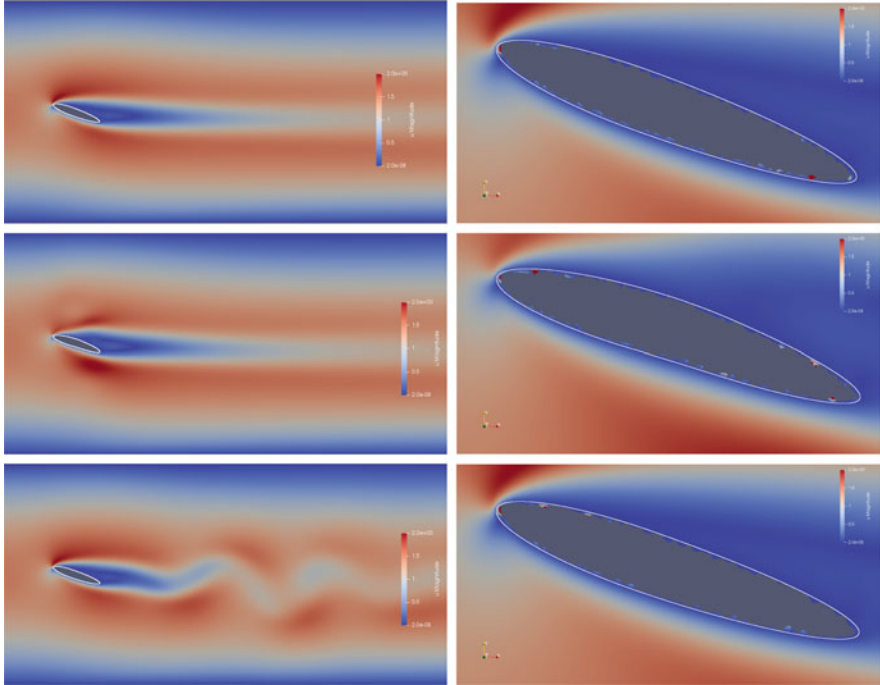


Fig. 8 The velocity magnitude profile (red: high, blue: low), left: whole domain, right: zoom on the structure. The pictures have been captured at $t = 0s$, $t = 1.5s$ and $t = 6s$

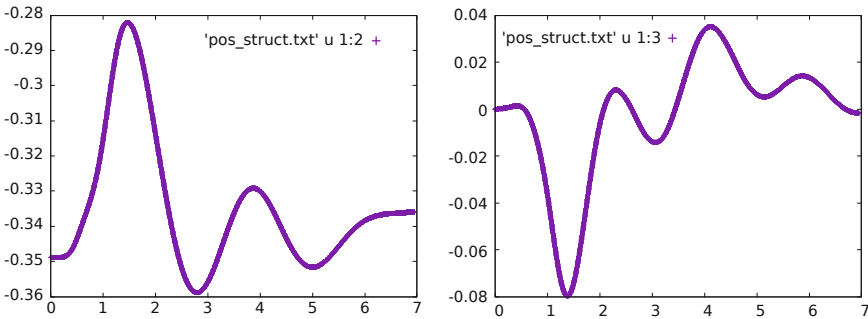


Fig. 9 The evolution of the structure parameters; left: θ_1 , right: θ_2

7 Summary

In the present article, we proposed a method to compute numerically the evolution of a fluid–structure system with a fixed mesh. It is based on a finite difference method w.r.t. time and on XFEM w.r.t. space. The structure and fluid steps are partitioned. We presented some solutions to implement this method. We finally ran numerical

simulations with small deformations of the structure. Since the advantage of using XFEM over ALE is to tackle large deformations, more complex test cases should be considered to fully validate this approach.

Acknowledgements The first author has been supported by IFSMACS ANR-15-CE40-0010.

References

1. Alauzet, F., Fabrèges, B., Fernández, M.A., Landajuela, M.: Nitsche–XFEM for the coupling of an incompressible fluid with immersed thin–walled structures. *Comput. Methods Appl. Mech. Eng.* **301**, 300–335 (2016)
2. Angot, P., Bruneau, C.-H., Fabrie, P.: A penalization method to take into account obstacles in incompressible viscous flows. *Numer. Math.* **81**(4), 497–520 (1999)
3. Barber, C.B., Dobkin, D.P., Huhdanpaa, H.: The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* **22**(4), 469–483 (1996)
4. Bordas, S.P.A., Burman, E., Larson, M.G., Olshanskii, M.A.: *Geometrically Unfitted Finite Element Methods and Applications*. Springer, Berlin (2017)
5. Court, S., Fournié, M., Lozinski, A.: A fictitious domain approach for the Stokes problem based on the extended finite element method. *Int. J. Numer. Methods Fluids* **74**(2), 73–99 (2014)
6. Delay, G.: Étude d’un problème d’interaction fluide–structure: Modélisation, Analyse, Stabilisation et Simulations numériques. Ph.D. thesis. Université de Toulouse (2018)
7. Delay, G.: Existence of strong solutions to a fluid–structure system with a structure given by a finite number of parameters. Available on <https://www.math.univ-toulouse.fr/~gdelay/> (2020)
8. Dillon, R.H., Li, Z.: An introduction to the immersed boundary and the immersed interface methods — from Interfaces problems and methods in biological and physical flows. *Lect. Notes Ser. Inst. Math. Sci. Natl. Univ. Singap.* **17**, 1–67 (2009)
9. Dolbow, J., Moës, N., Belytschko, T.: An extended finite element method for modeling crack growth with frictional contact. *Comput. Methods Appl. Mech. Eng.* **190**(51–52), 6825–6846 (2001)
10. Fernández, M.A., Mullaert, J., Vidrascu, M.: Generalized Robin–Neumann explicit coupling schemes for incompressible fluid–structure interaction: stability analysis and numerics. *Int. J. Numer. Methods Eng.* **101**(3), 199–229 (2014)
11. Fournié, M., Lozinski, A.: Stability and optimal convergence of unfitted extended finite element methods with Lagrange multipliers for the stokes equations. In: Bordas, S.P.A., Burman, E., Larson, M.G., Olshanskii, M.A. (eds.) *Geometrically Unfitted Finite Element Methods and Applications*, pp. 143–182. Springer, Berlin (2017)
12. Hou, G., Wang, J., Layton, A.: Numerical methods for fluid–structure interaction – a review. *Commun. Comput. Phys.* **12**(2), 337–377 (2012)
13. Hu, H.H., Patankar, N.A., Zhu, M.Y.: Direct numerical simulations of fluid–solid systems using arbitrary Lagrangian Eulerian technique. *J. Comput. Phys.* **169**, 427–462 (2001)
14. Kamensky, D., Hsu, M.-C., Schillinger, D., Evans, J.A., Aggarwal, A., Bazilevs, Y., Sacks, M.S., Hughes, T.J.R.: An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves. *Comput. Methods Appl. Mech. Eng.* **284**, 1005–1053 (2015)
15. Landajuela, M., Vidrascu, M., Chapelle, D., Fernández, M.A.: Coupling schemes for the FSI forward prediction challenge: comparative study and validation. *Int. J. Numer. Methods Biomed. Eng.* **33**(4), pp.e02813 (2017)
16. Lefebvre, A.: Numerical simulation of gluey particles. *M2AN Math. Model. Numer. Anal.* **43**(1), 53–80 (2009)

17. Massing, A., Larson, M.G., Logg, A., Rognes, M.E.: A Nitsche-based cut finite element method for a fluid–structure interaction problem. *Commun. Appl. Math. Comput. Sci.* **10**(2), 97–120 (2015)
18. Maury, B.: Numerical analysis of a finite element/volume penalty method. *SIAM Numer. Anal.* **47**(2), 1126–1148 (2009)
19. Mittal, R., Iaccarino, G.: Immersed boundary methods. *Annu. Rev. Fluid Mech.* **37**, 239–261 (2005)
20. Peskin, C.S.: The immersed boundary method. *Acta Numer.* **11**, 476–517 (2002)
21. Renard, Y., Pommier, J.: Getfem finite element library. <http://home.gna.org/getfem/>
22. Richter, T.: A monolithic geometric multigrid solver for fluid–structure interactions in ALE formulation. *Int. J. Numer. Methods Eng.* **104**(5), 372–390 (2015)
23. Sethian, J.A.: *Level Set Methods – From Volume 3 of Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge (1999)