A Second Order Penalized Direct Forcing for Hybrid Cartesian/Immersed Boundary Flow Simulations

Clement Introini, Michel Belliard, Clarisse Fournier

European Congress on Computational Methods
in Applied Sciences and Engineering (ECCOMAS 2012)
J. Eberhardsteiner et.al. (eds.)
Vienna, Austria, September 10-14, 2012

A SECOND ORDER PENALIZED DIRECT FORCING FOR HYBRID CARTESIAN/IMMERSED BOUNDARY FLOW SIMULATIONS

C. Introïni[1], M. Belliard[1], and C. Fournier[2]

[1]Commissariat à l'Énergie Atomique et aux Énergies Alternatives
CEA, DEN/DANS/DM2S/STMF/LMEC
CEA Cadarache, Bt. 238, F-13108 St Paul-lez-Durance
e-mail: {clement.introini,michel.belliard}@cea.fr

[2] Commissariat à l'Énergie Atomique et aux Énergies Alternatives
CEA, DEN/DANS/DM2S/STMF/LMSF
CEA Grenoble, Bt 1005, 17 rue des Martyrs, F-38054 Grenoble Cedex 9
e-mail: clarisse.fournier@cea.fr

Keywords: immersed boundaries, penalized direct forcing, projection scheme, Cartesian grids, interpolation scheme

Abstract. *Flows around complex stationary/moving solids take an important place in life-science context or in many engineering applications. Usually, these problems are solved by body-fitted approaches on unstructured meshes with boundary conditions directly imposed on the domain boundary. Another way is using immersed boundary (IB) techniques: the physical domain is immersed in a fixed fictitious one of simpler geometry on Cartesian grids. It allows to use efficient, fast and accurate numerical methods avoiding the tedious task of re-meshing in case of time varying geometry. In contrast, one needs specific methods to take into account the IB conditions (IBC). Here, we propose a second order penalized direct forcing method for unsteady incompressible flows with Dirichlet's IBC. It consists in adding a penalized forcing term to the initial problem, applied only on Cartesian nodes near the IB, in order to bring back the variable to the imposed one. Regarding Navier-Stokes solvers using a projection scheme, the forcing term is distributed both in the velocity prediction and in the correction equations. It leads to a natural way to prescribe the pressure boundary conditions around obstacles. Numerical experiments, performed for laminar flows around static/moving solids, assess the validity and illustrate the ability of our method, showing in particular a quadratic convergence rate.*

1 INTRODUCTION

Fluid flows with heat and mass transfers around/inside complex stationary or moving geometries appear in a large number of situations of practical interest including biological fluid mechanic applications (*e.g.* blood flow in heart) or engineering applications (*e.g.* heat exchangers, aerospace vehicles or nuclear safety). The numerical treatment of these kinds of problem appears to be a challenging task because of time varying geometries, often combined with complex flow regimes. To tackle numerically these complex problems, the well-known body-fitted approach is usually followed. Such an approach consists in discretizing the governing equations on a non-structured mesh for which the boundaries of the computational domain coincide with those of the physical domain. Thereby, boundary conditions are directly imposed on the physical domain boundary. However, the main drawbacks of the body-fitted like techniques lie in their lack of ability to handle complex industrial problems involving moving bodies which require the development of complex numerical schemes to deal with the difficult issue of re-gridding.
Another approach consists in using non-boundary conforming techniques in which the physical domain is immersed in a fixed fictitious one of simpler geometry on Cartesian grids. Such techniques allow to use efficient, fast and accurate numerical methods avoiding the tedious task of re-meshing caused by time-varying geometries. In contrast, as the physical boundaries are described by a set of Lagrangian points that do not generally coincide with those of the Eulerian grid or by a level-set function, numerical methods are needed to account for the Immersed Boundary conditions (IBC). Roughly, the non-boundary conforming techniques proposed in literature may be classified in two categories.
The first category, including for instance Cartesian methods (*e.g.* [1, 2]), the Immersed Interface Method (IIM) [3] or the Jump Embedded Boundary Condition method (JEBC) [4], mimicks the presence of embedded geometries by modifying the numerical scheme in the immediate vicinity of the immersed interface. Such an approach leads to a sharp representation of the immersed interface.
In the second category, rather than changing locally the numerical scheme, a forcing term is added in the governing equations. Since the Peskin's pioneering work of the Immersed Boundary Method (IBM) [5], several IB like methods with different forcing terms (or forcing strategies) have been proposed in literature: *e.g.* the Goldstein's Feedback Forcing method [6] or the Direct Forcing (DF) method proposed by Mohd-Yusof [7] and then adapted by Fadlun *& al.* [8]. The DF technique consists in directly applying the desired boundary conditions on the Cartesian nodes closest to the interface leading to a sharper representation of the interface than in the Peskin's method. In that sense, by using the terminology employed in [9], the DF method may be referred to as a hybrid Cartesian/Immersed Boundary (HCIB) approach. Since its development by Mohd-Yusof [7], the DF method have gained in popularity and have been successfully applied to various fluid-structure interaction problems (*e.g.* [9, 10, 11, 12, 13, 14]). More recently, Belliard & Fournier [15, 16] have proposed a variant of HCIB techniques, called Penalized Direct Forcing (PDF) method, that combines both the basic features of the DF method and those of L^2-penalty methods (*e.g.* [17]). Links can be found with the works of Sarthou *& al.* [18] and those of Bergmann & Iollo [19].

In the present paper, after introducing the discretization of the governing equations in the Section 2, our PDF algorithm including the velocity reconstruction near the immersed boundaries is detailed in the Section 3. We have developed an original robust interpolation scheme,

second-order accurate in space, that relies mainly on an averaged reconstruction of the velocity gradient near the IB and on an approximate projection operator onto the IB. Finally, in Section 4, some 2D numerical experiments are performed for laminar flows around/between solids to assess the validity and the ability of the proposed method both for IBC. Without loss of generality, we restrict our presentation to Dirichlet's IBCs. We show in particular that the numerical rate of convergence is quasi-quadratic for all studied cases.

2 Governing equations and numerical scheme

In this section, we focus on the governing equations and we present the resolution method, based on a fractional step method, and the space and time discretizations.

2.1 Governing equations

The governing equations used to described unsteady incompressible flows around complex obstacles Ω are the incompressible Navier-Stokes equations:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \nabla P - \nu \nabla^2 \mathbf{u} &= \mathbf{f} \quad \text{in } \Omega & \text{(1a)} \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{in } \Omega & \text{(1b)} \\ \mathbf{u} &= \mathbf{u}_D \text{ on } \partial\Omega \text{ and } \mathbf{u}(t_0) \text{ given in } \Omega & \text{(1c)} \end{aligned}$$

where $\mathbf{u}$ denotes the solenoidal velocity and ν the kinematic viscosity. Here above, P is the total pressure defined by $\rho \nabla P = \nabla p - \rho \mathbf{g}$ where p is the hydrodynamic pressure, ρ the constant density and $\mathbf{g}$ the gravity force. For clarity reason, we have assumed full Dirichlet boundary conditions and we will consider $\mathbf{f} = \mathbf{0}$ in the rest of this paper.

2.2 Space and time discretization

The time advancement of the velocity is performed by means of a degenerate fourth - order explicit Runge-Kutta scheme [20]. Given $\mathbf{u}^0 = \mathbf{u}(t_0)$ and $\frac{\partial \mathbf{u}}{\partial t} = f(\mathbf{u})$, the new velocity $\mathbf{u}^{n+1} = \mathbf{u}(t_0 + (n+1)\Delta t)$ is obtained by: $\forall\, n \in \mathbb{N}$, $\mathbf{u}^{n+1} = \mathbf{u}^n + \Sigma_{k=1}^{3} \beta_k \mathbf{q}_k$ and $\mathbf{q}_k = \Delta t f\left(\mathbf{u}^n + \Sigma_{m=1;k>1}^{k-1} \beta_m \mathbf{q}_m\right) + \alpha_k \mathbf{q}_{k-1}$ with α_k equals to 0, $-\frac{1}{2}$ and -2 and β_k equals to $\frac{1}{2}$, 1 and $\frac{1}{6}$ for $k = 1, 2$ or 3 respectively. At each stage k of the Runge-Kutta scheme, we get $f(\mathbf{u}^n + ...)$ by solving the couple $(\mathbf{u}, P)$ by a fractional step method, c.f. Section 2.2.1, for which the temporal discretization is based on a semi-implicit scheme: explicit discretization of the convection term and implicit discretization of the diffusion.
The space discretisation is based on a finite volume approximation with a staggered grid arrangement of the primitive variables $(\mathbf{u}, P)$ (velocity components at the middle of the cell edges and pressure at the cell center). In this frame, the governing equations Eqs. (1) are integrated over each control volumes ensuring the conservation of mass and momentum balance. The convection and diffusion terms are respectively approached by the QUICK and the centered schemes. We denote by $\star_h$ a discrete space operator $\star$. At each stage $k = 0, .., 3$ of the Runge-Kutta scheme, the discrete form of the governing equations given by Eqs. (1) reads

$$\begin{aligned} f(\mathbf{u}^{k-1}) &= \frac{\mathbf{u}^* - \mathbf{u}^{k-1}}{\Delta t} \text{ in } \Omega & \text{(2a)} \\ \frac{\mathbf{u}^* - \mathbf{u}^{k-1}}{\Delta t} + \nabla_h \cdot \left(\mathbf{u}^{k-1} \otimes \mathbf{u}^{k-1}\right) + \nabla_h P^* &= \nu \nabla_h^2 \mathbf{u}^* \text{ in } \Omega & \text{(2b)} \\ \nabla_h \cdot \mathbf{u}^* &= 0 \quad \text{in } \Omega & \text{(2c)} \end{aligned}$$

with suitable boundary on $\partial\Omega$, $\mathbf{u}^k = \mathbf{u}^n + \Sigma_{m=1}^{k} \beta_m \mathbf{q}_m$, $\mathbf{u}^{k-1} = \mathbf{u}^n$ for $k = 1$ and $(\mathbf{u}^{n+1}, P^{n+1}) = (\mathbf{u}^k, P^*)$ for $k = 3$.

2.2.1 A projection algorithm

The fractional step method or projection method was introduced by Chorin and Temam in 1968 for incompressible flows [21, 22]. On the basis of their work, many variants have been proposed, as for example, the incremental projection method for incompressible flows [23], the projection schemes for dilatable or barotropic fluids [24, 25] or, more recently, the novel fractional time stepping technique massively parallel for incompresible Navier-Stokes equations developed by Guermond & Minev [26], to cite among other. We refer the reader to [27] for a recent review of these methods.
In our case, the first step of the algorithm consists in solving a predicted velocity $\widetilde{\mathbf{u}}$ without pressure gradient term as follows[1]:

$$\frac{\widetilde{\mathbf{u}} - \mathbf{u}^n}{\Delta t} + \nabla_h \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) - \nu \nabla_h^2 \widetilde{\mathbf{u}} = 0 \quad \text{in } \Omega \tag{3a}$$

$$\widetilde{\mathbf{u}} = \mathbf{u}^{n+1} = \mathbf{u}_D \text{ on } \partial\Omega \tag{3b}$$

The second step of the algorithm corresponds to a correction stage which consists in computing a new pressure P^{n+1} and recovering a new solenoidal velocity $\mathbf{u}^{n+1}$. By assuming that $\nabla_h^2 \widetilde{\mathbf{u}} \sim \nabla_h^2 \mathbf{u}^{n+1}$, this step reads

$$\frac{\mathbf{u}^{n+1} - \widetilde{\mathbf{u}}}{\Delta t} = -\nabla_h P^{n+1} \text{ in } \Omega \tag{4a}$$

$$\nabla_h \cdot \mathbf{u}^{n+1} = 0 \text{ in } \Omega \tag{4b}$$

$$\mathbf{n} \cdot \widetilde{\mathbf{u}} = \mathbf{n} \cdot \mathbf{u}^{n+1} = \mathbf{n} \cdot \mathbf{u}_D \text{ on } \partial\Omega \tag{4c}$$

Finally, the correction Eqs. (4) allows to compute the new velocity $\mathbf{u}^{n+1}$ as follows

$$\mathbf{u}^{n+1} = \widetilde{\mathbf{u}} - \Delta t \nabla_h P^{n+1} \text{ in } \Omega \tag{5}$$

3 Penalized Direct Forcing

As previously mentioned, the immersed boundary technique consists in immersing the physical domain in a fixed fictitious one. Here above, we will denote by Ω_f the physical domain (fluid) and by Ω the fictitious domain discretized on a Cartesian mesh. We denoted by Ω_s the external domain (solid) defined by $\Omega_s = \Omega \backslash \Omega_f$. The domains Ω_f and Ω_s are separated by an immersed boundary Σ discretized by a set of Lagrangian points. In the numerical experiments presented in the present paper, the interface tracking is performed by means of a Front-Tracking technique, c.f. for instance [28, 29]. We will use subscript h for discrete quantities.
To begin, we introduce in a first part the concept of the penalized direct forcing method and we present a fractional step method consistent with the penalized direct forcing [15]. Then, in a second part, we focus on the interpolation schemes in order to reconstruct accurately the velocity field (second order in space) near the immersed boundary. A detailed presentation can be found in [16].

[1] But it should be with the pressure at the previous time step or an extrapolation of the pressure as well.

3.1 Penalized Direct Forcing

In the frame of immersed boundary-like methods [5, 7, 8], the governing equations used to described unsteady incompressible flows around complex geometries are given by:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \nabla P - \nu \nabla^2 \mathbf{u} = \mathbf{F} \quad \text{in } \Omega, \tag{6a}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega. \tag{6b}$$

Eqs. (6) are the incompressible Navier-Stokes Eqs. (1) with a penalized forcing term $\mathbf{F}$ here defined by [15]:

$$\mathbf{F} = \frac{\alpha \chi_s}{\eta} (\mathbf{u}_i - \mathbf{u}) \text{ with } \alpha > 0,\ 0 < \eta \ll 1 \text{ and } \chi_s \rightarrow [0, 1] \tag{7}$$

where χ_s is the characteristic function of the obstacle, η a penalty coefficient (in this work $\eta = 10^{-12}$) and $\mathbf{u}_i$ the imposed fluid velocity around/inside the obstacle Ω_s depending on the obstacle velocity $\mathbf{u}_s$ but also on the fluid velocity $\mathbf{u}$ following the requested accuracy. Eq. (7) can be view as an implicit limit version of direct forcing expression [7, 16]. The forcing term $\mathbf{F}$ is only applied on Cartesian nodes near or inside the immersed boundary (*i.e.* $\chi_s(\mathbf{x}) > 0$), leading to $\mathbf{u}_i(\mathbf{x}) = \mathbf{u}(\mathbf{x})$. Outside, the classical incompressible Navier-Stokes equations are recovered in the fluid domain Ω_f (*i.e.* $\chi_s(\mathbf{x}) = 0$).

3.2 Resolution of the Navier-Stokes equations: a consistent fractional step method

In this section, we propose a fractional step method to solve the Navier-Stokes equations Eqs. (6) in the framework of the Penalized Direct Forcing. The new feature of our algorithm is that the forcing term is distributed both in the prediction and the correction stages of the projection. This leads to a natural consistent scheme in the sense that the immersed boundary conditions are well satisfied not only by the predicted velocity in the prediction stage but also by the new velocity at the end of the projection stage [30].

As usual, the first step of our scheme consists in solving a predicted velocity $\widetilde{\mathbf{u}}$, see Eqs. (3), with the forcing term given by Eq. (7) and $\alpha = \frac{1}{\Delta t}$:

$$\frac{\widetilde{\mathbf{u}} - \mathbf{u}^n}{\Delta t} + \nabla_h \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) - \nu \nabla_h^2 \widetilde{\mathbf{u}} = \frac{\chi_s}{\eta \Delta t} (\mathbf{u}_i^{n+1} - \widetilde{\mathbf{u}}) \quad \text{in } \Omega \tag{8}$$

In this equation, the imposed fluid velocity $\mathbf{u}_i^{n+1}$ depends on the obstacle velocity and on the fluid velocity around the obstacle. This dependency is here treated in an explicit time-discretization way, solving first a free-obstacle explicit N.-S. equations ($\widetilde{\mathbf{u}}^\star$). Then the imposed velocity $\mathbf{u}_i^{n+1}$ is expressed in terms of $\mathbf{u}_s^{n+1}$ and $\widetilde{\mathbf{u}}^\star$. The solution of Eq. (8) respects the Dirichlet boundary conditions on Σ_h up to a given order depending on the space interpolation scheme, see Section 3.3.

The second step consists in a correction stage that reads as usual, see System (4), except for the Eq. $(4a)$:

$$\frac{\mathbf{u}^{n+1} - \widetilde{\mathbf{u}}}{\Delta t} = -\nabla_h P^{n+1} + \frac{\chi_s}{\eta \Delta t} (\widetilde{\mathbf{u}} - \mathbf{u}^{n+1}) \text{ in } \Omega \tag{9}$$

that can be simplified as

$$\check{\rho} \frac{\mathbf{u}^{n+1} - \widetilde{\mathbf{u}}}{\Delta t} = -\nabla_h P^{n+1} \text{ in } \Omega. \tag{10}$$

This equation is similar to (4a) with a modified density $\check{\rho} = (1 + \chi_s/\eta)$. Therefore the rest of the projection algorithm is similar to the standard one using this modified density $\check{\rho}$. In fact, we get a consistent fractional step scheme in the sense that the Dirichlet immersed boundary condition is well satisfy by the new velocity $\mathbf{u}^{n+1}$ (not only by $\widetilde{\mathbf{u}}$) and that the homogeneous Neumann immersed boundary condition is satisfy by the pressure on the obstacles.

3.3 Reconstruction of velocity field: interpolation schemes

The immersed boundaries are described by a set of Lagrangian points which do not generally coincide with the nodes of the Eulerian mesh. Under these circumstances, the velocity field must be reconstructed near the immersed boundary in order to take into account the immersed boundary conditions. An simple approach consists in prescribing directly the velocity of the obstacle $\mathbf{u}_s$ on the Cartesian nodes closest to the immersed boundary Σ_h without any interpolation scheme [8] (referred here as the base model, first-order accurate in space). This model leads to a stepwise description of the immersed interface Σ_h.

Among the numerical algorithms proposed in literature to reconstruct the velocity field in the interfacial region, the most widely employed to improve the accuracy of the solution in the neighborhood of Σ_h is based on an interpolation or an extrapolation procedure (*e.g.* [10, 8, 30, 11, 9, 12]) involving solid and fluid velocities to calculate the velocity at the forcing nodes. Unlike the classical approaches, consisting in a local interpolation or extrapolation along a specific direction (often the grid-line directions or the orthogonal direction onto Σ_h), our proposed linear interpolation method, second-order accurate in space, involves an averaged reconstruction of the velocity gradient near the immersed boundary [16]. The fluid contribution is built following similar ideas used in [30] and the solid contribution is determined by means of a minimization problem. It consists in estimating the imposed fluid velocity $\mathbf{u}_i$ at the forcing or penalized node $\mathbf{x}$ according to

$$\begin{aligned} \mathbf{u}_i(\mathbf{x}) \;\; := \;\; & \mathbf{u}_s(\Pi_\Sigma(\mathbf{x})) + \frac{d(\mathbf{x})}{N} \sum_{p=1}^{N} \frac{\mathbf{u}(\mathbf{x}^p) - \mathbf{u}_s(\Pi_\Sigma(\mathbf{x}^p))}{d(\mathbf{x}^p)} + \mathcal{O}\left(h^2\right) \\ & \text{for } \mathbf{x} \in \Omega \text{ such that } \chi_s(\mathbf{x}) > 0 \text{ and } d(\mathbf{x}) > 0 \end{aligned} \tag{11}$$

where $\mathbf{x}^p$ is the p-th fluid node in the immediate vicinity of the penalized node $\mathbf{x}$. Hence, we take into account the local influence of the fluid flow around $\mathbf{x}$. $\Pi_\Sigma(\mathbf{x})$ denotes the projection of $\mathbf{x}$ onto the immediate neighborhood of Σ_h and is defined throughout an algorithm based on the following minimization problem:

$$\text{Find } \mathbf{z}\,(= \Pi_\Sigma(\mathbf{x})) \in \Omega \text{ such that } J(\mathbf{z}) \;\; = \;\; \inf_{\mathbf{y}\in\Omega} J(\mathbf{y}) \tag{12}$$

where $J(\cdot)$ is defined by $\forall\, \mathbf{y} \in \Omega,\ J(\mathbf{y}) = \|\mathbf{y} - \mathbf{x}\|^2$. To tackle this minimization problem, we partially reconstruct the immersed boundary in the immediate vicinity of $\mathbf{x}$, collecting all the Lagrangian facets m that belong to the cells intersected by Σ_h around $\mathbf{x}$ and determining for each of them his plan equation $\sum_{j=1}^{d} c_{mj} y_j = f_m$. Here d is the spatial dimension of the problem, y_j the coordinates of one vertex belonging to the m-th facet and c_{mj} the coordinates of the unit normal vector at the centroid of this facet. Then for all the facets associated with the Cartesian node $\mathbf{x}$, we have $\mathbf{C} \cdot \mathbf{y} \approx \mathbf{f}$.
To solve this minimization problem, we use an Uzawa algorithm which can be summarized as follows:

1. Initialisation ($k = 0$): we assume $\boldsymbol{\lambda}^0 = 0$
2. k-th iteration: by assuming λ^k known, we are able to compute

$$\star\ \mathbf{y}^{k+1} \text{ by solving } \mathbf{y}^{k+1} = \mathbf{x} - \frac{1}{2}\mathbf{C}\cdot\boldsymbol{\lambda}^k$$
$$\star\ \boldsymbol{\lambda}^{k+1} \text{ by solving } \boldsymbol{\lambda}^{k+1} = \max\left[\boldsymbol{\lambda}^k + \varrho\left(\mathbf{C}\cdot\mathbf{y}^{k+1} - \mathbf{f}\right), 0\right]$$

with $\varrho = \left(\|\mathbf{C}\|_1\|\mathbf{C}\|_\infty\right)^{-1}$ and $\boldsymbol{\lambda}^k$ the vectorial Lagrange multiplier at k-th iteration associated with the constraint $\mathbf{C}\cdot\mathbf{y}^{k+1} = \mathbf{f}$. There may be cases in which several Cartesian cells contain a significant number of Lagrangian facets, for instance when the mesh is generated by CAD, leading to a prohibitive CPU-time cost. To overcome this difficulty, we only selects the most representative facets and detects also those that are collinear in order to remove the duplicate. This algorithm is very robust and $z = \Pi_\Sigma(\mathbf{x}) \in \Omega$ converges toward $z \in \Sigma_h$ when h is decreasing.

4 Numerical experiments

This section is devoted to the numerical validation of the proposed Penalized Direct Forcing method. On the one hand, Poiseuille flow in an inclined channel, cylindrical Couette flow and steady flow around static/rotating cylinder are considered to assess the validity and the ability of our method both for Dirichlet's IBC. Grid convergence studies have been done in order to obtain the numerical rate of convergence of the method in the $\mathrm{L}^2(\Omega_f)$ norm ε_2 and the $\mathrm{L}^\infty(\Omega_f)$ norm ε_∞ given by $\varepsilon_2 = \sqrt{\sum_{i=0}^{N_f}\left(u^i - u^i_{ref}\right)^2}/\sqrt{\sum_{i=0}^{N_f}{u^i_{ref}}^2}$ and $\varepsilon_\infty = \max_{0\leq i\leq N_f}\left(|u^i - u^i_{ref}|\right)$ where the upperscript i denotes the i-th face of the Eulerian grid, u^i_{ref} a referenced velocity calculated on the i-th face and N_f the total number of faces in the fluid region Ω_f. All these academical numerical tests have been performed with both the linear interpolation scheme and the base model. On the other hand, interestingly for pratical purposes, 2D calculations involving unsteady laminar flow and the interpolation scheme are presented.

All the calculations have been performed by using the CFD code Trio_U developed at the French "Commissariat à l'Énergie Atomique et aux Énergies Alternatives" [31].

4.1 Poiseuille flow in an inclined channel

Here, the problem under consideration is the well-known test case of a Poiseuille flow in an inclined channel ($\theta = \pi/4$). The computational domain Ω is define by $\Omega_s \cup \Omega_f$ and the immersed boundary by $\Sigma = \Omega_s \cap \Omega_f$. Null pressure boundary conditions are imposed on $\partial\Omega$ except on the entry where we prescribed the following parabolic velocity $\mathbf{u}_{in} = \left(U_\infty - Y^2, 0\right)^{\mathrm{T}}$ with $Y = \frac{y-x}{\sqrt{2}}$, (X, Y) the system of coordinates in the frame associated with the inclined channel and $U_\infty = 0.605\ \mathrm{m.s}^{-1}$ the maximum velocity. For this numerical test, no-slip boundary condition are imposed on the immersed boundary Σ (*i.e.* $\mathbf{u}_s = \mathbf{0}$).
For a convergence study, four computational grid sizes have been considered over the range $25.10^{-3} \leq h \leq 2.10^{-1}$. Fig.1 represents the evolution of the $\mathrm{L}^2(\Omega_f)$ norm ε_2 and the $\mathrm{L}^\infty(\Omega_f)$ norm ε_∞ of the error. As expected, the proposed linear interpolation model leads to a quasi-quadratic numerical rate of convergence while the numerical order of the method using the base model is about one. We may remark that, as the whole, the errors in the case of the linear interpolation model are at least one order of magnitude lesser than the ones obtained with the base model. Indeed, the approximation obtained in the case of the coarsest grid with the linear

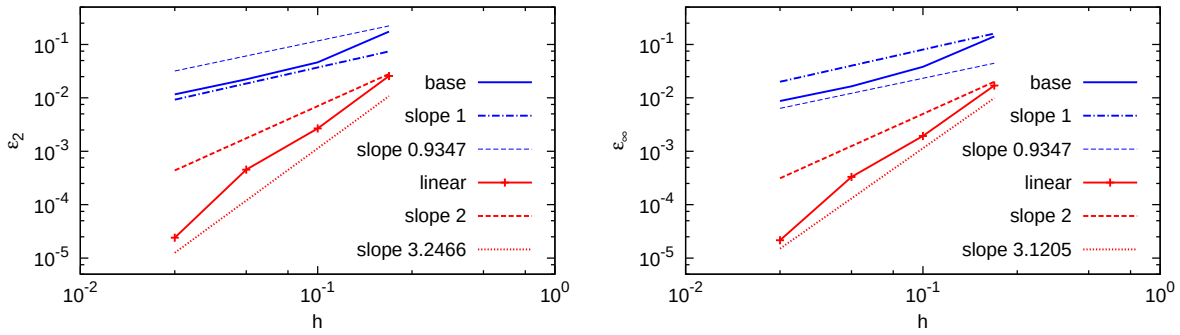


Figure 1: Poiseuille flow in an inclined channel: $\mathrm{L}^2(\Omega_f)$ norm (left) and $\mathrm{L}^\infty(\Omega_f)$ norm (right) of the error *vs.* the computational grid size h. Comparison of the linear interpolation model with the base model.

interpolation model is very close to the one computed with the base model on the finest grid. Therefore, for this test, the results clearly show that the proposed method coupled with the linear interpolation model allows to obtain an accurate solution at lower computational cost.

4.2 Taylor-Couette problem

Here, we focus on a steady flow between two rotating concentric cylinders, also called Taylor-Couette flow. This problem has already been considered in the frame of immersed boundary-like methods (*e.g.* [10, 32]). The computational domain is defined by a square $\Omega = \Omega_s \cup \Omega_f = [0, L] \times [0, L]$ in which are immersed the boundaries Σ_1 and Σ_2 mimicking the inner ($r_1 = 0.1\mathrm{m}$) and outer ($r_2 = 0.2\mathrm{m}$) cylinders, respectively. The length L is about $2\,(r_1 + r_2)$. The inner cylinder rotates clockwise ($\omega_1 = 1\mathrm{s}^{-1}$) while the outer cylinder rotates in the counterclockwise ($\omega_2 = -1\mathrm{s}^{-1}$). In overall calculations, we assume that $\rho = 1\mathrm{kg.m}^{-3}$, $\mu = r_1/\Re$ Pa.s and that the Reynolds number, defined by $\Re = |\omega_1| r_1^2/\nu$, is set to be 1. Such an assumption allows us to write $\mathrm{Ta} < \mathrm{Ta}_c$ where $\mathrm{Ta} = 3/2$ is the Taylor number defined by $\mathrm{Ta} = 0.5\Re^2(r_1 + r_2)(r_2 - r_1)^3/r_1^4$ and $\mathrm{Ta}_c = 1.712$ is the theoretical critical one [33]. This relation implies that the following simulations remain strictly planes. Regarding the boundary conditions, we prescribed symmetry conditions on the boundaries of the computational domain

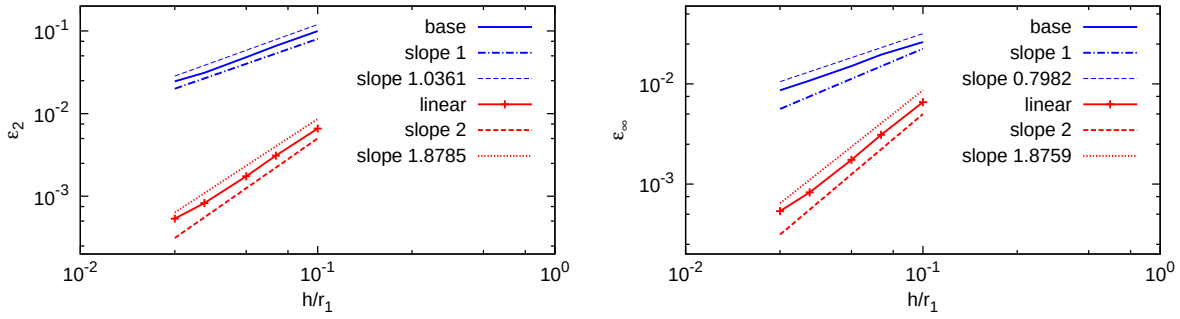


Figure 2: Taylor-Couette problem: $\mathrm{L}^2(\Omega_f)$ norm (left) and $\mathrm{L}^\infty(\Omega_f)$ norm (right) of the error *vs.* the ratio of the computational grid size h over the smallest radius. Comparison of the linear interpolation model with the base model.

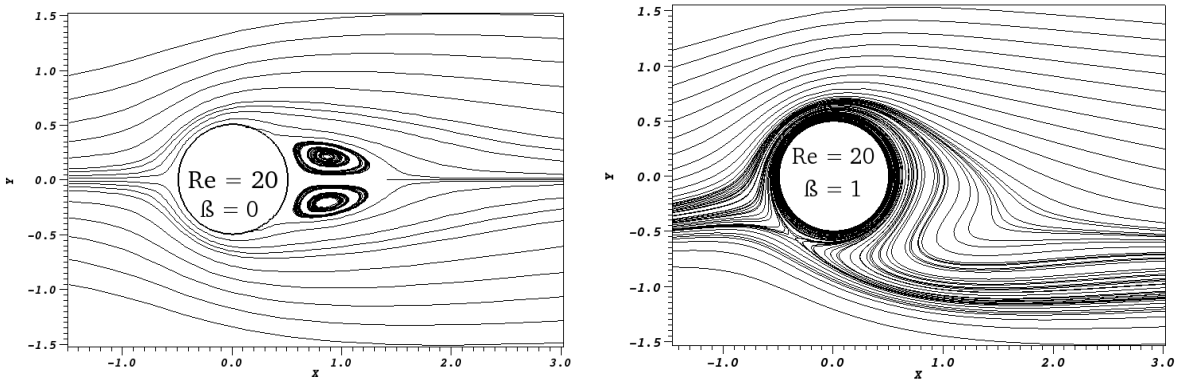
$\partial\Omega$ and the following analytical velocities on the immersed boundaries Σ_1 and Σ_2: $\forall\, i = 1, 2$ $\mathbf{u}_s = \omega_i r_i \mathbf{e}_r$ on Σ_i where $\mathbf{e}_r$ is the radial unit vector. The flow is initially at rest and converges toward a steady state.
Here again, a grid convergence study have been done in order to estimate the numerical order of the proposed method. Five meshes have been considered over the range $25.10^{-3} \leq h/r_1 \leq 10^{-1}$ where h denotes the step size. Fig. 2 presents the $\mathrm{L}^2(\Omega_f)$ norm ε_2 and the $\mathrm{L}^\infty(\Omega_f)$ norm ε_∞ of the error. As in the case of the Poiseuille flow, a quasi-linear numerical rate of convergence is obtained with the base model and a quasi-quadratic rate with the linear interpolation scheme. Finally, as previously observed, the graphs presented in Fig.2 show that the calculations performed on the coarsest grid with the linear interpolation scheme give better results (in terms of the error magnitude) than those of the finest grid with the base model.

4.3 Laminar flows around a cylinder

In this part, we are concerned with laminar flows around a circular cylinder. This problem has been the object of many experimental and numerical studies. In the latter case, problems involving steady and unsteady fluid flows past a static cylinder receive a particular attention. In contrast, to our knowledge, the case of fluid flows around rotating cylinder have been rarely investigated.
In this paper, we consider laminar fluid flows around static and rotating cylinders. Lets $U = 1\,\mathrm{m.s}^{-1}$ be the oncoming velocity, $D = 1\,\mathrm{m}$ the cylinder diameter and $\nu = 1/\Re\ \mathrm{m}^2.\mathrm{s}^{-1}$ the kinematic viscosity of the fluid. Our simulations are performed both in the steady laminar regime (*i.e.* $\Re = \frac{UD}{\nu} \leq 47$ [33]) with $\Re = 20$ and in the unsteady laminar regime with $\Re = 100$ ($\rho = 1\mathrm{kg.m}^{-3}$ and $\mu = 1/\Re$ Pa.s). We also introduce the dimensionless number β in order to characterize the rotation of solid. It is defined as the ratio of the rotational velocity ω over the oncoming velocity and reads $\beta = \frac{D\omega}{2U}$. Here, the rotational velocity ω is equal to $2\,\mathrm{s}^{-1}$ ($\beta = 1$). The computational domain Ω corresponds to a square of length L with an immersed cylinder centered at the coordinate origin. The boundaries of the computational domain $\partial\Omega$ must be located sufficiently far enough to reduce the impact of boundary conditions on vortex development behind the cylinder [34]. In this work, the ratio of the length L over the diameter of the cylinder D is set to 60. This allows us to obtain, with a reasonable computational cost, results in good agreement with those given in literature. Symmetry conditions are prescribed on $\partial\Omega$ except at th e in-flow boundary (uniform normal velocity $u_{in} = U$ and null tangential velocity) and at the out-flow one (null pressure). Regarding the immersed Dirichlet boundary conditions on Σ, we have: $\mathbf{u}_s = \frac{\omega D}{2}\mathbf{e}_r$ where $\mathbf{e}_r$ is the radial unit vector. In the case of a non-rotating cylinder (*i.e.* $\omega = 0$), it boils down to a no-slip boundary condition.

4.3.1 Steady laminar case: $\Re = 20$

Qualitatively, the linear interpolation model and the base one provide similar results concerning the flow pattern. Fig.3(a) shows the streamlines contours obtained with a static cylinder. In this case, the flow pattern is characterized by a pair of two steady symmetric vortices attached to the surface of the cylinder in agreement with the results proposed in literature (*e.g.* [35, 13, 1, 36, 37, 38]). The rotation of the cylinder disturbs the flow pattern as illustrated in Fig.3(b). The flow becomes asymmetric with especially, a complete disappearing of the two vortices located behind the cylinder in the static case. Again, these results are in very good agreement with [2] and also concur with those published in [34]. At this stage, we refer the reader to the work of Stojković & *al.* [34] in order to understand the effect of rotation on the



(a) Static cylinder (b) Rotating cylinder

Figure 3: Streamlines around rotating cylinder obtained with the linear interpolation scheme: $\Re = 20$

		D/h			References					
		10	20	40	[1]	[13]	[35]	[36]	[37]	[38]
C_d	base	2.066	2.094	2.059	2.03	2.02	2.06	2.06	2.00	2.09
	linear	2.085	2.071	2.054						
$\frac{L_w}{D}$	base	0.82	0.98	0.925	0.92	0.9	0.94	0.93	0.91	-
	linear	0.92	0.91	0.9						

Table 1: Hydrodynamic coefficients associated with the problem of steady flow around static cylinder

flow around the cylinder. Quantitatively, the linear interpolation model and the base one are compared in terms of drag coefficient $C_d = F_x/0.5U^2D$ (static and rotating cases), lift coefficient $C_l = F_y/0.5U^2D$ (rotating case), recirculation length L_w (static case) and direction of the total force $\theta = \tan^{-1}(C_l/C_d)$ (rotating case). Here, F_x and F_y correspond respectively to the tangential and normal components of the total force $\mathbf{F}_t$. This comparison is done by using three levels of refinement ranging from $D/h = 10$ to $D/h = 40$ with h the size of Cartesian cells. The physical coefficients obtained with a static cylinder are summarized in Tab.1 and confronted with the set of data proposed in [35, 13, 1, 36, 37, 38]. Whatever the number of cells D/h is, the linear interpolation model is in good agreement with literature concerning the evaluation of the couples of coefficients $(C_d, L_w/D)$, whereas the finest grid resolution is required by the base model. Tab.2 summarizes the obtained values of C_d, C_l and θ in the rotating case

		D/h			References			
		10	20	40	[2]	[34]	[39]	[40]
C_d	base	1.8968	1.8703	1.8608	1.888	~ 1.85	1.925	2.000
	linear	1.9104	1.8746	1.8679				
C_l	base	3.0284	3.1097	2.9419	2.629	~ 2.75	2.617	2.740
	linear	2.6248	2.7740	2.7745				
θ	base	57.93°	58.97°	57.68°	54.31°	$\sim 56°$	53.66°	53.87°
	linear	53.95°	55.95°	56.05°				

Table 2: Hydrodynamic coefficients associated with the problem of steady flow around rotating cylinder

and are compared with those given in [34, 2, 39, 40]. The trend observed is similar to the one previously noted in the static case and, as expected, the linear interpolation model yields better results than the base model.

Restricting ourselves to the behavior of the penalized direct forcing method, a smaller com-

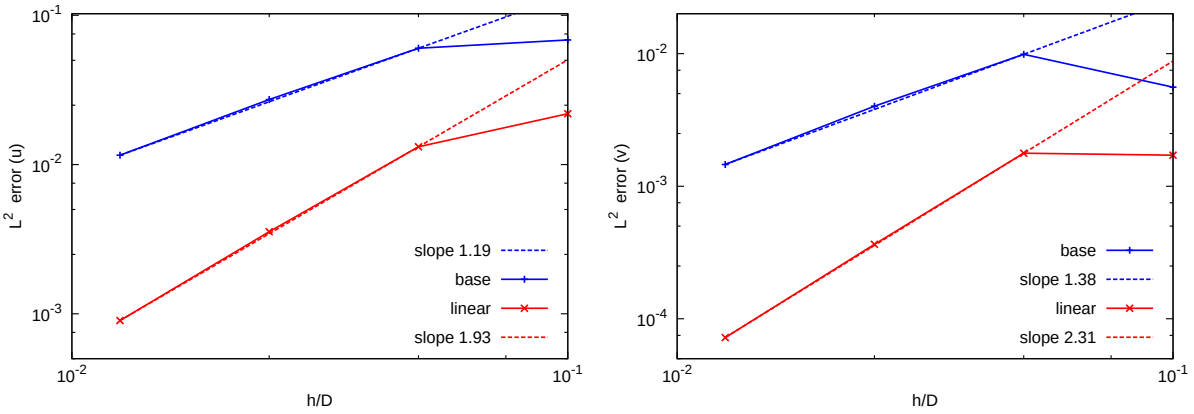


Figure 4: Static case: $\mathrm{L}^2(\Omega_f)$ norm error of the streamwise u (left) and spanwise v (right) velocity components *vs.* the ratio of the computational grid size h over the smallest radius. Errors calculated on the whole fluid domain. Comparison of the linear interpolation model with the base model.

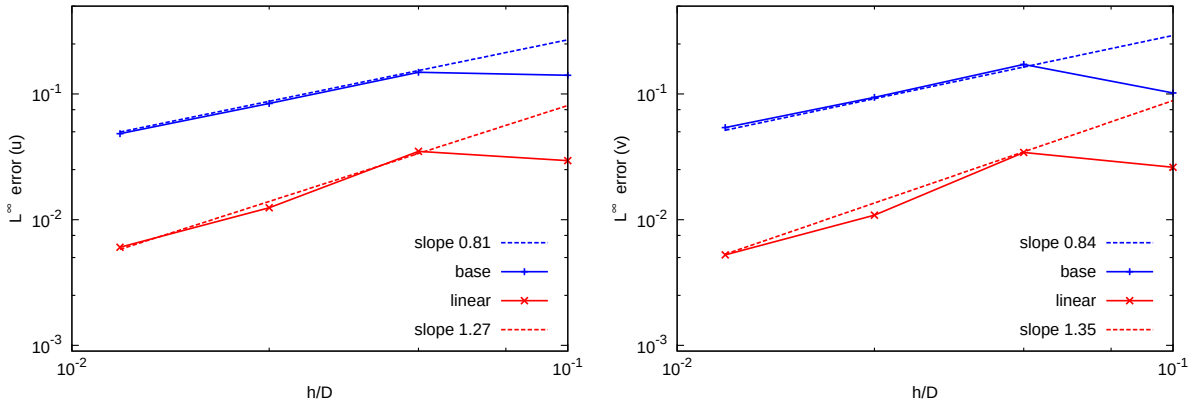


Figure 5: Static case: $\mathrm{L}^\infty(\Omega_f)$ norm error of the streamwise u (left) and spanwise v (right) velocity components *vs.* the ratio of the computational grid size h over the smallest radius. Errors calculated on the whole fluid domain. Comparison of the linear interpolation model with the base model.

putational domain is chosen, namely $\Omega = [-10D, 10D] \times [-10D, 10D]$, to perform a grid convergence study, allowing to consider fine grids with a reasonable computational cost. Finest grid solution is chosen as the reference solution to compute the error norms ε_2 and ε_∞. Five levels of refinement are used ranging from $h/D = 10^{-1}$ to $h/D = 6.25 \times 10^{-3}$. Fig.4 and Fig.5 presents, respectively, the evolution of error norms ε_2 and ε_∞ measured on the whole fluid domain. As expected, the numerical rate of convergence in L^2 norm of the method employed with the linear interpolation scheme is consistent with the second order in space and the one calculated with the base model is slightly higher that one. However, the convergence in L^∞ norm is lower with a numerical rate of convergence close to one for the both approaches. But, as previously point out, the linear interpolation scheme on the coarsest grid is again more accurate than the base model on the finest grid. In fact, if we compute the ε_∞ norm on 90% of the fluid domain (far from the immersed boundary), the numerical rate of convergence tends to the second order for the linear interpolation scheme (1.87 for u and 1.84 for v). This indicates

that the maximum of the error is located in the immediate vicinity of the immersed interface, as already pointed out in the frame of Cartesian methods by Cheny and Botella [32]. Concerning the case of the rotating cylinder, Fig.6 and Fig.7 present the evolution of the ε_2 and ε_∞ norms measured on the whole fluid domain. These results confirm the trend observed with a static cylinder. The error norm ε_∞ is about 1.87 for u et 1.84 for v.

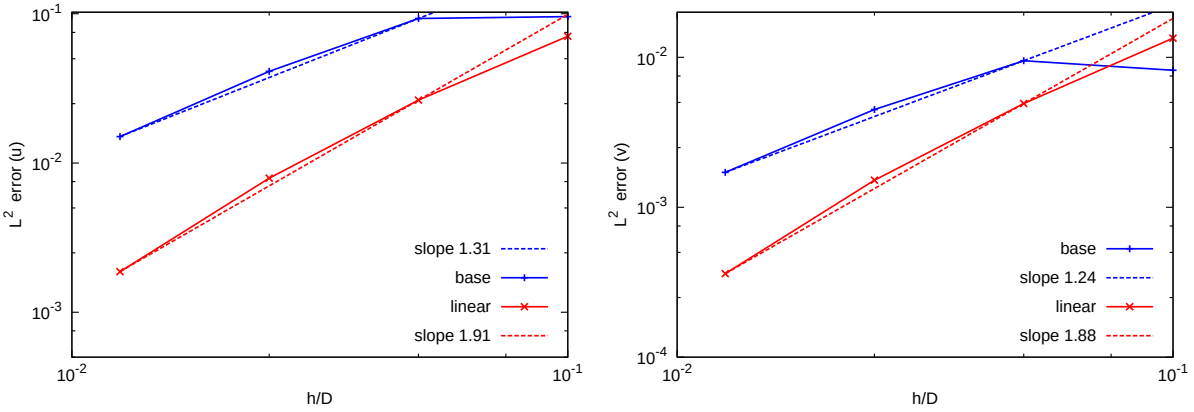


Figure 6: Rotating case: $L^2(\Omega_f)$ norm error of the streamwise u (left) and spanwise v (right) velocity components *vs.* the ratio of the computational grid size h over the smallest radius. Errors calculated on the whole fluid domain. Comparison of the linear interpolation model with the base model.

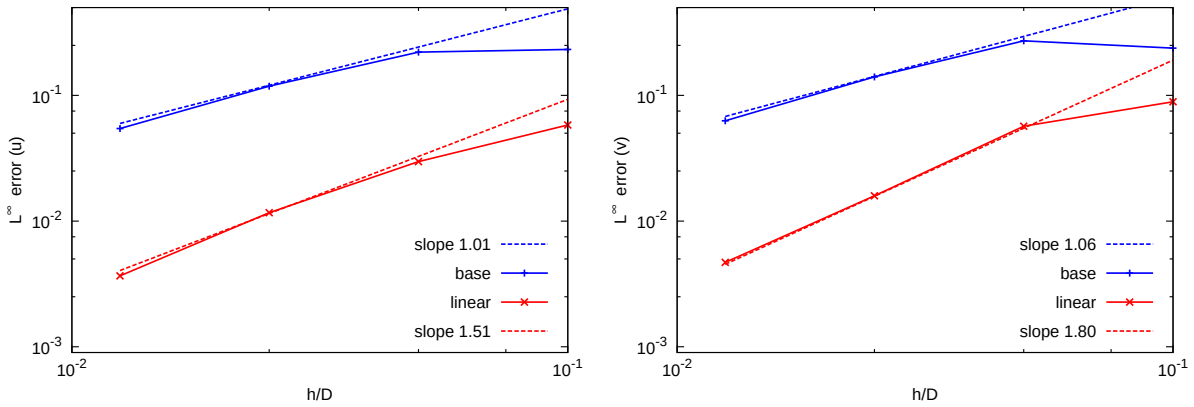
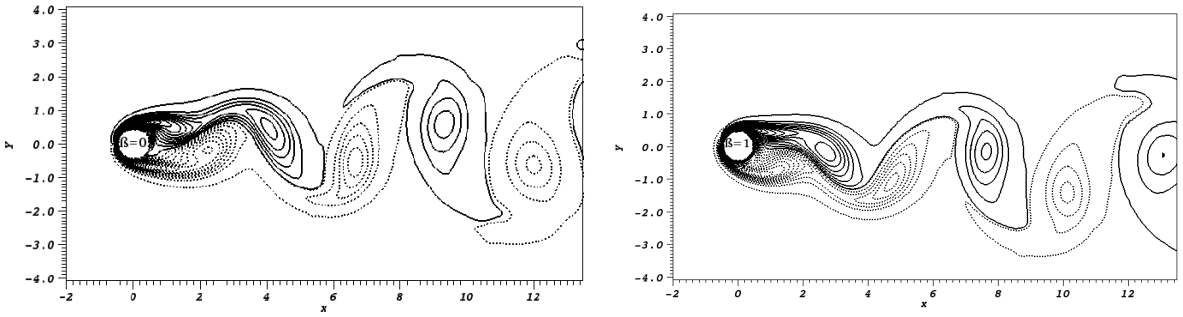


Figure 7: Rotating case: $L^\infty(\Omega_f)$ norm error of the streamwise u (left) and spanwise v (right) velocity components *vs.* the ratio of the computational grid size h over the smallest radius. Errors calculated on the whole fluid domain. Comparison of the linear interpolation model with the base model.

4.3.2 Unsteady laminar case: $\Re = 100$

Here we face the simulation of the unsteady regime with $\Re = 100$, using only the linear interpolation model. The geometrical features are identical to those considered previously in the steady case and all the calculations have been done by using a non-uniform grid with fifty Cartesian cells in the diameter of the cylinder.
Fig.8 presents an instantaneous view of vorticity contours. The flow pattern is characterized



(a) Static cylinder (solid lines: negative contours ; dotted lines: positive contours)

(b) Rotating cylinder (solid lines: negative contours ; dotted lines: positive contours)

Figure 8: Vorticity contours around static and rotating cylinders: $\Re = 100$

by the well-known Kármán vortex street (*e.g.* [34, 36, 2, 13, 12, 32, 14] in the static case and [34, 2, 41] in the rotating case) which are well captured by our immersed boundary technique. Tab.3 (static case) and Tab.4 (rotating case) present time averaged drag coefficient $\overline{C_d}$, time averaged lift coefficient $\overline{C_l}$, the amplitude of their fluctuations $C_d^{'}$ and $C_l^{'}$ and the Strouhal number $\mathrm{St} = fD/U$ which is the dimensionless number used to characterize the shedding frequency f estimated from the periodic variation of the lift coefficient C_l. The values of hydrodynamic coefficients obtained with the proposed penalized direct forcing method are in good agreement with those published in literature.

	Present results	References						
		[34]	[36]	[2]	[13]	[12]	[32]	[14]
$\overline{C_d}$	1.347	1.3371	1.34	1.392	1.34	1.35	1.317	1.3757
$C_d^{'}$	± 0.009	± 0.0091	± 0.009	$-$	± 0.011	± 0.012	± 0.009	± 0.0096
$C_l^{'}$	0.326	0.3259	0.333	$-$	0.315	0.303	0.349	0.3393
St	0.165	0.165	0.166	0.172	0.164	0.167	0.170	0.1692

Table 3: Hydrodynamic coefficients associated with the problem of unsteady flow around static cylinder

	Present results	References		
		[34]	[2]	[41]
$\overline{C_d}$	1.12	1.1080	1.189	1.0979
$C_d^{'}$	± 0.11	± 0.0986	± 0.1195	± 0.0988
$\overline{C_l}$	2.51	2.504	2.405	2.4833
$C_l^{'}$	± 0.37	± 0.3616	± 0.4427	± 0.3603
St	0.165	0.1658	0.1732	0.1650

Table 4: Hydrodynamic coefficients associated with the problem of unsteady flow around rotating cylinder

5 CONCLUSIONS

In this paper, we have presented the Penalized Direct Forcing method, using here a finite volume space approximation with a staggered grid arrangement of variables. Such a method

allows to solve on a Cartesian grid the Navier-Stokes equations with a penalized forcing term taking into account immersed boundary conditions (here Dirichlet BC, but it is possible to deal with Neumann BC). We have developed an original robust linear interpolation scheme in order to reconstruct the velocity field near the immersed boundary, not depending on a preferred direction, as often done in literature, but based on a minimization problem relying on a local reconstruction of the immersed boundary. Furthermore, the resolution of governing equations is done by using a fractional step scheme that is modified in such a way that the immersed boundary condition for velocity is verified not only in the prediction equation but also in the correction one.
Several numerical experiments have been carried out. All the results demonstrate the effectiveness and the potentiality of application of our method. First, the validity and the accuracy in space have been assessed throughout two academical problems: the numerical rate of convergence is quasi-quadratic in L^2- and L^∞-norms. Second, the problem of laminar flows around static and rotating cylinders has been studied both in the steady and unsteady regimes. Our results are in good agreement with those published in literature and a steady-case grid convergence study confirms the numerical rate of convergence of the method. The extension to problems involving 3D geometries is trivial and requires no coding efforts. Actually, in the nuclear safety context, 3D purely hydraulic numerical simulations are in progress to study the vitrification processes for the storage of radioactive wastes handling problems with moving geometries. In this case, an important issue concerns the treatment of the so-called “freshly fluid or solid cells”.

REFERENCES

[1] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy: An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*, 156 (1999), 209-240.

[2] M.-H. Chung: Cartesian cut cell approach for simulating incompressible flows with rigid bodies of arbitrary shape. *Computers & Fluids*, 35 (2006), 607-623.

[3] R.J. LeVeque, Z. Li: The immersed interface method for elliptic equations with discontinuous coeffcients and singular sources. *SIAM J. Numer. Anal.*, 31:4 (1994), 1019-1044.

[4] I. Ramière, Ph. Angot, M. Belliard: A general fictitious domain method with immersed jumps and non-conforming structured mesh. *Journal of Computational Physics*, 225:2 (2007), 1347-1387.

[5] C. Peskin: Numerical Analysis of blood flow in heart. *Journal of Computational Physics*, 25 (1977), 220-252.

[6] D. Goldstein, R. Handler, L. Sirovich: Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105 (1993), 354-366.

[7] J. Mohd-Yusof: Combined Immersed Boundaries/B-Splines Methods for Simulations of Flows in Complex Geometries. *CTR Annual Research Briefs*, NASA Ames/Stanford University, (1997).

[8] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof: Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161 (2000), 35-60.

[9] A. Gilmanov, F. Sotiropoulos: A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *Journal of Computational Physics*, 207 (2005), 457-492.

[10] N. Peller, A. Le Duc, F. Tremblay, M. Manhart: High-order stable interpolations for immersed boundary methods. *Int. J. Numer. Meth. Fluids*, 52 (2006), 1175-1193.

[11] C.-C. Liao, Y.-W. Chang, C.-A. Lin, J.M. McDonough: Simulating flows with moving rigid boundary using immersed-boundary method. *Computers & Fluids*, 39 (2010), 152-167.

[12] P.H. Chiu, R.K. Lin, T.W.H. Sheu: A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier-stokes equations in time-varying complex geometries. *Journal of Computational Physics*, 229 (2010), 4476-4500.

[13] J.-I. Choi, R.C. Oberoi, J.R. Edwards, J.A. Rosati: An immersed boundary method for complex incompressible flows. *Journal of Computational Physics*, 224 (2007), 757-784.

[14] C. Ji, A. Munjiza, J.J.R. Williams: A novel iterative direct-forcing immersed boundary method and its finite volume applications. *Journal of Computational Physics*, 231 (2012), 1797-1821.

[15] M. Belliard, C. Fournier: Penalized direct forcing and projection schemes for Navier Stokes. *C. R. Acad. Sci. Paris*, Ser. I 348 (2010), 1133-1136.

[16] C. Introni, M. Belliard, C. Fournier: A Second Order Penalized Direct Forcing for Hybrid Cartesian/Immersed Boundary Flow Simulations. *In progress*, (2012).

[17] Ph. Angot, Ch.-H. Bruneau, P. Fabrie: A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81 (1999), 497-520.

[18] A. Sarthou, S. Vincent, J. P. Caltagirone: Consistant velocity-pressure coupling for second-order L^2-penalty and direct-forcing methods. *hal-00592079 version 1*, (2011).

[19] M. Bergmann, A. Iollo: Modeling and simulation of fish-like swimming. *Journal of Computational Physics*, 230 (2011), 329-348.

[20] J. H. Williamson: Low-storage Runge-Kutta schemes. *Journal of Computational Physics*, 35 (1980), 48-56.

[21] A. Chorin: Numerical Solution of the Navier-Stokes Equations. *Mathematics of Computation*, 22 (1968), 745-762.

[22] R. Temam: Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires. *Arch.Rational Mech. Anal.*, 32 (1969), 135-153.

[23] J. Shen: On Error Estimates of some Higher Order Projection and Penalty-Projection Methods for Navier-Stokes Equations. *Numerische Mathematik.*, 62 (1992), 49-73.

[24] M. Jobelin, B. Piar, Ph. Angot, J.-C. Latché: Une méthode de pénalité-projection pour les écoulements dilatables. *European J. Comput. Mech.*, 17 (2008), 453-480.

[25] Th.Gallouët, L. Gastaldo, R. Herbin, J.-C. Latché: An unconditionally stable pressure correction scheme for the compressible barotropic Navier-Stokes equations. *M2AN*, 42 (2008), 303-331.

[26] J.L. Guermond, P.D. Minev: A new class of massively parallel direction splitting for the incompressible Navier Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 200 (2011), 2083-2093.

[27] J.L. Guermond, P.D. Minev, J. Shen: An overview of projection methods for incompressible flows. *Comput. Methods Appl. Mech. Engrg.*, 195 (2006), 6011-6045.

[28] S.O. Unverdi, G. Tryggvason: A Front-Tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100 (1992), 25-37.

[29] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan: A Front-Tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169 (2001), 708-759.

[30] T. Ikeno, T. Kajishima: Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations. *Journal of Computational Physics*, 226 (2007), 1485-1508.

[31] C. Calvin: A development framework for parallel CFD applications: Trio_U project. *SuperComputing in Nuclear Applications (SNA-2003)*, Paris, France (2003). http://www-trio-u.cea.fr

[32] 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. *Journal of Computational Physics*, 229 (2010), 1043-1076.

[33] E. Guyon, J.P. Hulin, L. Petit: Physical Hydrodynamics. Oxford University Press, 2001.

[34] D. Stojković, M. Breuer, F. Durst: Effect of high rotation rates on the laminar flow around a circular cylinder. *Physics of Fluids*, 14:9 (2002), 3160-3178.

[35] K. Taira, T. Colonius: The immersed boundary method: a projection approach. *Journal of Computational Physics*, 225 (2007), 2118-2137.

[36] M.N. Linnick, H.F. Fasel: A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics*, 204 (2005), 157-192.

[37] B. Fornberg: A numerical study of steady viscous flow past a circular cylinder. *Journal of Fluids Mechanic*, 98:4 (1980), 819-855.

[38] D.J. Tritton: Experiments on the flow past a circular cylinder at low Reynolds number. *Journal of Fluids Mechanic*, 6 (1959), 547-567.

[39] D.B. Ingham, T. Tang: A numerical investigation into steady flow past a rotating circular cylinder at low and intermediate Reynolds numbers. *Journal of Computational Physics*, 87 (1990), 91-107.

[40] H. M. Badr, S. C. R. Dennis, P. J. S. Young: Steady and unsteady flow past a rotating cylinder at low Reynolds numbers. *Computers & Fluids*, 17:4 (1989), 579-609.

[41] S. Kang, H. Choi, S. Lee: Laminar flow past a rotating circular cylinder. *Phys. Fluids*, 11:11 (1999), 3312-3321.