A Finite Element Penalized Direct Forcing Immersed Boundary Method for infinitely thin obstacles in a dilatable flow

Georis Billo, Michel Belliard, Pierre Sagaut

▶ **To cite this version:**

Georis Billo, Michel Belliard, Pierre Sagaut. A Finite Element Penalized Direct Forcing Immersed Boundary Method for infinitely thin obstacles in a dilatable flow. Computers & Mathematics with Applications, 2021, 99, pp.292-304. 10.1016/j.camwa.2021.08.005 . hal-03596009

HAL Id: hal-03596009
https://hal.science/hal-03596009v1

Submitted on 3 Mar 2022

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.

Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

A Finite Element Penalized Direct Forcing Immersed Boundary Method for infinitely thin obstacles in a dilatable flow

Georis Billo [a,*], Michel Belliard [a], Pierre Sagaut [b]

[a] French Alternative Energies and Atomic Energy Commission (CEA), DES, IRESNE, DER, SESI, LEMS, Cadarache, F-13118 Saint-Paul-Lez-Durance, France
[b] Aix-Marseille University, CNRS, Centrale Marseille, M2P2, Marseille, France

ARTICLE INFO

Keywords:
Navier-Stokes equations
Finite Element Method
Immersed Boundary Method
Direct Forcing Method
Penalty
Projection scheme

ABSTRACT

In the framework of the development of new passive safety systems for the second and third generations of nuclear reactors, the numerical simulations, involving complex turbulent two-phase flows around thin or massive inflow obstacles, are privileged tools to model, optimize and assess new design shapes. In order to match industrial demands, computational fluid dynamics tools must be the fastest, most accurate and most robust possible. To face this issue, we have chosen to solve the Navier-Stokes equations using a projection scheme for a mixture fluid coupled with an Immersed Boundary (IB) approach: the penalized direct forcing method – a technique whose characteristics inherit from both penalty and immersed boundary methods – adapted to infinitely thin obstacles and to a Finite Element (FE) formulation. Various IB conditions (slip, no-slip or Neumann) for the velocity on the IB can be managed by imposing Dirichlet values in the vicinity of the thin obstacles. To deal with these imposed Dirichlet velocities, we investigated two variants: one in which we use the obstacle velocity and another one in which we use linear interpolations based on discrete geometrical properties of the IB (barycenters and normal vectors) and the FE basis functions. This last variant is motivated by an increase of the accuracy/computation time ratio for coarse meshes. As a first step, concerning academic test cases for one-phase dilatable-fluid laminar flows, the results obtained via those two variants are in good agreement with analytical and experimental data. Moreover, when compared to each other, the linear interpolation variant increases the spatial order of convergence as expected. An industrial test case illustrates the advantages and drawbacks of this approach. In a shortcoming second step, to face two-phase turbulent fluid simulations, some methodology modifications will be considered such as adapting the projection scheme to low-compressible fluid and immersed wall-law boundary conditions.

1. Introduction

Many engineering works concern applications involving numerical flow simulations with fixed or moving boundaries. In case of moving boundaries (fluid-structure interaction, flows induced by a stirrer, *etc.*), the fluid computation mesh must be rebuilt at each modification of the boundaries. This can lead to time consuming simulations which is also true when considering a geometric shape optimization process for fixed boundaries. In this case, a large number of computations, each using a particular geometry, may be needed to build a statistic answer surface in order to determine the optimized shapes. To overtake this issue, methods allowing the decoupling of the computation mesh from the fluid domain can be very useful; notably the Fictitious Domain (FD) approach, originally introduced by V.K. Saul'ev [1], which has been investigated in many various domains of industrial interest such as tire design [2], fluid-transported solid particles [3], nuclear-waste vitrification [4], *etc.* In the present paper, this approach has been chosen to carry out flow simulations around "infinitely thin" obstacles in the field of nuclear Pressurized Water Reactor (PWR) safety systems.

The increasing performance and safety requirements for the third generation of nuclear reactors led to new research and development studies. In this context, new innovative systems, especially based on hydraulic diodes, are designed to prevent or mitigate potential accidental or incidental situations. An example of the use of the hydraulic diode, to postpone PWR core dewatering in case of a large-break Loss Of Coolant Accident (LOCA), is what we call the advanced water accumulator for which a change in the discharge regime is leaded by a flow vortex creation [5]. In this paper, another passive system, imagined and patented

* Corresponding author.
E-mail address: georis.billo@cea.fr (G. Billo).

by CEA [6], is considered: the flow limiter. It is composed of fins which, during normal operation, let the regular flow heading into the vessel quite unchanged but, in accidental behavior (for instance fluid coming out from the vessel due a break on the cold leg), force the creation of a large vortex of fluid and so reduce the flowrate going outflow of the vessel (*i.e.* postpone the core dewatering). For more information about the background, the interested reader can refer to [7,8].

As a whole, the aimed application induces a compressible two-phase flow at low Mach number and with a thermodynamic disequilibrium. Keeping in mind the statistical optimization goal, a Homogeneous Equilibrium Model (HEM) with disequilibrium closure laws [9] should be considered to preserve low computational times while taking into account two-phase aspects. However, for the sake of simplicity of the presentation, only one-phase dilatable Navier-Stokes equations are considered in this paper, (but two-phase flow illustration can be found in [8]).

From this point of view, projection schemes are often used to deal with those equations [10] at low Mach number. Moreover, most of CEA codes involving Computational Fluid Dynamics (CFD) use those fractional-step techniques.

At last, the flow regime, together with the complex geometry of the devices, also indicate that the turbulence will be a major phenomenon but will not be fully treated in this first approach – it will have to in future models – as only a scalar Schlichting's model approach [11] is considered here.

In order to model the thin fins which compose passive safety systems such as the flow-limiter (*cf.* Section 1 while preserving low computation times, the FD approach, widely used for flows around complex geometries, has been chosen. Since this approach was introduced, many techniques have been proposed in different fields of physics. In any case, the common main idea, which is pretty well summarized by K. KHADRA et al. in the context of the Navier-Stokes equations [12], is to solve Partial Derivative Equations (PDE) on a much simpler computation domain Ω (for instance a parallelepipedal box in 3D), called "fictitious domain", in which the physical domain Ω_f is embedded. Here, Ω_f and Ω_s are two subsets embedded in the FD Ω such as $\overline{\Omega} = \overline{\Omega}_f \cup \overline{\Omega}_s$ and $\dim(\overline{\Omega}_f \cap \overline{\Omega}_s) = \dim(\Omega) - 1$. The interface between $\overline{\Omega}_f$ and $\overline{\Omega}_s$ – which can be a union of interfaces – is denoted $\Gamma = \partial\Omega_f \cap \partial\Omega_s \setminus \partial\Omega$. This approach allows using simple mesh types and fast methods such as, for instance, Cartesian grid with an ijk finite difference scheme.

The presence of the physical immersed boundaries or interfaces Γ is then taken into account by modifying the PDE on the fictitious domain or the numerical scheme used to solve these equations. A way to classify them is to discriminate between the ones which add new terms in the governing equations and the ones that modify the numerical operators. Without any claim to be exhaustive, we can mention:

- *Adds new terms:* the Lagrange Multipliers method [13,3,14], the Penalty method [15–17], the Fat Boundary Method [18,19], the Immersed Spread Interface method [20,21], the Immersed Boundary Method [22–24], the diffuse domain approach [25], *etc.*
- *Modifies the operators:* the Immersed Interface Method [26–28], the Ghost [29–31] and Cut Cell methods [32], the Cartesian Grid Embedded Boundary method [33–35], the Jump Embedded Boundary Conditions [36,21], the Finite Cell Method [37], the X-FEM-based FDMs [38], and so on.

For the aimed application, the class of Immersed Boundary Methods (IBM) has been chosen. Its original version was introduced by C. S. PESKIN to model cardiac mechanics and blood flow [22]. The idea was to take into account elastic boundaries via a backmoving force imposed at the interface by the mean of Dirac delta functions. This force is considered as a source term added to the governing equations at a continuous level. At a discrete level, this approach implies two meshes: a volumetric one – often Cartesian – on which the Eulerian variables are computed (fluid velocity, pressure, density, *etc.*) and a curvilinear one – corresponding to the immersed boundary – on which Lagrangian variables are computed (velocity and position of the immersed boundary, backmoving force, *etc.*). The backmoving force computed at the interface is then distributed over several cells on each side of the boundary by the mean of discrete Dirac delta functions. The interested reader can refer to [23] for a more precise explanation, a review of different techniques and many references about this topic. This method is quite simple to implement and it preserves the use of a Cartesian Eulerian grid. However, the application of this method to immersed rigid boundaries leads to numerical instabilities.

Then J. MOHD-YUSOF developed a new IBM, based on the one of C.S. PESKIN, often referred as "Direct Forcing" [39,40]. In his method, the local forcing term is added in the equations at a semi-discrete level (discrete time and continuous space) and is constructed so that it cancels the inertia, viscous, pressure and Right-Hand-Side (RHS) terms along Γ, leaving only a backmoving force which tends to bring the value of the computed velocity back to value of the immersed Dirichlet BC. However, in practice, first, the forcing term is added at the discrete level on the degrees of freedom of Ω_f near Γ and, second, the forcing term cannot be computed implicitly (as it needs information about the values of the PDE terms). This implies two steps: one calculation without considering any boundary and another with the forcing term added to the RHS.

To preserve an implicit forcing term, M. BELLIARD et al. developed a new technique, called Penalized Direct Forcing (PDF) [41], inspired by the Direct Forcing methods and penalization techniques [16,15,42,17]. The PDF method has been applied to the incompressible Navier-Stokes equations in the framework of the finite volume method using a finite difference scheme [43]. Note that the PDF method is also coupled with a multi-directional linear interpolation of the imposed velocity in the vicinity of the immersed boundary in [43]. Generally speaking, data reconstruction on a boundary/interface or in its vicinity is a key issue in the scope of fictitious domain methods. Thus, various techniques have been developed in order to reconstruct fields coming from a boundary on the computation domain (*i.e.* the computation of the velocity imposed by the IB in our case) or the contrary (reconstruction of stress on the envelope of an airfoil to compute aerodynamic forces for instance [44]). In this paper, we focus more specifically on interpolation techniques, widely used in all kind of fictitious domain methods [40,43,45–47]. Those techniques, aside from involving polynomials or spline functions, can rather be directional (1D) [40] or spatial (multi-D) [43]. Those two variants are not used in the same frameworks and do not require the same type of information about the boundary/interface.

In this paper, we propose an adaptation of the PDF method to a FE framework. Furthermore, as we plan, in near future, to use turbulent wall laws in order to interpolate the imposed velocity in the vicinity of the immersed boundary, we also propose, as a first step, a FE-based directional linear interpolation of the imposed velocity in the vicinity of the immersed boundary. After the presentation of the time and space discretization of the governing equations in Section 2, the adaptation of the PDF method to the fractional-step algorithm and the FE formulation is discussed in Section 3, as well as the directional linear interpolation technique. Numerical results concerning various academic 2D test cases and a quasi-industrial 3D test case for one-phase dilatable-fluid flows are the object of Section 4. Finally, the paper is concluded and perspectives are outlined.

2. Preliminaries

2.1. Governing equations

As explained in Section 1, a HEM is considered for the two-phase flow [48]. Hence the governing equations reduce to the Navier-Stokes equations for an equivalent mixture with non-linear coefficients and

source terms to take disequilibrium into account [9]. The mixture Equation Of State (EOS) is deduced from each phase EOS, given some hypothesis. Moreover, for the shake of computational efficiency and looking for stationary time-averaged flow regimes, we model a dilatable mixture fluid, neglecting the density waves ($\partial_t \rho \equiv 0$) of a compressible fluid flow at low Mach number. Obviously these mixture balance equations boil down to the standard Navier-Stokes equations for one-phase incompressible flow.

Equally, in this paper, we assume that the viscous stress $\bar{\bar{\sigma}}$ only depends on the dynamic viscosity μ and on the strain rate tensor $\bar{\bar{\varepsilon}}$, which is defined as follows:

$$\bar{\bar{\varepsilon}} = \frac{1}{2}\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right) \tag{1}$$

with $\mathbf{u}$ the velocity of the fluid. Hence $\bar{\bar{\sigma}} = \mu \bar{\bar{\varepsilon}}$ with μ given by one-phase or mixture EOS or by a turbulent viscosity model.

2.2. Time discretization

Let denote $\delta t \in \mathbb{R}^{+*}$ the adaptive time step and $N_T \in \mathbb{N}$ the number of time steps. Given $\Omega \subseteq \mathbb{R}^d$ a d-dimensional open compact domain with a piecewise regular boundary denoted $\partial\Omega$ such as $\overline{\Omega} = \Omega \cup \partial\Omega$ and the sequence associated to the discrete time steps $(t_n)_{n<N_T+1}$, we define $\mathbf{u}^n : \Omega \to \mathbb{R}^d$ (resp. $p^n : \Omega \to \mathbb{R}$, $\rho^n : \Omega \to \mathbb{R}$ and $\bar{\bar{\sigma}}^n : \Omega \to \mathbb{R}^{d\times d}$) as the approximations of the velocity (resp. the pressure, the density and the viscous stress tensor) at time t_n. In order to enhance the readability of this document, the inertia and viscous terms are gathered together by introducing the notations: $\bar{\bar{\Xi}}^{n,n+1} = \rho^n \mathbf{u}^n \otimes \mathbf{u}^{n+1} - \bar{\bar{\sigma}}^{n+1}$ and $\bar{\bar{\Sigma}}^{n,n+1} = \bar{\bar{\Xi}}^{n,n+1} + \bar{\bar{I}} p^n$. Considering a semi-implicit scheme with time-implicit diffusive and (linearized) advective terms, the semi-discrete system can be written as follows:

$$\begin{cases} \delta t^{-1}\rho^n(\mathbf{u}^{n+1} - \mathbf{u}^n) + \nabla \cdot \bar{\bar{\Xi}}^{n,n+1} + \nabla p^{n+1} = \rho^n \mathbf{g} & \text{on } \Omega \\ \nabla \cdot (\rho^n \mathbf{u}^{n+1}) = 0 & \text{on } \Omega \\ +\text{BC} & \text{on } \partial\Omega \\ +\text{IC} & \text{on } \Omega \end{cases} \tag{2}$$

In this system of equations, we have assumed that the physical quantities as the density, the dynamic viscosity, *etc.* are computed with EOS involving the main variables $\mathbf{u}$ and p at the previous time step and are kept constant during the current time step.

2.3. Fractional-step algorithm

Projection schemes [10] are often used to solve the incompressible Navier-Stokes equations and are based on the Helmholtz-Hodge theorem. Here, we present its extension to dilatable fluid ($\nabla \cdot \rho\mathbf{u} = 0$). It reads as follows:

1. *Prediction*: only time, inertia, viscous and source terms are considered, the pressure term is kept at the previous time step. An intermediate velocity, called predicted and denoted $\mathbf{u}^*$ is computed.
$$\frac{\rho^n}{\delta t}\mathbf{u}^* + \nabla \cdot \bar{\bar{\Sigma}}^{n,*} = \rho^n \mathbf{g} + \frac{\rho^n}{\delta t}\mathbf{u}^n \tag{3}$$
2. *Projection*: the pressure corrector field is computed using the predicted velocity and mass balance equation.
$$\frac{1}{\delta t}\rho^n\left(\mathbf{u}^{n+1} - \mathbf{u}^*\right) + \nabla \phi^{n+1} = 0 \tag{4}$$
$$\nabla \cdot (4) \Rightarrow \Delta \phi^{n+1} = \frac{1}{\delta t}\nabla \cdot \left(\rho^n \mathbf{u}^*\right) \tag{5}$$
3. *Correction*: the velocity is computed using the pressure corrector gradient.
$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\delta t}{\rho^n}\nabla \phi^{n+1} \quad \text{and} \quad p^{n+1} = p^n + \phi^{n+1} \tag{6}$$

with $\mathbf{u}^*$ a provisional variable called the predicted velocity and $\phi^{n+1} = p^{n+1} - p^n$ the pressure corrector.

2.4. Finite Element formulation

The computation domain Ω is divided in $N_E \in \mathbb{N}$ hexahedral elements, each denoted K_e with $e \in [\![1, N_E]\!]$ and Ω_e the portion of Ω associated to the element K_e. Those elements are composed of nodes and the total number of nodes is denoted N_N. The mixed FEM is used, which means that the discrete unknowns of the problem are decomposed in two different FE basis. For the velocity, a $\mathbb{Q}_1$ basis (*i.e.* trilinear decomposition at nodes) is used while, for pressure, a $\mathbb{Q}_0$ basis (*i.e.* the discrete pressure field is constant by element) is used. This pair of elements is known to be unstable and can induce checkerboard pattern for the pressure but it is kept for two main reasons: i) it counts very few degrees of freedom per element – 8 for velocity and 1 for pressure compared to 27 and 8 when using a $\mathbb{Q}_2$–$\mathbb{Q}_1$ pair which, for instance, is stable – so it is quite fast; ii) the pressure instabilities are softened when the diffusivity is high enough – *i.e.* in laminar cases or when the turbulent viscosity is high, which is a case of interest for safety passive system design. Let us denote $\mathbf{M}_e$ (resp. $\mathbf{D}_e$, $\mathbf{N}_e$ and $\mathbf{B}_e$) the lumped (*cf.* Section 3.2) mass (resp. diffusivity, advective and gradient-divergence) elemental matrix, λ_e^n (resp. λ_e^* and g_e) the components of the velocity (resp. predicted velocity and gravity vector) in the $\mathbb{Q}_1$ finite elements basis, ϕ_e^{n+1} (resp. ρ_e^n) the discrete pressure corrector (resp. fluid density) in an element K_e (*i.e.* $\mathbb{Q}_0$) and, finally, λ_Γ^{n+1} the decomposition of the imposed velocity $\mathbf{u}_\Gamma^{n+1}$ in the $\mathbb{Q}_1$ finite elements basis. Then, the space-discrete versions of the prediction equation (3) and projection equation (4) can be written as follows:

$$\left(\frac{1}{\delta t}\mathbf{M}_e\rho_e^n + \mathbf{D}_e\mu_e^n + \mathbf{N}_e\rho_e^n\right)\lambda_e^* = \frac{1}{\delta t}\mathbf{M}_e\rho_e^n\lambda_e^n + \mathbf{B}_e p_e^n + \mathbf{M}_e\rho_e^n g_e \tag{7}$$

$$\frac{1}{\delta t}\mathbf{M}_e\rho_e^n\left(\lambda_e^{n+1} - \lambda_e^*\right) = \mathbf{B}_e\phi_e^{n+1} \tag{8}$$

Moreover, let us notice that the general idea of the PDF method (*cf.* Section 3) does not depend strongly on the space discretization. If needed, it could be adapted to another pair of elements with little effort and without changing the philosophy behind it. For general information about solving the Navier-Stokes equations with the FEM, one can refer to [49]. For more detailed information about the Finite Element formulation used in this paper, the interested reader can refer to [9].

3. Penalized Direct Forcing method

3.1. Adaptation to the projection scheme

In the context of the dilatable Navier-Stokes equations considered in this paper, the forcing term has to be modified (in comparison to the one presented in [43] which considers an incompressible fluid) in order to preserve its good mathematical properties. It is defined as follows:

$$\mathbf{f}^{n+1} := \frac{\chi}{\eta\delta t}\rho^n\left(\mathbf{u}_\Gamma^{n+1} - \mathbf{u}^{n+1}\right) \tag{9}$$

with $\eta \in \mathbb{R}^{+*}$ (such as $\eta \ll 1$) the penalty parameter, $\chi : \Omega \to \{0,1\}$ the characteristic function of the immersed boundary and $\mathbf{u}_\Gamma^{n+1}$ the velocity imposed by the immersed boundary (or immersed Dirichlet boundary condition). This formulation provides two advantages: it is implicit due to the penalization and has a suitable mathematical formulation that allows an easy adaptation to the projection of fractional-step schemes.

When using a projection scheme together with IBM, one has shown that the forcing term should be split between the prediction and correction equations in order to preserve the accuracy of the method [45,50,51]. Inspired by the work of M. Belliard and C. Fournier [41], the following splitting is defined ($\mathbf{f}_P^{n+1} + \mathbf{f}_C^{n+1} = \mathbf{f}^{n+1}$):

$$\mathbf{f}_P^{n+1} := \frac{\chi}{\eta\delta t}\rho^n\left(\mathbf{u}_\Gamma^{n+1} - \mathbf{u}^*\right) \tag{10}$$

$$\mathbf{f}_C^{n+1} := \frac{\chi}{\eta\delta t}\rho^n\left(\mathbf{u}^* - \mathbf{u}^{n+1}\right) \tag{11}$$

where $\mathbf{f}_P^{n+1}$ and $\mathbf{f}_C^{n+1}$ are the parts of the forcing term respectively added during the prediction and correction steps.

Those two forcing terms (*cf.* equations (10) and (11)) are respectively added in equations (3) and (4). Then, the fractional-step scheme becomes:

$$\frac{\rho^n}{\delta t}(\mathbf{u}^* - \mathbf{u}^n) + \nabla \cdot \bar{\bar{\Sigma}}^{n,*} = \rho^n \mathbf{g} + \frac{\chi}{\eta \delta t} \rho^n \left(\mathbf{u}_\Gamma^{n+1} - \mathbf{u}^*\right) \tag{12}$$

$$\frac{\rho^n}{\delta t}\left(\mathbf{u}^{n+1} - \mathbf{u}^*\right) + \nabla \phi^{n+1} = \frac{\chi}{\eta \delta t} \rho^n \left(\mathbf{u}^* - \mathbf{u}^{n+1}\right) \tag{13}$$

and, by gathering terms together in the correction equation (13), we obtain:

$$\frac{\rho^n}{\delta t}\mathbf{u}^{n+1} + \frac{\eta}{\eta + \chi}\nabla \phi^{n+1} = \frac{\rho^n}{\delta t}\mathbf{u}^* \tag{14}$$

However, the characteristic function χ is discontinuous among the immersed boundary Γ. Therefore, the divergence cannot be directly applied to (14) to get the projection equation. The divergence is applied at a space-discrete level (*cf.* Section 3.2).

Beyond the preservation of the accuracy of the fractional-step method, equation (14) provides a way to isolate $\nabla \phi^{n+1}$ on one side of the immersed obstacle. It is similar to the standard Neumann condition $\nabla(p^{n+1} - p^n) \cdot \mathbf{n} = 0$ on the obstacles, provided that the initial condition verifies it.

3.2. Adaptation to the Finite Element formulation

As already mentioned, the PDF method was initially developed in the finite-volume framework using a finite-difference MAC scheme. Here, we present a Galerkin Finite Element Method (FEM) version that has been implemented in a preexisting CEA application. As the location of the vector and scalar unknowns is completely different from the MAC scheme, new properties appear for the PDF projection equation (*cf.* Section 3.2). Concerning the $\mathbb{Q}_1$-$\mathbb{Q}_0$ formulation of the PDF equations (12) to (14), the first issue to address is the discretization of function χ (*cf.* Fig. 1). We consider that the discrete characteristic function is equal to 1 in the elements crossed by Γ and 0 elsewhere (*i.e.* the χ is decomposed in the $\mathbb{Q}_0$ FE basis). Thus, we can define χ_e and ξ_e such as:

$$\forall e \in [\![1, N_E]\!], \begin{cases} \chi_e = 1 & \text{if } \Omega_e \cap \Gamma \neq \emptyset \\ \chi_e = 0 & \text{else} \end{cases} \quad \text{and} \quad \xi_e = 1 + \frac{\chi_e}{\eta} \tag{15}$$

Then, the discrete version of the prediction equation (12) is obtained from equation (7), adding the discrete forcing term $\frac{1}{\delta t}\mathbf{M}_e \rho_e^n (\xi_e - 1)\left(\lambda_\Gamma^{n+1} - \lambda_e^*\right)$ to the RHS. Here $\lambda_\Gamma^{n+1} \in \mathbb{R}^{m_e}$ are the components of the discrete imposed velocity on Γ in the $\mathbb{Q}_1$ FE basis (*i.e.* at nodes) with $m_e = n_e d$ and $n_e \in \mathbb{N}$ the number of nodes belonging to the element K_e. At this stage, those components are considered as known values coming directly from the IB condition – what we call a direct assignment.

From equation (14), the discrete PDF projection equation is obtained the same way. Considering the discrete projection equation (8) and the discrete forcing term, we get:

$$\frac{1}{\delta t}\mathbf{M}_e \rho_e^n \left(\lambda_e^{n+1} - \lambda_e^*\right) - \frac{1}{\xi_e}\mathbf{B}_e \phi_e^{n+1} = \mathbf{0}_{\mathbb{R}^{m_e}} \tag{16}$$

and, left multiplying this equation by $B_e^T M_e^{-1}$ and considering the discrete mass balance, the projection equation, only depending on the pressure corrector, can be obtained (*cf.* Section 3.2). Finally, the full algorithm including PDF terms is summarized as follows:

$$\left(\frac{1}{\delta t}\widehat{\mathbf{M}}_e \rho_e^n + \mathbf{D}_e \mu_e^n + \mathbf{N}_e \rho_e^n\right)\lambda_e^* = -\mathbf{B}_e p_e^n + \frac{1}{\delta t}\mathbf{M}_e \rho_e^n \left(\lambda_e^n + g_e\right) + \frac{1}{\delta t}\mathbf{M}_e \rho_e^n (\xi_e - 1)\lambda_\Gamma^{n+1} \tag{17}$$

$$\mathbf{B}_e^T \widehat{\mathbf{M}}_e^{-1}\mathbf{B}_e \phi_e^{n+1} = -\frac{1}{\delta t}\mathbf{B}_e^T \rho_e^n \lambda_e^* \tag{18}$$

$$\lambda_e^{n+1} = \lambda_e^* + \delta t \widehat{\mathbf{M}}_e^{-1}\frac{1}{\rho_e^n}\mathbf{B}_e \phi_e^{n+1} \tag{19}$$

where $\widehat{\mathbf{M}}_e = \mathbf{M}_e \xi_e$ denotes a modified mass matrix. In the prediction equation, the PDF forcing is split into an implicit contribution (an addition to the mass matrix via the coefficient ξ_e) and an explicit source term. Using the matrix $\widehat{\mathbf{M}}_e$ in the time terms, the PDF fractional-step algorithm is very similar to the standard algorithm with the extra source term $\frac{1}{\delta t}\mathbf{M}_e \rho_e^n (\xi_e - 1)(\lambda_\Gamma^{n+1} - \lambda_e^n)$ that vanishes when $\chi_e = 0$. In a way, the proposed PDF method, when the forcing term is split, can be considered as hybrid (with respect to the categories presented in the introduction) because it involves both an addition of new terms (prediction equation) and a modification of operators (kind of Laplace operator in the projection equation).

One can also note that we consider a lumped mass matrix (*i.e.* the mass matrix is reduced to a diagonal matrix). Obviously, this involves a loss of information. However, in case of steady-state computations, the dynamic behavior is not of interest, neither the form of the mass matrix. And, in case of transient computations, R. L. T. Bevan et al. have shown that the impact of the lumping process on the results is moderate [52]. This approach motivated the idea to lump the implicit part of the contribution of the forcing term to the mass matrix. Indeed, the factor η^{-1}, present in the forcing term, can badly degrade the mass matrix conditioning as $\eta \ll 1$. Thus, lumping the contribution of the forcing term in conjunction with the use of a diagonal preconditioner greatly enhances the robustness of the method without degrading the results too much (*cf.* Section 4).

As mentioned in Section 3.1, the matrix related to the Laplace operator, in the projection equation (18), is not directly constructed from the weak form. Indeed, the discrete divergence matrix $\mathbf{B}_e^T$ is applied at a discrete level to the correction equation (16), considering the fact that the discrete mass equation stands as $\mathbf{B}_e^T \rho_e^n \lambda_e^{n+1} = \mathbf{0}_{\mathbb{R}^{m_e}}$.

Finally, let us note that, as all the components of the pressure-correction gradient term $\widehat{\mathbf{M}}_e^{-1}\mathbf{B}_e \phi_e^{n+1}$ are located at the same element nodes in this FEM scheme, the isolation mentioned in Section 3.1 occurs for all the space directions contrary to the previous finite difference scheme (isolation in the face-normal direction only) [43].

3.3. FE-based directional linear interpolation

In this work, we consider a directional (1D) polynomial interpolation to compute $\mathbf{u}_\Gamma^{n+1}$ (as in case of laminar flows), even if, in our target applications, it is not sufficient to properly take into account specific phenomena (turbulent flows for instance) and information must be added by the means of analytical or empirical laws. For the turbulent flows case, numerous turbulent wall laws have been developed and used to compute data on or in the vicinity of a boundary (see [53] for instance), but it will be treated in an upcoming paper.

Previously, we considered that the decomposition $\mathbf{u}_\Gamma$ in the $\mathbb{Q}_1$ FE basis (needed to compute the value of the explicit source added in the prediction equation) was directly given by the IB condition. In a way, this amounts to saying that the nodes belonging to the elements in which χ_e is non-zero are precisely located on the immersed obstacle (*i.e.* we have conforming boundaries like in a body-fitted case). Nonetheless, it is not actually true which means that, in order to enhance the fidelity of the method, the velocity imposed by the IB at those nodes has to be computed, or reconstructed, somehow. Otherwise, a direct assignment, using the obstacle boundary conditions onto the forced nodes, leads to a $\mathcal{L}^2$-norm space convergence of first order, as shown by theoretical or numerical studies [54,43]. This implies to have more geometrical information about the non-conforming boundary than its discrete characteristic function alone. To be more specific, a position at which the velocity imposed by the IB is actually known (*i.e.* a point of the immersed boundary) and the orientation of the IB at the given position are needed. A way to aggregate those two pieces of information is to consider the normal projection, on the IB, of the node at which $\mathbf{u}_\Gamma^{n+1}$ is computed. Indeed, the boundary velocity (or its normal derivative in case of slip condition) is known at this position and the normal vector can be deduced from the coordinates of the two points. The following

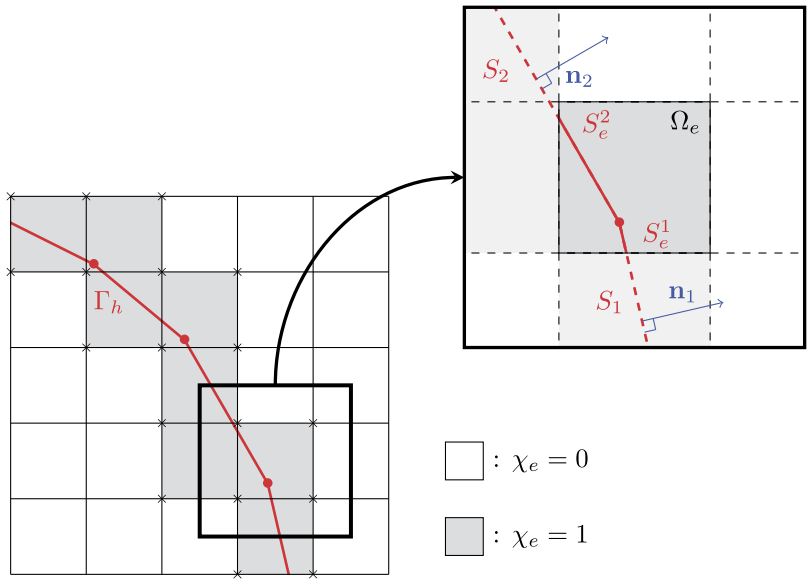


Fig. 1. Schematic representation of the computation grid and the collection of facets associated to the immersed boundary.

section described the method used to compute this projection point. In order to make the following explanations clearer, let us define some notations:

- $J_e = \left\{ j \in [\![1, N_N]\!] \,\middle|\, \mathbf{x}_j \in \Omega_e \right\}$: the set of node numbers included in element K_e,
- $E_j^0 = \{ e \in [\![1, N_E]\!] \,|\, j \in J_e \}$: the set of element numbers which share the node j,
- $E_j^\chi = \left\{ e \in E_j^0 \,|\, \chi_e = 1 \right\}$: the set of element crossed by Γ_h which share the node j,
- $J = \{ j \in [\![1, N_N]\!] \,|\, \exists e \in E_j^0 / \chi_e = 1 \}$: the set of node numbers for which a projection point need to be computed (points marked with a cross on Fig. 1).

3.3.1. *Geometrical data reconstruction*

From practical point of view, immersed boundaries are rather described by an equation when their geometry is simple – planes, circles, cylinders, *etc.* – and by a spline surface or a collection of facets – meshes coming from Computer Assisted Design (CAD) software for instance – when their geometry is complex. With respect to the aimed applications and CEA tools, only collection of plane facets – although exact equations are used on simple cases for purpose of verification and validation – are considered in the investigated approach. The projection of the nodes included in elements crossed by, at least one facets of the immersed boundary mesh, is achieved by using weighted data. The full process is detailed in this section. Although it does not depend on the problem dimension, it is illustrated with a 2D example for purpose of clarity.

We consider the collection of plane facets (*i.e.* polygons) $(S_i)_{i \in [\![1, N_S]\!]}$, representative of the actual immersed boundary Γ, with their respective normal vectors $(\mathbf{n}_i)_{i \in [\![1, N_S]\!]}$, and Ω_e the portion of Ω covered by the element K_e. The union of all facets can be seen as the approximation of the immersed boundary:

$$\Gamma_h = \bigcup_{i=1}^{N_S} S_i \tag{20}$$

Then, we can define S_e^i, the portion of facet S_i contained in Ω_e (*cf.* Fig. 1), as follows:

$$\forall e \in [\![1, N_E]\!], \forall i \in [\![1, N_S]\!], \quad S_e^i = S_i \cap \Omega_e \tag{21}$$

For each S_e^i, we can compute an area, denoted $\mathcal{A}_e^i$, and a barycenter, denoted $\mathbf{b}_e^i$ – note that S_e^i and S_i share the same normal vector $\mathbf{n}_i$ (*cf.* Fig. 1). Using area weighting, we can compute elemental data:

$$\forall e \in [\![1, N_E]\!] / \chi_e = 1, \begin{cases} \mathcal{A}_e = \displaystyle\sum_{i=1}^{N_S} \mathcal{A}_e^i \\ \mathbf{n}_e = \dfrac{1}{\mathcal{A}_e} \displaystyle\sum_{i=1}^{N_S} \mathcal{A}_e^i \mathbf{n}_i \\ \mathbf{b}_e = \dfrac{1}{\mathcal{A}_e} \displaystyle\sum_{i=1}^{N_S} \mathcal{A}_e^i \mathbf{b}_e^i \end{cases} \tag{22}$$

In a way, the triplet $(\mathcal{A}_e, \mathbf{n}_e, \mathbf{b}_e)$ represents a plane facet that is considered "equivalent" to the collection $(S_i)_{i \in [\![1, N_S]\!]}$ in the element K_e. Then, each node j of element e can be projected on this equivalent plane facet, which gives:

$$\forall e \in [\![1, N_E]\!] / \chi_e = 1, \forall j \in J_e, \quad \mathbf{p}_e^j = \mathbf{x}_j + l_e^j \mathbf{n}_e \tag{23}$$

with $\mathbf{x}_j$ the coordinates of node number j, $\mathbf{p}_e^j$ the coordinates of the projection of node number j on the plane facet associated to K_e and $l_e^j = (\mathbf{b}_e - \mathbf{x}_j) \cdot \mathbf{n}_e$ the oriented normal distance between the node j and the plane facet associated to K_e. Finally, $\mathbf{p}_e^j$ is assembled in a FE way to obtain:

$$\forall j \in J, \quad \alpha_j = \sum_{e \in E_j^\chi} \alpha_e^j, \quad \mathbf{x}_j^p = \frac{1}{\alpha_j} \sum_{e \in E_j^\chi} \alpha_e^j \mathbf{p}_e^j \tag{24}$$

with $\mathbf{x}_j^p$: the coordinates of the actual discrete projection of node j and the weight α_e^j. Several variants, depending on the value of α_e^j, have been developed: arithmetic mean, area weighting, invert distance weighting and area over distance weighting. But, only the area weighting is considered in the rest of the paper.

3.3.2. *Interpolation of the imposed velocity*

With those new pieces of information coming from the geometry of the immersed boundary Γ, directional interpolation methods can be used to compute the imposed velocity $\mathbf{u}_\Gamma^{n+1}$ (the time index exponent notation will be omitted in this section for purpose of readability, keeping in mind that the imposed velocity is interpolated at each time step). Given the coordinates $\mathbf{x}_j^p$ of the approximate projection of node j, we can reconstruct an outward normal vector:

$$\forall j \in J, \quad \mathbf{n}_j = \frac{\mathbf{x}_j - \mathbf{x}_j^p}{|\mathbf{x}_j - \mathbf{x}_j^p|} \tag{25}$$

Then, we find a point $\mathbf{x}_j^f$ in the prolongation of $\mathbf{n}_j$:

$$\mathbf{x}_j^f = \mathbf{x}_j + d_j^f \, c \, \mathbf{n}_e \tag{26}$$

with the distance d_j^f, the maximum distance between a node j and its neighbors, defined as follows:

$$\forall j \in J, \quad d_j^f = \max_{e \in E_j^0} \left(\max_{i \in J_e} \left| \mathbf{x}_j - \mathbf{x}_i \right| \right) \tag{27}$$

and $c \in]1, +\infty]$ a coefficient. The velocity at this point $\mathbf{x}_j^f$ is computed using the FE basis functions:

$$\mathbf{u}\left(\mathbf{x} = \mathbf{x}_j^f\right) \approx \sum_{k=1}^{N} \lambda_k \varphi_k \left(\mathbf{x} = \mathbf{x}_j^f\right) \tag{28}$$

with $\lambda_k \in \mathbb{R}^d$ the component k of the decomposition of the velocity in the FE basis and φ_k the $\mathbb{Q}_1$ basis function associated to the node k. However, as $\varphi_k \in \mathcal{D}\left({}_k\Omega\right)$ which means that the support of φ_k is compact and equal to ${}_k\Omega$ with:

$${}_k\Omega = \bigcup_{e \in E_k^0} \Omega_e \tag{29}$$

we can consider only the nodes belonging to the element f which contains the point $\mathbf{x}_j^f$. Then equation (28) becomes:

$$\mathbf{u}\left(\mathbf{x} = \mathbf{x}_j^f\right) \approx \mathbf{u}_j^f = \sum_{k \in J_f} \lambda_k \varphi_k \left(\mathbf{x} = \mathbf{x}_j^f\right) \tag{30}$$

Note that all nodes of J_f have to be "purely fluid" which means they have to belong only to elements not crossed by the boundary or, said otherwise, we must have:

$$\forall k \in J_f, \forall e \in E_k^0, \quad \chi_e = 0 \tag{31}$$

in order for $\mathbf{u}_j^f$ to be clearly defined. Indeed, if it is not the case, we obtain a circular definition for $\mathbf{u}_j$. To prevent this from happening in practice, we take $c = 1.1$ at first. Then, if the element f is not "purely fluid", we retry with $c = 2.1$.

Finally, if we denote $\mathbf{u}_j^p = \mathbf{u}\left(\mathbf{x} = \mathbf{x}_j^p\right)$ (given by the IB condition), we can interpolate the velocity at node j as follows:

$$\mathbf{u}(\mathbf{x} = \mathbf{x}_j) \approx \mathbf{u}_j = \mathbf{u}_j^p + \frac{\mathbf{u}_j^f - \mathbf{u}_j^p}{|\mathbf{x}_j^f - \mathbf{x}_j^p|} |\mathbf{x}_j - \mathbf{x}_j^p| \tag{32}$$

4. Numerical results

These numerical results have been obtained with a specific application based on the TRUST open-source platform [55,56], developed by the CEA, called GENEPI+, a component scale code used for the modeling of steam generator. It has mainly been chosen because it already features an immersed obstacle model which can easily be modified, *i.e.* turned into a PDF forcing term. Moreover, even if the numerical results provided in this Section are obtained for one-phase flows, it uses a relaxed homogeneous equilibrium model for two-phase flows. For more detailed information about the GENEPI+ code, the interested reader can refer to [57,9].

As a preamble, we detail our approximation of the aerodynamic force induced by the flow around an obstacle. At a fully continuous level, the definition of the aerodynamic force is the following:

$$\mathbf{t}_\Gamma := \int_\Gamma (\bar{\bar{\sigma}} - \bar{\bar{I}} p) \cdot \mathbf{n}_\Gamma = \int_{\Omega_s} \nabla \cdot (\bar{\bar{I}} p - \bar{\bar{\sigma}}) \tag{33}$$

If we denote:

$$\Omega_s^h = \bigcup_{e \in C} \Omega_e \quad \text{with} \quad C = \left\{ e \in [\![1, N_E]\!] / \Omega_e \cap \Omega_s \neq \emptyset \right\} \tag{34}$$

we can approximate the aerodynamic force at a time-discrete level:

$$\mathbf{t}_\Gamma = \int_{\Omega_s} \nabla \cdot (\bar{\bar{I}} p - \bar{\bar{\sigma}}) \approx \int_{\Omega_s^h} \nabla \cdot (\bar{\bar{I}} p^{n+1} - \bar{\bar{\sigma}}^{n+1}) = \mathbf{t}_\Gamma^h \tag{35}$$

Then, if we consider the semi-discrete weak formulation of the complete momentum balance equation on Ω_s^h, the approximate value of the aerodynamic force can be computed as follows:

$$\mathbf{t}_\Gamma^h = \int_{\Omega_s^h} \mathbf{f}^{n+1} - \int_{\Omega_s^h} \left[\rho^n \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\delta t} + \nabla \cdot (\rho^n \mathbf{u}^n \otimes \mathbf{u}^{n+1}) \right] \tag{36}$$

As a first approximation, the integral of the total derivative is assumed negligible – *i.e.* steady state, uniformity and symmetry assumptions which can lead to incorrect estimations (*cf.* Section 4.3). Thus we have:

$$\mathbf{t}_\Gamma^h \approx \int_{\Omega_s^h} \mathbf{f}^{n+1} = \int_{\Omega} \mathbf{f}^{n+1} \tag{37}$$

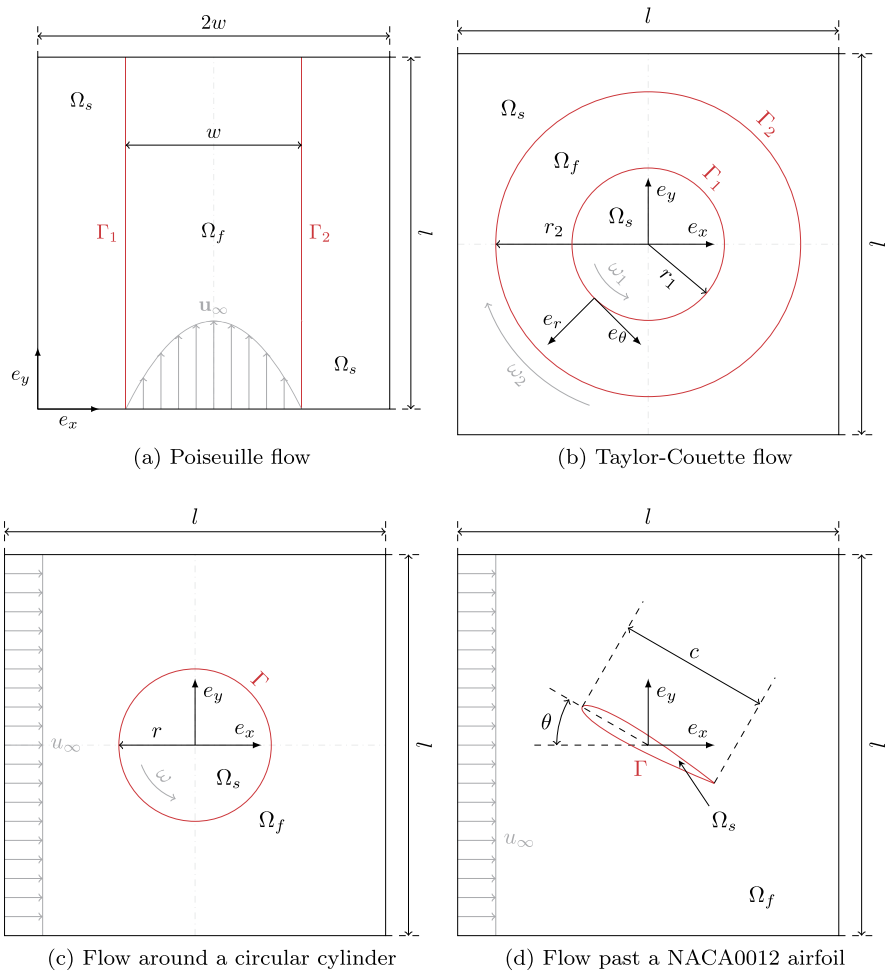
4.1. Laminar 2D Poiseuille flow (channel aligned with the mesh)

4.1.1. Description

Poiseuille flow refers to a viscous fluid flow between two parallel plane plates, what we call channel, or within a cylindrical tube. In two dimension, which is our case of interest for verification purposes, both cases are equivalent. Thus we will only consider the flow between two plates. Fig. 2a gives our test case configuration where the Reynolds number $\mathfrak{R}$ is equal to 1, Ω_f (resp. Ω_s) represents the physical (resp. non-physical) domain ($\Omega = \Omega_f \cup \Omega_s$ is the computational or fictitious domain), Γ_1 and Γ_2 are the two immersed boundaries at which a no-slip condition is considered, $l = 4m$ is the length of the channel, $w = 1m$ is the width of the channel and $\mathbf{u}_\infty$ represents the inlet velocity field (given by the analytical solutions of the Poiseuille flow problem detailed hereafter) with $U_\infty = \max\left(|\mathbf{u}_\infty|\right) = 1$ m.s^{-1}. The distance between the plates is assumed widely smaller than the dimensions of the plates (*i.e.* $w \ll l$). Therefore, the lubrication theory is used to obtain analytical solutions.

4.1.2. Studies

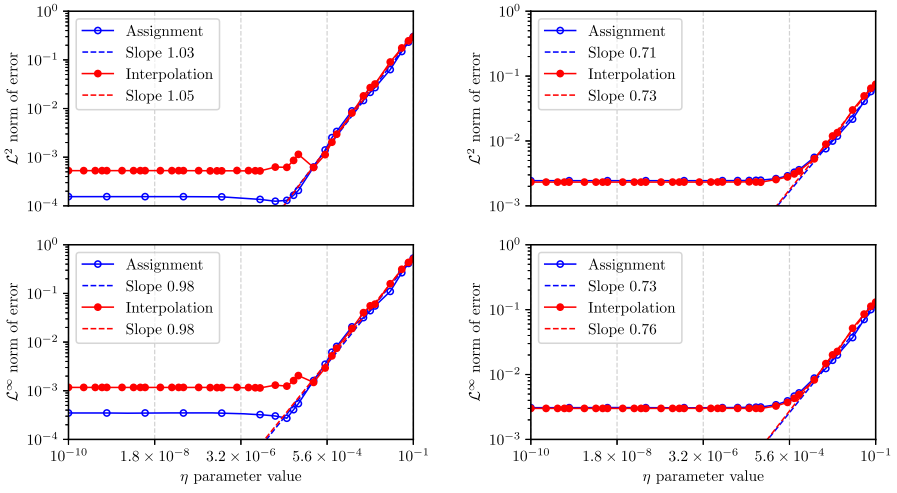
The behavior of our method with respect to the penalty parameter η has been studied in the case of a 2D Poiseuille flow. The channel is aligned with the grid and the immersed boundaries are conforming to element interfaces in order to minimize the spatial error and focus on the penalty error. Fig. 3a shows the $\mathcal{L}^2$ and $\mathcal{L}^\infty$ relative norms of the error of the component of the velocity oriented along the channel axis obtained from several values of the penalty parameter. The blue curve marked with empty circles corresponds to results obtained with the variant of the method in which the value of the imposed velocity is directly assigned to the immersed Dirichlet boundary condition value. The red curve marked with disks corresponds to the results obtained with the directional velocity interpolation technique described in Section 3.3.2. In both cases, the numerical order of convergence is close to 1, which is consistent with the conclusions of P. Angot et al. in [16,15]. However, it can be noted that the converged value of the relative error, in both norms, is higher when using the interpolation technique. It could be explained by the fact that the interpolation process adds a spatial error which is absent when using the direct assignment – in that case, the exact solution value is directly assigned to the imposed velocity because the immersed boundaries are conforming. The evolution of the pressure gradient with respect to the penalty parameter η tends to be similar and so it is not shown in this paper. As pressure gradient is not interpolated, this tends to confirm the assumption about an added space error due to the interpolation process. Concerning pressure, in



(a) Poiseuille flow (b) Taylor-Couette flow

(c) Flow around a circular cylinder (d) Flow past a NACA0012 airfoil

Fig. 2. Schematic representation of the computational domain for the different laminar validation cases.

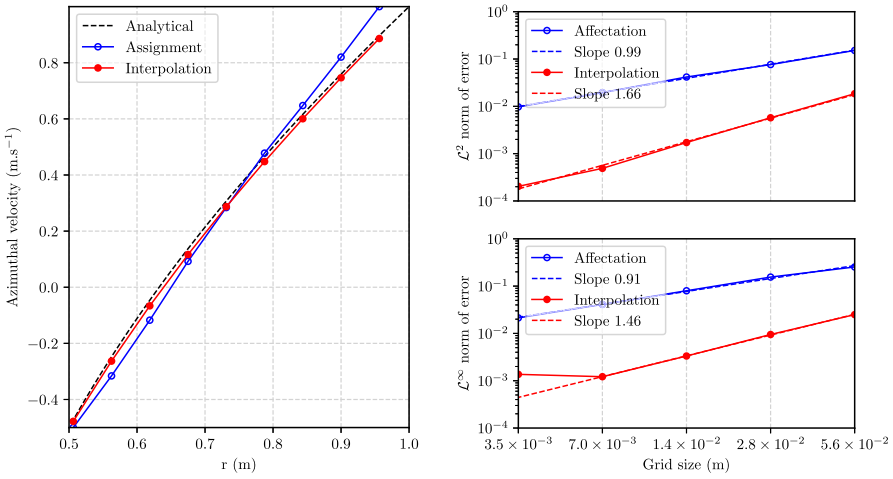


(a) Velocity (b) Pressure

Fig. 3. Evolution of relative norms ($\mathcal{L}^2$ and $\mathcal{L}^\infty$) of the error related to the velocity along e_y and related to the pressure with respect to the value of the penalty parameter η.

Fig. 3b, results are similar for both techniques and an order of convergence of about 3/4 is found – a value which is, once again, consistent with the conclusions of P. Angot.

All the norms have been computed on a part of the fluid domain far from boundary to avoid boundary effects (*i.e.* for $0.5 < x < 1.5$ and $1 < y < 3$).



(a) Azimuthal velocity profile

(b) Azimuthal velocity convergence

Fig. 4. Evolution of azimuthal velocity along the radius and evolution of the relative $\mathcal{L}^2$ and $\mathcal{L}^\infty$ norms of error of the azimuthal velocity with respect to the grid step.

4.2. Laminar Taylor-Couette flow

4.2.1. Description

Taylor-Couette flow refers to the flow between two infinitely long concentric circular cylinders which are rotating at different angular velocities. Using previous notations, Fig. 2b gives a schematic view of the case configuration where $Re = 1$, $r_1 = 0.5m$ (resp. $r_2 = 1m$) is the radius of the inner (resp. outer) cylinder, $\omega_1 = 1rad.s^{-1}$ (resp. $\omega_2 = -1rad.s^{-1}$) is the angular velocity of the inner (resp. outer) cylinder, Γ_1 (resp. Γ_2) is the immersed boundaries corresponding to the inner (resp. outer) cylinders (they correspond to a no-slip condition, which means a Dirichlet BC with the Dirichlet fluid velocities given by the cylinders velocities), $l = 2r_2 + \frac{1}{4}$ is the side of the square domain and (e_r, e_θ) is the polar frame. The stability of this flow is ensured by a condition on the Taylor number, denoted Ta:

$$Ta := \frac{{\omega_1}^2 r_m (r_2 - r_1)^3}{\nu^2} < Ta_c \quad \text{with} \quad r_m = \frac{r_1 + r_2}{2} \tag{38}$$

where $Ta_c \approx 1.712$ is the critical Taylor number computed by linear analysis [58]. The numerical application gives, in our case, $Ta = 1.5$, so the criterion is respected. This means that, using lubrication theory once again, we are able to compute a steady-state solution in which the fluid velocity is purely azimuthal.

4.2.2. Studies

The mentioned laminar Taylor-Couette case was used to carry out a mesh convergence study. The obtained results are good agreement with the analytical solution and the spatial convergence is numerically observed (*cf.* Fig. 4). Even if the orders of convergence (about 1.7 in $\mathcal{L}^2$ norm and 1.1 in $\mathcal{L}^\infty$ norm), when using interpolation, are not as close to 2 as the one computed for the tilted Poiseuille flow (results are not shown presented in this paper for purpose of concision, this technique greatly enhances the results (a factor 10^{-1} applied on the $\mathcal{L}^2$ relative norm of error and a better agreement with the analytical solution observed on the profile view).

4.3. Laminar flow around a circular cylinder

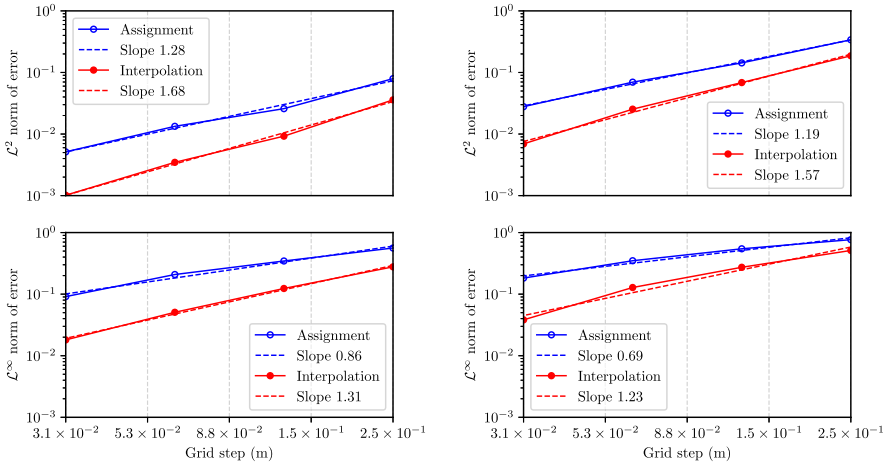
4.3.1. Description

The flow around a circular cylinder is a widely studied problem in the field of fluid dynamics. Using previous notations, Fig. 2c gives the test case configuration where Γ is the immersed boundaries corresponding to the surface of the cylinder, $r = 0.5m$ is the cylinder radius, $l = 120r$ is the side of the square domain. It is large to avoid boundary effects, $\mathbf{u}_\infty$ represents the uniform inlet velocity field and $\omega \in \{0, 2\}$ is the angular velocity of the cylinder. There is no analytical solution, but many experiments and simulations give macroscopic indicators, such as the drag and lift coefficients, as comparison elements in the static case (*i.e.* $\omega = 0$). Some experimental data are also available for the rotating case.

4.3.2. Studies

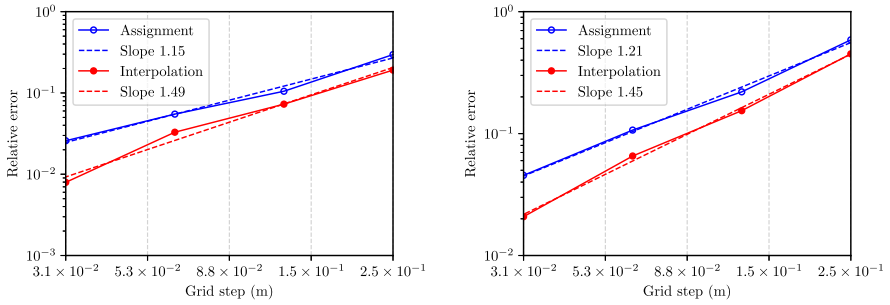
First, a mesh convergence study was carried out using the whole solutions for laminar steady cases ($Re = 20$) and the drag coefficients for laminar unsteady cases ($Re = 100$) – we use an adaptative time step which depends on the grid step, so the solutions of unsteady cases desynchronize when the mesh changes. As there is no analytical solution for the Navier-Stokes equations in the case of a flow around a circular cylinder, the simulation results obtained using the finest grid are considered as a reference. Moreover, as we are only concerned with convergence aspects in this study, we can consider a smaller domain ($10d \times 10d$) without loss of generality. When looking at the mesh convergence curves, in $\mathcal{L}^2$ and $\mathcal{L}^\infty$ norm, for the static cylinder at $Re = 20$ (*cf.* Figs. 5a and 5b), the conclusion is the same as the one formulated for the Poiseuille and Taylor-Couette flows: the linear interpolation of the velocity in the vicinity of the obstacles reduces the error between the computed and reference solutions while increasing the rate of convergence (conclusion is also the same for the rotating cylinder at $Re = 20$, results are not shown in this paper for purpose of concision). However, the order of convergence is slightly lower than two in $\mathcal{L}^2$ norm when using the linear interpolation (in [1.4, 1.7]) and slightly larger than 1 when using only direct assignment (in [1.2, 1.3]). This may be linked to the lack of analytical solution. This difference in convergence order is also noticed in the $\mathcal{L}^\infty$ norm: in [1.2, 1.3] when using the linear interpolation and in [0.7, 0.9] when using the direct assignment of the velocity. This analysis can be extended to unsteady cases at $Re = 100$ considering the evolution of the relative error of the drag coefficient with respect to the grid step (*cf.* Figs. 5c and 5d).

Second, a validation study has been carried out. The streamlines computed with our approach have been compared to the numerical solutions, presented in [43], in both steady and unsteady cases (*cf.* Fig. 6 for examples). Globally, the shapes are in good agreement with the literature, even if the balancing tensor diffusivity (BTD) scheme – used to deal with the advective term – induces an extra numerical diffusion [59]. Global quantities such as drag and lift coefficients (respectively C_d and C_l with, in unsteady cases, their mean values denoted with a bar and their fluctuations denoted with an apostrophe), the angle between



(a) Streamwise velocity ($w = 0$ rad.s^{-1}s, $Re = 20$)

(b) Spanwise velocity ($w = 2$ rad.s^{-1}, $Re = 20$)



(c) Mean drag coefficient ($w = 0$ rad.s^{-1}, $Re = 100$)

(d) Mean drag coefficient ($w = 2$ rad.s^{-1}, $Re = 100$)

Fig. 5. Mesh convergence (relative error, $\mathcal{L}^2$ and $\mathcal{L}^\infty$) of the velocity components and drag coefficients for the flow around a circular cylinder.

Table 1
Aerodynamic coefficients computed for the steady laminar flow around a static circular cylinder ($Re = 20$) using both direct assignment (label "A") and linear interpolation techniques (label "B") where $2r/h$ represents the number of elements within the diameter of the cylinder.

		$2r/h$		References						
		10	20	[43]	[60]	[61]	[62]	[63]	[64]	[65]
C_d	A	2.245	2.154	2.059	2.03	2.02	2.06	2.06	2.00	2.09
	B	2.142	2.075	2.054						
L_w	A	1.297	1.166	0.925	0.92	0.90	0.94	0.93	0.91	–
	B	0.749	0.901	0.900						

Table 2
Aerodynamic coefficients computed for the steady laminar flow around a rotating circular cylinder ($Re = 20$) using both direct assignment (label "A") and linear interpolation techniques (label "B") where $2r/h$ represents the number of elements within the diameter of the cylinder.

		$2r/h$		References				
		10	20	[43]	[66]	[67]	[68]	[69]
C_d	A	2.057	1.984	1.8608	1.888	1.85	1.925	2.000
	B	2.002	1.913	1.8679				
C_l	A	3.181	3.032	2.9419	2.629	2.75	2.617	2.740
	B	2.970	2.868	2.7745				
$\alpha(°)$	A	57.11	56.79	57.68	54.31	56	53.66	53.87
	B	56.02	56.29	56.05				

Table 3
Aerodynamic coefficients related to the unsteady laminar flow around a static circular cylinder ($Re = 100$) where $2r/h$ represents the number of elements within the diameter of the cylinder.

	$2r/h$		References							
	20	40	[43]	[67]	[63]	[61]	[70]	[71]	[72]	[73]
$\overline{C_d}$	1.338	1.319	1.347	1.337	1.340	1.340	1.350	1.317	1.376	–
C_d'	0.010	0.012	0.009	0.009	0.009	0.011	0.012	0.009	0.010	–
C_l'	0.276	0.285	0.326	0.326	0.333	0.315	0.303	0.349	0.339	0.227
St	0.145	0.161	0.165	0.165	0.166	0.164	0.167	0.170	0.170	0.164

the aerodynamic force and the horizontal axis (α) and the Strouhal number (St) have been computed and compiled in Tables 1, 2, 3 and 4. Those values are in good agreement with the literature (physical and numerical experiments), even if some minor discrepancies are noticed (lift-related coefficients and Strouhal number) in unsteady cases ($Re = 100$). Those discrepancies might be explained by the lack of the particular derivative of the velocity in the approximation of the aerodynamic force (*cf.* preamble of Section 4) – this assumption will be tested in upcoming works.

4.4. Laminar flow past a NACA0012 airfoil

4.4.1. Description

The flow past a NACA airfoil is a typical issue in aerodynamics. Usually, due to the high velocity of flying objects such as planes, it involves turbulence modeling. However, some laminar cases have been studied and simulated by R.C. SWANSON et al. [75]. This provides elements of comparison and test configurations, detailed in Fig. 2d where θ is the angle of attack, c is the chord of the NACA airfoil, $l = 10c$ is the

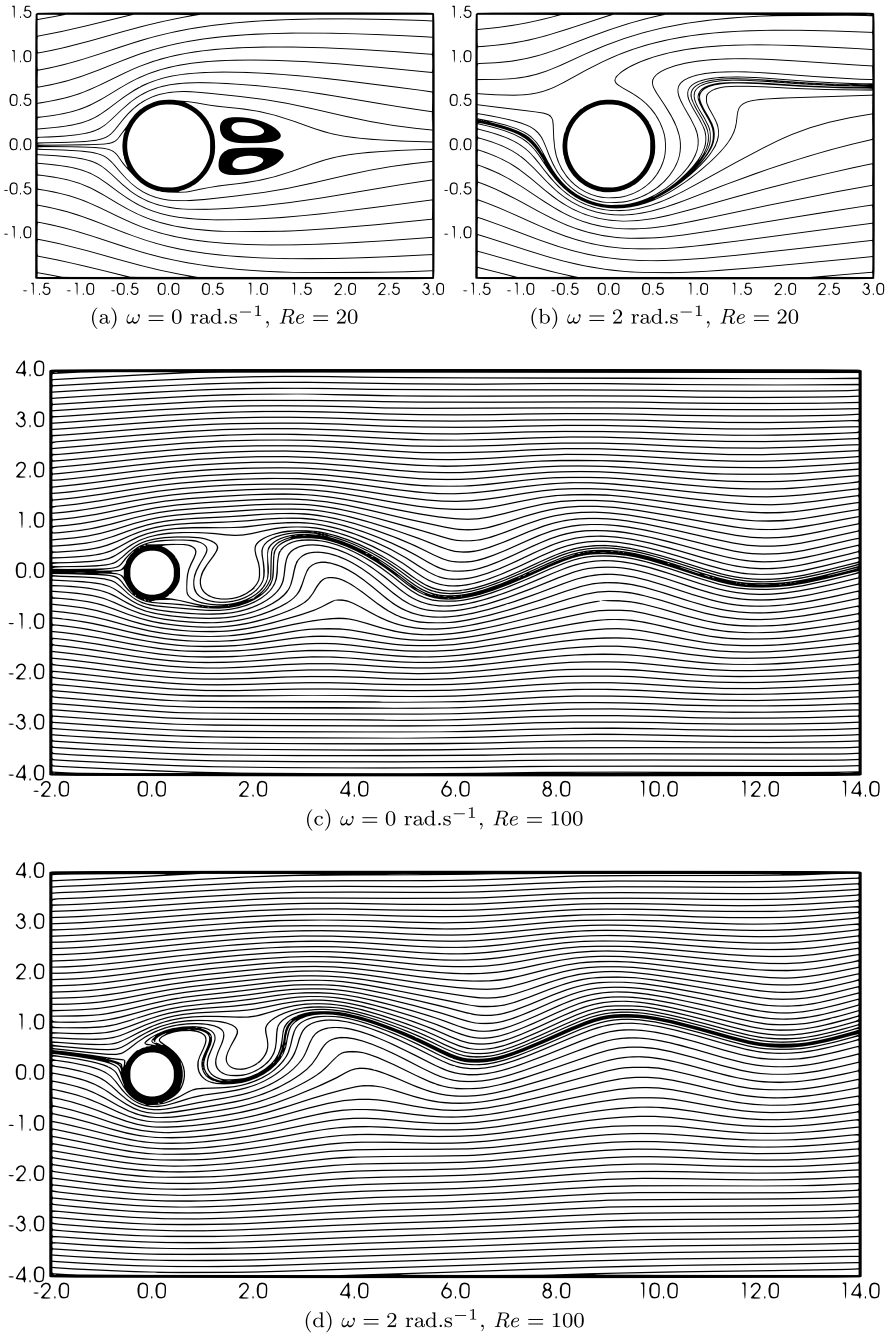


Fig. 6. Streamlines computed for various configurations of the flow around a circular cylinder.

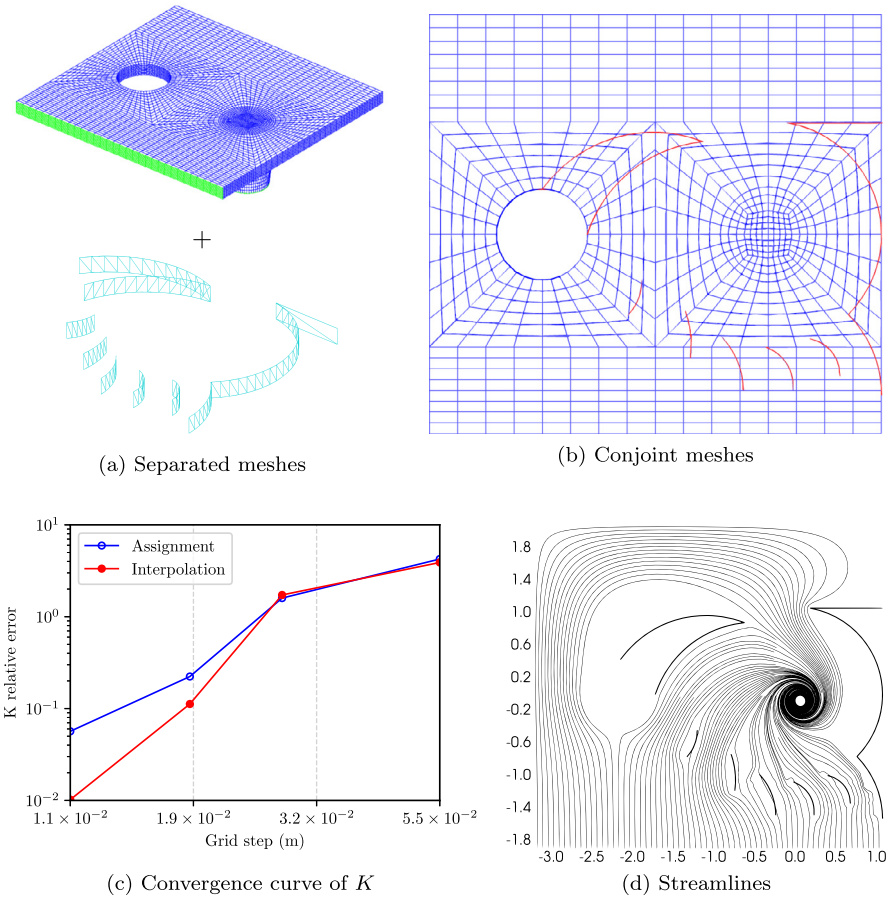
Table 4
Aerodynamic coefficients related to the unsteady laminar flow around a rotating circular cylinder ($Re = 100$) where $2r/h$ represents the number of elements within the diameter of the cylinder.

	$2r/h$		References			
	20	40	[43]	[67]	[66]	[74]
$\overline{C_d}$	1.143	1.097	1.12	1.1080	1.1890	1.0979
$C_d{}'$	0.098	0.091	0.11	0.0986	0.1195	0.0988
$\overline{C_l}$	2.634	2.573	2.51	2.5040	2.4050	2.4833
$C_l{}'$	0.285	0.291	0.37	0.3616	0.4427	0.3603
St	0.151	0.165	0.165	0.1658	0.1732	0.1650

side of the square domain and $\mathbf{u}_\infty$ represents the uniform inlet velocity field with $U_\infty = \max\left(|\mathbf{u}_\infty|\right) = 0.5$ m.s^{-1}. We tested five configurations: $\theta \in \{0°, 1°, 2°, 3°\}$ at $\mathfrak{R} = 5000$ and $\theta = 10°$ at $Re = 500$. All those configurations are supposed to reach a steady state, as stated in [75].

4.4.2. Studies

For the case with a 0° angle of attack, shapes and recirculation length are really similar to the one presented in [75] (about 0.17 m in both cases). Concerning aerodynamic coefficients, we gathered the steady regime results obtained for angles of attack of 0 and 1° in Table 5. Taking into account that an angle difference of 1° can not be exactly reproduced using our immersed boundary geometrical model, we conclude that the aerodynamic coefficients are in a good agreement. However, for cases involving an angle of attack superior to 1°, our method was not able to recover steady state solutions so only the mean values of the aerodynamic coefficients are shown in Table 5. This can be induced by several phenomena such as: residuals being



(a) Separated meshes (b) Conjoint meshes

(c) Convergence curve of K (d) Streamlines

Fig. 7. Mesh configuration of the case involving a device representative of the flow limiter as well as the evolution of the head loss coefficient K with respect to the equivalent grid step h and streamlines computed with a grid composed of 164160 cells.

Table 5
Aerodynamic coefficients related to the laminar flow past a NACA airfoil. In total, the domain mesh counts 360,000 elements with approximatively 115 within the chord length of the airfoil.

θ	C_d		C_l	
	Interpolation	[75]	Interpolation	[75]
0	0.0563	0.0555	0.0094	0.0000
1	0.0571	0.0559	0.0092	0.0184

too large because of high convergence criteria (as suggested by R.C. SWANSON), boundary effects (the size of the used domain is smaller than the one used for the circular cylinder cases), grid being too coarse and not adapted to the problem (*i.e.* staircase description of the obstacle which could decrease the transition between steady / unsteady regimes in terms of Reynolds number) and the use of super-convergence algorithm in the work of R.C. SWANSON.

4.5. Industrial case involving the flow limiter

4.5.1. Description

This case is representative of the flow into a hydraulic diode. As we can see in Figs. 7a and 7b, we have a volume mesh (which represents the fluid located in the downcomer of a PWR including the inlet and outlet vessel) and a surface mesh (which comes from a CAD software and represents the shape of the flow limiter). The combination of the two, gives a volume mesh with embedded data about the immersed boundary (characteristic function, normal vector, *etc.*).

The case in itself corresponds to a LOCA induced by a break on the cold inlet vessel. Thus, as the primary circuit is pressurized, the flow comes from the bottom of the core, with a flowrate of 5.2 kg.s^{-1} (this estimation is coming from system scale computations), to the break (*i.e.* the inlet vessel). The outflow pressure is fixed at an *ad hoc* value of 50 bar. A Schlichting's scalar turbulence model is also used with a characteristic length of turbulence $L_T = 0.3$ m. For a more detailed description of the case, the interested reader can refer to [8].

4.5.2. Studies

Using our method, we were able to successfully run 3D flow limiter simulations. As shown on Fig. 7d, the expected behavior (*i.e.* the creation of a vortex to dissipate kinetic energy) is respected. However, we can note the appearance of a chessboard pressure pattern for this rather coarse mesh. Practically, this phenomena intensity is decreasing with the space step. Then, in order to soften this phenomenon, we should optimize the mesh configuration by increasing the density of the cells at the vicinity of the immersed boundaries while reducing it in the cold leg (in order to preserve the same number of cells and so a similar computation time), *cf.* Fig. 7b.

We also used the results of those simulations to carry out a mesh convergence study. We focused on the evolution of the head loss coefficient K over the different grids. As the grids are not uniform (*cf.* Fig. 7b), we define an average space step $h = N^{-\frac{1}{3}}$ with N the number of cells of the considered grid. Alike previous mesh convergence studies with no analytical solution, the solution of the finest grid is considered as reference. The obtained results for the head loss coefficients are gathered in Table 6. The values justify the preliminary study results [8] which use an approximated soft penalty added source term which is only present in the prediction equation. Moreover, those results show once again that the interpolation technique tends to enhance the spatial convergence (*cf.* Fig. 7c).

Table 6
Values of the head loss coefficient K computed with the PDF method using both direct assignment and linear interpolation techniques for different grids.

		K	
N	h (m)	Direct assignment	Interpolation
6,080	5.48×10^{-2}	28.3	26.4
48,640	2.74×10^{-2}	14	14.7
164,160	1.83×10^{-2}	6.6	6.0
794,880	1.08×10^{-2}	5.7	5.5
3,594,240	6.53×10^{-3}	5.5	5.4

5. Conclusion and perspectives

In this document, the theoretical aspects of the Penalized Direct Forcing method (an IBM whose characteristics inherit from both penalty and direct forcing methods) are detailed: it involves solving the dilatable Navier-Stokes equations using a fractional-step algorithm and an additional source term representative of the modeled obstacles (some kind of backmoving force). It was initially developed and tested in the context of a space-time discretization based on a finite difference scheme. Here, we specify and test the method using a Galerkin finite element discretization with a lumping of the mass matrix. Moreover, an enhancement of the method is also proposed. It consists in a directional interpolation of the velocity in the vicinity of the obstacles. To be able to interpolate, new data about the immersed boundary are needed and the method to obtain them is also described here. A list of validation test cases is presented, as well as numerical results. The steady laminar Poiseuille and Taylor-Couette flows, for which analytical solutions are known, allowed us to carry out numerical convergence studies, with respect to the penalty parameter and the grid step. It is worth noting that the new linear interpolation technique reduces the difference between the computation results and the analytical solution while increasing the spatial order of convergence (almost reaching two for the velocity in $\mathcal{L}^2$ norm). Another validation case presented in this document is the laminar flow around a circular cylinder. It is declined in four configurations: steady regime with static cylinder, steady regime with rotating cylinder, unsteady regime with static cylinder and unsteady regime with rotating cylinder. Even if no analytical solution is available for the flow around a cylinder, a mesh convergence study is presented (the numerical results computed with the finest grid are considered as a reference) and shows, once again, that the interpolation technique increases the spatial order of the method. The results obtained with our approach are also compared to experimental data and other simulations via quantities such as aerodynamic coefficients and the Strouhal number on the circular cylinder test case and a laminar flow past a NACA0012 airfoil test case. Globally, the values obtained are in good agreement with the literature which provides valuable validation data. An industrial study, involving a steady turbulent flow past a complex geometry (representative of a flow limiter) is also presented. Overall, the results are in good agreement with the previous preliminary studies [8]. Moreover, simulations involving finer grids have been carried out, showing that the present method is more robust than the one used in [8].

In the near future, two developments are considered. The first one consists in interpolating the normal component of the pressure gradient in the vicinity of the immersed boundaries. Indeed, the velocity is interpolated using the presented methodology but not the pressure gradient (nor pressure corrector). Yet, the pressure gradient appears in the Navier-Stokes prediction equation. An idea to deal with this issue could be to interpolate the normal pressure gradient or corrector (as we consider $\nabla p^{n+1} \cdot \mathbf{n} = 0$ as immersed boundary condition for the pressure) using the same methodology as the one used of the velocity. The second one involves turbulence modeling: the idea is to extend the interpolation process to turbulent wall laws (*i.e.* power wall law [53]). Thus, we will be able to carry out turbulent simulations of the flow limiter involving well-established turbulent models as the wall laws, RANS and LES models. The modeling of other passive safety systems, such as the advanced accumulator [5], is also considered to demonstrate the interest of using our method to carry out design and shape optimization studies.

CRediT authorship contribution statement

Georis Billo: Conceptualization, Formal analysis, Methodology, Software, Writing – original draft. **Michel Belliard:** Conceptualization, Formal analysis, Methodology, Software, Writing – review & editing. **Pierre Sagaut:** Resources, Supervision, Writing – review & editing.

Acknowledgements

The authors would like to thank the TRUST platform and GENEPI+ development teams, especially Adrien Bruneton and Pierre Ledac, for their contributions and technical support.

Funding: This work was supported by the French Atomic Energy Commission (CEA), Centre de Cadarache, F-13108 Saint Paul-Lez-Durance, France.

References

[1] V. Saul'ev, On the solution of some boundary value problems on high performance computers by fictitious domain method, Sib. Math. J. 4 (4) (1963) 912–925 (in Russian).
[2] S. Vincent, A. Sarthou, J.-P. Caltagirone, F. Sonilhac, P. Février, C. Mignot, G. Pianet, Augmented Lagrangian and penalty methods for the simulation of two-phase flows interacting with moving solids. Application to hydroplaning flows interacting with real tire tread patterns, J. Comput. Phys. 230 (4) (2011) 956–983.
[3] R. Glowinski, T.-W. Pan, T. Hesla, D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiph. Flow 25 (1999) 755–794.
[4] M. Belliard, C. Introïni, Trio_u thermal-hydraulic simulations of flows induced by a stirrer and bubbling in a molten glass bath, in: Joint EUROMECH/ERCOFTAC Colloquium 549, Immersed Boundary Methods: Current Status and Future Research Directions, Leiden, The Netherlands, June 17-19, 2015.
[5] T. Shiraishi, Design of the advanced accumulator for the pressurized water reactor, Nucl. Eng. Des. 241 (9) (2011) 3910–3924.
[6] G. Gautier, Dispositif limiteur de débit inverse de fluide, patent n° 88 12665, 1988.
[7] E. Stratta, M. Belliard, Thermal-hydraulic study of passive safety systems based on the hydraulic diode principle for the management of large-break loss of coolant accidents, in: 17th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-17), Qujiang Int. Conference Center, Xi'an, China, September 3-8, 2017.
[8] M. Belliard, Numerical modeling of an in-vessel flow limiter using an immersed boundary approach, Nucl. Eng. Des. 330 (2018) 437–449.
[9] M. Grandotto, P. Obry, Calculs des écoulements diphasiques dans les échangeurs par une méthode aux éléments finis, Rev. Eur. Éléments Finis 5 (1) (1996) 53–74 (in French).
[10] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier-Stokes equations, J. Comput. Phys. 168 (2001) 464–499.
[11] H. Schlichting, Boundary Layer Theory, Mac Graw Hill, New York, USA, 1968.
[12] K. Khadra, P. Angot, S. Parneix, J. Caltagirone, Fictitious domain approach for numerical modelling of Navier-Stokes equations, Int. J. Numer. Methods Fluids 34 (8) (2000) 651–684.
[13] R. Glowinski, T.-W. Pan, J. Periaux, A fictitious domain method for Dirichlet problem and applications, Comput. Methods Appl. Mech. Eng. 111 (1994) 283–303.
[14] V. Girault, R. Glowinski, Error analysis of a fictitious domain method applied to a Dirichlet problem, Jpn. J. Ind. Appl. Math. 12 (1995) 487–514.
[15] P. Angot, C. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Numer. Math. 81 (1999) 497–520.
[16] P. Angot, Analysis of singular perturbations on the Brinkman problem for fictitious domain of viscous flows, Math. Methods Appl. Sci. 22 (1999) 1395–1412.
[17] A. Sarthou, S. Vincent, P. Angot, J. Caltagirone, The sub mesh penalty method, in: Finite Volumes for Complex Applications V, 2008, p. 633.
[18] B. Maury, A fat boundary method for the Poisson problem in a domain with holes, J. Sci. Comput. 16 (2001) 319–339.
[19] M. Ismail, The fat boundary method for the numerical resolution of elliptic problems. Application to 3d fluid flows, Ph.D. thesis, Université Pierre et Marie Curie - Paris VI, France, 2004.
[20] I. Ramière, P. Angot, M. Belliard, A fictitious domain approach with spread interface for elliptic problems with general boundary conditions, Comput. Methods Appl. Mech. Eng. 196 (2007) 766–781.
[21] M. Belliard, I. Ramière, Fictitious domain methods for two-phase flow energy balance computations in nuclear components, Int. J. Numer. Methods Fluids 68 (8) (2012) 939–957.

[22] C. Peskin, Flow patterns around heart valves: a digital computer method for solving the equations of motion, Ph.D. thesis, Albert Einstein College of Medicine, 1972.
[23] C. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[24] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
[25] X. Li, J. Lowengrub, A. Rätz, A. Voigt, Solving PDEs in complex geometries: a diffuse domain approach, Commun. Math. Sci. 7 (1) (2009) 81–107.
[26] R. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, J. Numer. Anal. 31 (1994) 1019–1044.
[27] L. Lee, R. LeVeque, An immersed interface method for incompressible Navier-Stokes equations, J. Sci. Comput. 25 (2003) 832–856.
[28] Z. Li, An overview of the immersed interface method and its applications, Taiwan. J. Math. 7 (2003) 1–49.
[29] Y. Tseng, J. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2003) 593–623.
[30] A. Baeza, P. Mulet, D. Zorío, High order boundary extrapolation technique for finite difference methods on complex domains with Cartesian meshes, J. Sci. Comput. 66 (2016) 761–791.
[31] C. Chi, A. Abdelsamie, D. Thévenin, A directional ghost-cell immersed boundary method for incompressible flows, J. Comput. Phys. 404 (2020) 593–623.
[32] G. Yang, D. Causon, D. Ingram, R. Saunders, P. Batten, A Cartesian cut cell method for compressible flows, Aeronaut. J. 101 (1002) (1997) 47–56.
[33] D. Devendran, D. Graves, H. Johansen, T. Ligocki, A fourth-order Cartesian grid embedded boundary method for Poisson's equation, Commun. Appl. Math. Comput. Sci. 12 (2017) 51–79.
[34] H. Johansen, P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, J. Comput. Phys. 147 (1998) 60–85.
[35] P. Schwartz, M. Barad, P. Collela, T. Ligocki, A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions, J. Comput. Phys. 211 (2006) 531–550.
[36] I. Ramière, P. Angot, M. Belliard, A general fictitious domain method with immersed jumps and non-conforming structured mesh, J. Comput. Phys. 225 (2) (2007) 1347–1387.
[37] A. Parvizian, A. Düster, E. Rank, Finite cell method: h- and p-extension for embedded domain problems in solid mechanics, Comput. Mech. 41 (1) (2007) 121–133.
[38] G. Wagner, N. Moës, W. Liu, T. Belytschko, The extended finite element method for rigid particles in Stokes flow, Int. J. Numer. Methods Eng. 51 (3) (2001) 293–313.
[39] J. Mohd-Yusof, Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries, Annu. Res. Briefs (1997) 317–327.
[40] E. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (2000) 35–60.
[41] M. Belliard, C. Fournier, Penalized direct forcing and projection schemes for Navier-Stokes, C. R. Acad. Sci., Sér. 1 Math. 348 (2010) 1133–1136.
[42] A. Sarthou, Methodes de domaines fictifs d'ordres élevés pour les équations elliptiques et de Navier-Stokes. Application au couplage fluide-structure, Ph.D. thesis, Université de Bordeaux I, 2009 (in French).
[43] C. Introïni, M. Belliard, C. Fournier, A second order penalized direct forcing for hybrid Cartesian/immersed boundary flow simulations, Comput. Fluids 90 (2014) 21–41.
[44] L. Manueco, P. Weiss, S. Deck, On the estimation of unsteady aerodynamic forces and wall spectral content with immersed boundary conditions, Comput. Fluids 201 (2020) 104471.
[45] T. Ikeno, T. Kajishima, Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations, J. Comput. Phys. 226 (2007) 1485–1508.
[46] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, Appl. Mech. Rev. 56 (3) (2003) 331–347.
[47] A. Bharadwaj S, S. Ghosh, Data reconstruction at surface in immersed-boundary methods, Comput. Fluids 196 (2020) 104236.
[48] S. Clerc, Numerical simulation of the homogeneous equilibrium model for two-phase flows, J. Comput. Phys. 161 (2000) 354–375.
[49] V. Girault, P. Raviart, Finite Element Methods for Navier-Stokes Equations, Springer-Verlag, 1986.
[50] F. Domenichini, On the consistency of the direct forcing method in the fractional step solution of the Navier-Stokes equations, J. Comput. Phys. 227 (2008) 6372–6384.
[51] R. Guy, D. Hertenstgine, On the accuracy of direct forcing immersed boundary methods with projection methods, J. Comput. Phys. 229 (2010) 2479–2496.
[52] R.L.T. Bevan, P. Nithiarasu, Accelerating incompressible flow calculations using a quasi-implicit scheme: local and dual time stepping approaches, Comput. Mech. 50 (2012) 687–693.
[53] S. Wilhelm, J. Jacob, P. Sagaut, An explicit power-law-based wall model for lattice Boltzmann method-Reynolds-averaged numerical simulations of the flow around airfoils, Phys. Fluids 30 (2018) 065111.
[54] I. Ramière, Convergence analysis of the Q1-finite element method for elliptic problems with non-boundary-fitted meshes, Int. J. Numer. Methods Eng. 75 (9) (2008) 1007–1052.
[55] TRUST platform SourceForge mainlink, URL https://sourceforge.net/projects/trust-platform/.
[56] TRUST Generic Guide V1.8.0, URL https://sourceforge.net/projects/trust-platform/files/TRUST_Generic_Guide.pdf/download.
[57] M. Grandotto, J. Cheissoux, J. Gaillard, E. De Langre, P. Obry, M. Bernard, A 3D finite element code for solving two-phase flow analysis in PWR steam generators, in: European Two-Phase Flow Group Meeting, Joint Research Center Ispra, Varese, Italy, 1990.
[58] E. Guyon, J.-P. Hulin, L. Petit, Physical Hydrodynamics, Oxford University Press, 2001.
[59] P.M. Gresho, S.T. Chan, R.L. Lee, C.D. Upson, A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations. Part 1: theory, Int. J. Numer. Methods Fluids 4 (6) (1984) 557–598.
[60] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (2) (1999) 209–240.
[61] J.-I. Choi, R.C. Oberoi, J.R. Edwards, J.A. Rosati, An immersed boundary method for complex incompressible flows, J. Comput. Phys. 224 (2) (2007) 757–784.
[62] K. Taira, T. Colonius, The immersed boundary method: a projection approach, J. Comput. Phys. 225 (2) (2007) 2118–2137.
[63] M.N. Linnick, H.F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (1) (2005) 157–192.
[64] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, J. Fluid Mech. 98 (4) (1980) 819–855.
[65] D.J. Tritton, Experiments on the flow past a circular cylinder at low Reynolds numbers, J. Fluid Mech. 6 (4) (1959) 547–567.
[66] M.-H. Chung, Cartesian cut cell approach for simulating incompressible flows with rigid bodies of arbitrary shape, Comput. Fluids 35 (6) (2006) 607–623.
[67] D. Stojković, M. Breuer, F. Durst, Effect of high rotation rates on the laminar flow around a circular cylinder, Phys. Fluids 14 (9) (2002) 3160–3178.
[68] D. Ingham, T. Tang, A numerical investigation into the steady flow past a rotating circular cylinder at low and intermediate Reynolds numbers, J. Comput. Phys. 87 (1) (1990) 91–107.
[69] H. Badr, S. Dennis, P. Young, Steady and unsteady flow past a rotating circular cylinder at low Reynolds numbers, Comput. Fluids 17 (4) (1989) 579–609.
[70] P. Chiu, R. Lin, T.W. Sheu, A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier-Stokes equations in time-varying complex geometries, J. Comput. Phys. 229 (12) (2010) 4476–4500.
[71] Y. Cheny, O. Botella, The ls-stag method: a new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties, J. Comput. Phys. 229 (4) (2010) 1043–1076.
[72] C. Ji, A. Munjiza, J. Williams, A novel iterative direct-forcing immersed boundary method and its finite volume applications, J. Comput. Phys. 231 (4) (2012) 1797–1821.
[73] C. Norberg, Fluctuating lift on a circular cylinder: review and new measurements, J. Fluids Struct. 17 (1) (2003) 57–96.
[74] S. Kang, H. Choi, S. Lee, Laminar flow past a rotating circular cylinder, Phys. Fluids 11 (11) (1999) 3312–3321.
[75] R. Swanson, S. Langer, Steady-state laminar flow solutions for NACA 0012 airfoil, Comput. Fluids 126 (2016) 102–128.