Commun. Comput. Phys.
doi: 10.4208/cicp.220313.111013a

Vol. 15, No. 5, pp. 1266-1290
May 2014

An Accurate Cartesian Method for Incompressible Flows with Moving Boundaries

M. Bergmann*, J. Hovnanian and A. Iollo

Inria, F-33400 Talence, France, Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France, CNRS, IMB, UMR 5251, F-33400 Talence, France.

Received 22 March 2013; Accepted (in revised version) 11 October 2013

Communicated by Boo-Cheong Khoo

Available online 14 February 2014

Abstract. An accurate cartesian method is devised to simulate incompressible viscous flows past an arbitrary moving body. The Navier-Stokes equations are spatially discretized onto a fixed Cartesian mesh. The body is taken into account via the ghost-cell method and the so-called penalty method, resulting in second-order accuracy in velocity. The accuracy and the efficiency of the solver are tested through two-dimensional reference simulations. To show the versatility of this scheme we simulate a three-dimensional self propelled jellyfish prototype.

AMS subject classifications: 68Uxx, 76

Key words: Viscous incompressible flow, immersed boundary method, penalty method, cartesian grid, self-propelled jellyfish.

1 Introduction

Over the last decades, many works have been developed in order to precisely describe the interactions between fluids and structures, especially when the solid presents complex boundaries. These studies can be divided in two categories. The first category is represented by Arbitrary Lagrangian Eulerian methods [10]. These methods are accurate but are hard to set up and complicated to implement, moreover when moving and/or deforming obstacles are considered. In addition, a dynamic mesh partitioner is necessary when one deals with parallel computations. The second category is represented by interface methods that are usually more versatile. Among them, there are the Ghost Fluid method [12] or the Immersed Boundary method [24]. The current work focuses on the immersed boundary method on a cartesian grid.

*Corresponding author. *Email addresses:* michel.bergmann@inria.fr (M. Bergmann), jesshovna@msn.com (J. Hovnanian), angelo.iollo@math.u-bordeaux1.fr (A. Iollo)

The immersed boundary method has initially been introduced by Peskin in 1972 [29]. In his original version, the interactions between fluid and solid are described through a forcing term consisting in a Dirac function located at the interface. This term is added to the Navier-Stokes equations before discretization and thus is completely independent of the spacial discretization. This approach is defined as a *continuous forcing* method and was used by Peskin to simulate blood flow in a beating heart but has also been implemented to simulate multiphase flow [36, 38] and solidification [40]. In another class of immersed boundary methods one modifies the space discretization of the Navier-Stokes equations near the interface. They are called *discrete forcing* methods. This method was used and developed by Liu et al. [19], Marella et al. [21], Mittal et al. [11,23], Li et al. [17], among others. Despite being strongly dependent on the space discretization, this second category has the advantage of being sharp, as opposed to the first one.

The Ghost Cell method is inspired by this last approach since ghost-cell values are introduced based on an interpolation of neighboring fluid cells, such that the boundary conditions are satisfied on the immersed interface.

In this paper, the resolution of the Navier-Stokes equations follows a classical predictor-corrector scheme [5, 37] with second-order spatial accuracy near the immersed boundary. Other schemes have recently been proposed to achieve higher-order accuracy for simulating flows with moving boundaries on cartesian meshes, see for example [11, 18, 21, 23]. In those works, the main idea is to improve accuracy by an efficient and simple reconstruction of the solution near the immersed boundary. The schemes then usually differ near the immersed boundary in the specific solution reconstruction and in the treatment of mesh nodes that present geometric ambiguities with respect to the discretization stencil. In the same spirit, here we propose a local reconstruction of the solution near the immersed boundary that is based on the geometric information delivered by the distance function [26, 32, 33]. Moreover, as explained in the following, a penalty method [1] is employed in the prediction step, allowing a consistent and efficient solution of the pressure equation on the whole domain. Also, as it is shown hereafter, the penalty approach automatically solves the problem of fixing appropriate physically meaningful conditions for the ”fresh” nodes crossing the immersed boundary when the border is moving.

The organization of this paper is as follows: the Section 2 is devoted to the description of the numerical methodology including the scheme used at the immersed boundary. In Section 3, we present validations through computed results for several canonical tests and establish the accuracy of the scheme. The present method is then used to simulate the flow generated by a self-propelled jellyfish-like swimmer. Finally, the Section 4 is dedicated to conclusions.

2 Discretization of the governing equations

We call Ω_f the fluid domain, surrounding a body called Ω_s. The entire domain is

$$\Omega=\Omega_s\cup\Omega_f$$

and the interface fluid-solid is $\Gamma=\partial\Omega_s$ (see Fig. 1).

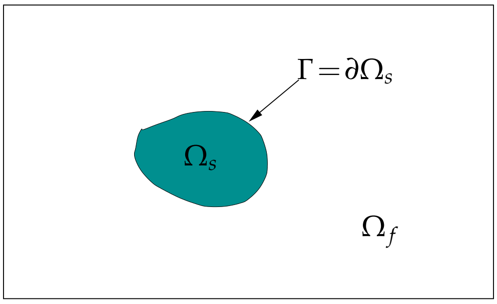


Figure 1: Sketch of a generic flow configuration.

2.1 The governing equations in the fluid

Under the hypothesis of viscous and incompressible flow, the governing equations considered are the incompressible Navier-Stokes equations:

$$\begin{aligned}
&\rho\left(\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right)=-\nabla p+\mu\Delta\boldsymbol{u} && \text{in } \Omega_f, && (2.1a)\\
&\nabla\cdot\boldsymbol{u}=0 && \text{in } \Omega_f, && (2.1b)\\
&\boldsymbol{u}=\boldsymbol{u}_B(\boldsymbol{x},t) && \text{in } \Omega_s, && (2.1c)\\
&\boldsymbol{u}(t=0,\boldsymbol{x})=\boldsymbol{u}_0(\boldsymbol{x}) && \text{in } \Omega_f, && (2.1d)\\
&\boldsymbol{u}(t,\boldsymbol{x})=\boldsymbol{u}_\Gamma(t) && \text{on } \Gamma, && (2.1e)
\end{aligned}$$

with ρ the fluid density, $\boldsymbol{u}=(u,v)$ the velocity, p the pressure, μ the dynamic viscosity and $\boldsymbol{u}_B$ is the body velocity that can be either imposed or resulting from the interaction between fluid and the body. The fields $\boldsymbol{u}_0$ and $\boldsymbol{u}_\Gamma$ represent the initial and boundary conditions, respectively. In some cases it is more convenient to work with the non-dimensional Navier-Stokes equations, where (2.1a) becomes:

$$\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}=-\nabla p+\frac{1}{Re}\Delta\boldsymbol{u} \quad \text{in } \Omega_f, \tag{2.2}$$

with $Re=\rho U_\infty D/\mu$ the Reynolds number.

We consider the whole system (fluid-solid) as a single flow using the so-called penalty method [1]. This approach considers the whole system as a porous media with a variable permeability K, where the impermeable body is consider as a fluid with very low permeability $K\ll 1$.

Indeed, in the limit of $K \to 0$, it is demonstrated in [1] that

$$\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}=-\nabla p+\frac{1}{Re}\Delta\boldsymbol{u}+\frac{\chi_B}{K}(\boldsymbol{u}_B-\boldsymbol{u}(\boldsymbol{x},t)) \qquad \text{in } \Omega, \tag{2.3a}$$

$$\nabla\cdot\boldsymbol{u}=0 \qquad \text{in } \Omega, \tag{2.3b}$$

tends to Eqs. (2.2), (2.1b) and (2.1c), with χ_B the characteristic function of the solid, defined as:

$$\chi_B=\begin{cases} 1, & \text{in the solid,} \\ 0, & \text{elsewhere.} \end{cases} \tag{2.4}$$

The presence of the body is explicitly described through the penalization term in Eq. (2.3a). For a moving and/or deforming body, the characteristic function χ_B has to be updated following a transport equation:

$$\frac{\partial \chi_B}{\partial t}+(\boldsymbol{u}\cdot\nabla)\chi_B=0 \quad \text{in } \Omega. \tag{2.5}$$

To this end, a level set function ϕ can be introduced. This method, first presented by Osher and Sethian [26, 27], implicitly represents the fluid/solid interface in the domain. The level set function is the signed distance from the boundary of the body with negative sign within the solid and positive one elsewhere. The interface between the fluid and the immersed body is thus defined by $\phi=0$. The link between the level set and the characteristic function is $\chi_B=1-H(\phi)$, where H represents the Heaviside function. Finally, the level set function satisfies the same transport equation, i.e.,

$$\frac{\partial \phi}{\partial t}+(\boldsymbol{u}\cdot\nabla)\phi=0 \quad \text{in } \Omega. \tag{2.6}$$

The outward normal vector of the body boundary, $\boldsymbol{n}$, is simply defined as:

$$\boldsymbol{n}=\frac{\nabla\phi}{|\nabla\phi|}. \tag{2.7}$$

The property of distance characteristic is not necessary conserved. The level-set function must be re-computed from the isoline zero, for instance using fast marching method [31, 32].

2.2 Numerical method

The Navier-Stokes equations are discretized using a finite-difference scheme with the collocated cell-centered primitive variables $(\boldsymbol{u},p)$. We also introduce a face-centered velocity $\boldsymbol{U}$ as presented by Mittal et al. [23] to improve the pressure smoothness. A predictor-corrector fractional step method is used to evaluate the solution in time. Based on the

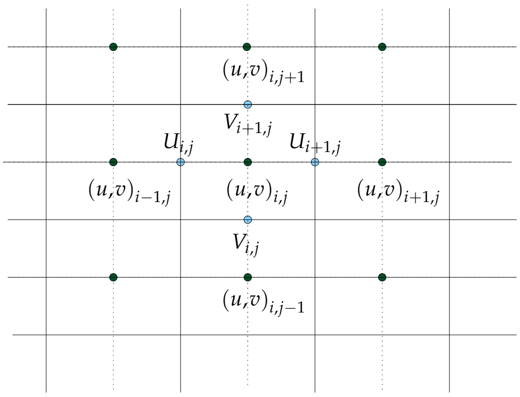


Figure 2: Sketch of velocity location.

Chorin-Temam scheme [5, 37], the first step evaluates an intermediate velocity $\boldsymbol{u}^*$ obtained by the resolution of an unsteady advection diffusion equation, consisting in the prediction step:

$$\frac{\boldsymbol{u}^*-\boldsymbol{u}^n}{\Delta t}=-\frac{1}{2}(3\boldsymbol{N}^n-\boldsymbol{N}^{n-1})-\nabla p^n+\frac{1}{2Re}(\boldsymbol{D}^n+\boldsymbol{D}^*)+\frac{\chi_B}{K}(\boldsymbol{u}_B-\boldsymbol{u}^*) \quad \text{in } \Omega. \tag{2.8}$$

The convective and diffusive terms are

$$\boldsymbol{N}=\Big(\frac{\delta(U_j u_i)}{\delta x_j}\Big)_i \quad \text{and} \quad \boldsymbol{D}=\Big(\frac{\delta^2 u_j}{\delta^2 x_j}\Big)_i,$$

respectively. The discretization operator $\delta/\delta x_j$ represents a second order central finite difference. A second order Adams-Bashforth scheme is used for the convective term and a second order Crank-Nicholson scheme the diffusive term. The velocity fields $\boldsymbol{u}^*$ is not divergence free. This predicted field $\boldsymbol{u}^*$ is then projected in a divergence free space. Taking the divergence of

$$\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^*}{\Delta t}=-(\nabla p^{n+1}-\nabla p^n) \quad \text{in } \Omega, \tag{2.9}$$

and imposing

$$\nabla\cdot\boldsymbol{u}^{n+1}=0 \quad \text{in } \Omega, \tag{2.10}$$

we obtain:

$$\nabla\cdot\boldsymbol{u}^*=\Delta t\nabla^2(p^{n+1}-p^n) \quad \text{in } \Omega. \tag{2.11}$$

At this point, we introduce the face centered velocity $\boldsymbol{U}$ (Fig. 2), computed by interpolating the cell centered neighboring velocities.

The face centered velocity is defined as follows in Ω:

$$\begin{cases} \tilde{\boldsymbol{u}}=\boldsymbol{u}^*+\Delta t(\nabla p^n)_{cc}, \\ \tilde{\boldsymbol{U}}=\mathcal{F}(\tilde{\boldsymbol{u}}), \\ \boldsymbol{U}=\tilde{\boldsymbol{U}}-\Delta t(\nabla p^n)_{fc}, \end{cases} \tag{2.12}$$

where the index cc and fc mean respectively cell centered and face centered, and $\mathcal{F}$ is an interpolation function.

Let us note $\psi = \Delta t(p^{n+1} - p^n)$. Hence, the Poisson equation that has to be solved in Ω is:

$$\nabla \cdot \boldsymbol{u}^* = \Delta\psi. \tag{2.13}$$

This Poisson equation is solved with homogeneous Neumann boundary conditions on $\partial\Omega$. These conditions ensure that the normal velocity is not perturbed on the boundaries of the domain during the correction step. Besides, using the penalty approach, there is no need to impose boundary conditions on the fluid-structure interface.

Pressure oscillations may occur if one uses the collocated cell centered velocity to compute the divergence. To suppress these oscillations, we use the face-centered velocity $\boldsymbol{U}$, corresponding to the face-centered velocity. Eq. (2.11) becomes:

$$\nabla \cdot \boldsymbol{U} = \Delta\psi. \tag{2.14}$$

The velocities and the pressure are then updated in Ω:

$$p^{n+1} = p^n + \frac{\psi}{\Delta t}, \tag{2.15a}$$

$$\boldsymbol{u}^{\boldsymbol{n+1}} = \boldsymbol{u}^* - (\nabla\psi)_{cc}, \tag{2.15b}$$

$$\boldsymbol{U}^{\boldsymbol{n+1}} = \tilde{\boldsymbol{U}} - (\nabla\psi)_{fc}. \tag{2.15c}$$

We arbitrarily choose an initial pressure value $p^0 = 0.5$. Depending on the applications, one can be interested in moving and/or deforming the body. For instance, for a self-propelled body like the jellyfish (Subsection 3.4), a deformation velocity can be imposed to the body resulting in translation and rotation velocities computed from the forces and the torques. The body velocity (deformation + translation + rotation) is $\boldsymbol{u}_B$. The body is then displaced following (2.6). The new fluid/structure interface is the zero level of that function.

The interface fluid/solid does not *a priori* fit the fluid mesh. In the classical approach of the penalty method, the penalized velocity is forced on all the grid nodes inside the bodies, with a first order accuracy in space. Here, we present the *Image Point Correction* method (IPC) which consists in a correction of the value of the penalized velocity at the solid points having at least one fluid neighbor using a ghost-cell approach in order to impose the desired velocity on the interface. This improvement step is performed before the prediction step (2.9) to impose accurate gradients near the interface. The main idea is to find the right correction in the normal direction of the interface, using the velocity gradient through the interface. In other words, considering a one-dimensional case, let us denote $\boldsymbol{u}_B$ the velocity we want on the boundary, $\boldsymbol{u}_S$ and $\boldsymbol{u}_F$, respectively, the penalized velocity enforced on the solid node situated in x_S, and the velocity in the fluid node situated in $x_F = x_S - \Delta x$, both neighboring the interface, like in Fig. 3. The value of the penalized velocity $\boldsymbol{u}_B$ must be such that the velocity gradient taken at the interface remains

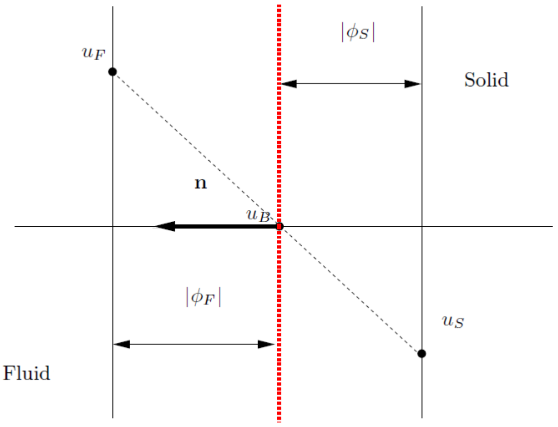


Figure 3: One-dimensional sketch of the methodology used to compute the corrected velocity.

the same when it is evaluated through the interface (see Fig. 3). Mathematically it means:

$$\frac{\boldsymbol{u}_S-\boldsymbol{u}_B}{|\phi_S|}=\frac{\boldsymbol{u}_B-\boldsymbol{u}_F}{|\phi_F|}, \tag{2.16}$$

where $|\phi_S|$ and $|\phi_F|$ are the absolute value of the level set function in $\boldsymbol{x}_S$ and $\boldsymbol{x}_F$ respectively, that is to say their distances from the interface. In other words,

$$\boldsymbol{u}_S=\boldsymbol{u}_B+|\phi_S|\left(\frac{\boldsymbol{u}_B-\boldsymbol{u}_F}{|\phi_F|}\right), \tag{2.17}$$

which allows us to exhibit the general expression of the corrected velocity:

$$\boldsymbol{u}_S=\boldsymbol{u}_B-|\phi|\left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right)\bigg|_{\phi=0}, \tag{2.18}$$

with $\boldsymbol{n}$ the outward normal unit vector of the interface. Finally, as ϕ is negative in the solid, the final form of the corrected velocity is:

$$\boldsymbol{u}_S=\boldsymbol{u}_B+\phi\left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}}\right)\bigg|_{\phi=0}. \tag{2.19}$$

The IPC method is applied to the solid points having a neighbor in the fluid. The penalization method is applied to all other solid points.

Now let us consider a two-dimensional case (see Fig. 4). In order to determine $(\partial \boldsymbol{u}/\partial \boldsymbol{n})|_{\phi=0}$, we must find all the symmetric points of the ghost points through the interface. This can easily be done thanks to the level set.

Indeed, as previously seen, the external outward unit vector of the interface can be computed based on the knowledge of the level set function. Besides, having the direction, we know that the interface point corresponding to the ghost node is located at a distance

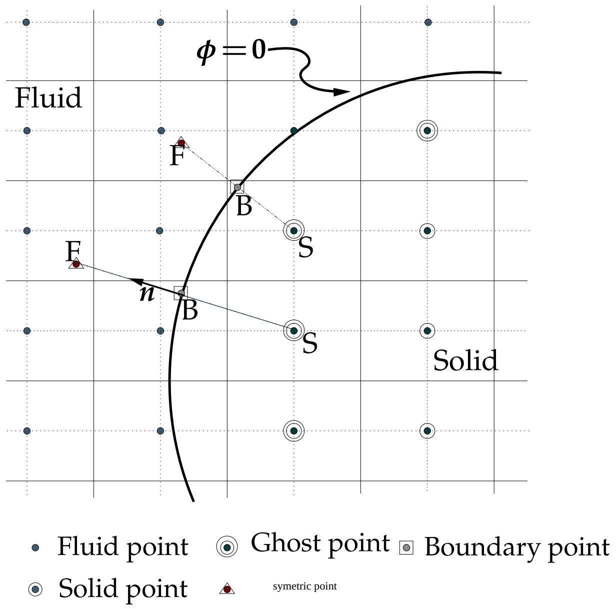


Figure 4: Sketch of the IPC method in a two-dimensional case. The IPC is applied to the Ghost points using the symmetric points. The value of the symmetric point is based on an interpolation of its four neighboring fluid points. The original penalty method is applied on the other solid points.

$|\phi|$ from the ghost node, so the symmetric point is at $2|\phi|$. The symmetric point of the solid point to be corrected is located in the fluid and does not *a priori* fit with a grid node. Consequently, its value must be interpolated. Usually it can be interpolated from its four closest neighbors (located at the extremities of the cell containing the symmetric point). Interpolation can directly be computed from those four points except if one of those point is precisely the point to be corrected. In this last case, this point is replaced by its closest point on the body boundary where the velocity is known, see Fig. 5. This correction is similar to that used in [23]. In this study a second order Lagrange interpolation is performed.

The penalization step in (2.9) is used to bypass two difficulties. Firstly, it has been proven (Remark 3.1, [1]) that the use of the penalty method leads to a Neumann condition for the pressure at the fluid/structure interface. This allows the Poisson solver to take full advantage of the data structure resulting from a uniform grid, leading to a simple parallel code and an efficient distributed solver. Using other approaches, boundary conditions should explicitly be applied on the fluid/structure interface (Poisson equation (2.13)). Secondly, when the body is moving, fresh cells are created. This means that some body cells at a given time step can become fluid cell at the next time step. Without penalty method, these fresh cells have the velocity that was just modified using the IPC correction. Clearly, these velocities are unphysical. The penalty step imposes divergence free fresh cells if the body motion and deformation is divergence free. In conclusion, the overall scheme is formally second-order accurate near the immersed boundary thanks to

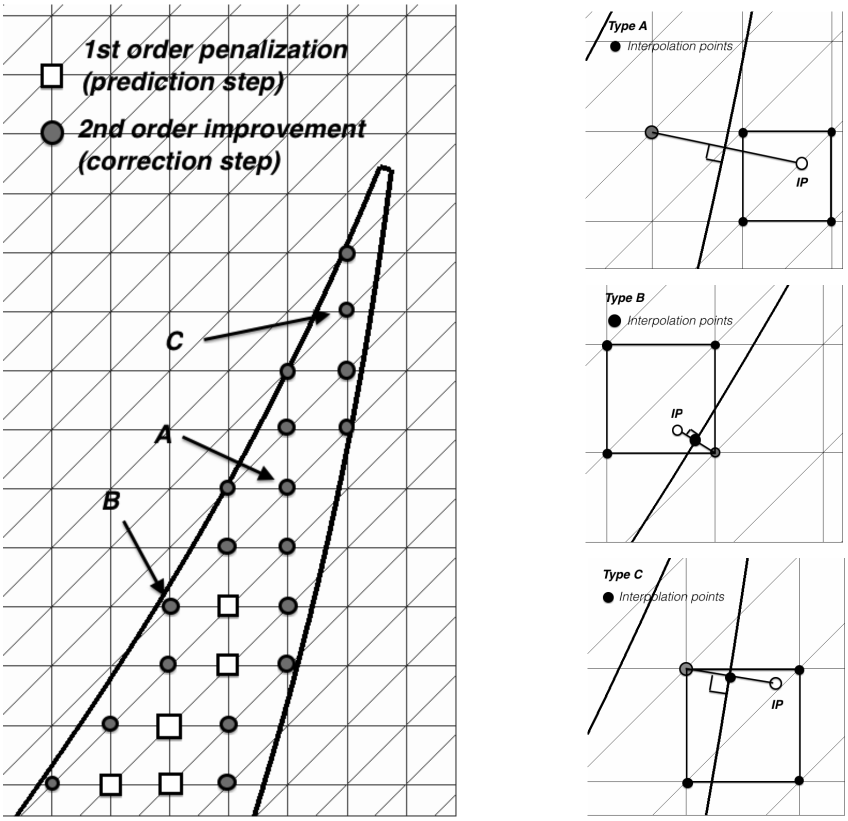


Figure 5: For simplicity of explication, we sketch a 2D example relative to an axial section of a 3D jellyfish (Subsection 3.4). 2D interpolation is performed using four points. Usually, the four corners of the cell containing the Image Point (IP) are used for interpolation. Several kinds of configurations can occur for velocity correction. For example type A (four fluid points), type B (three fluid points) and type C (two fluid points). In the case where the point to be corrected corresponds to one of the corners, this points is replaced with the nearest point on the boundary (types B and C). No special treatment is performed for the other points belonging to the moving body, like in type C. When a point does not belong neither to A, B or C type, it is penalized with a first order correction. The number of these points is however negligible. This approach is similarly extended in 3D over a cube.

the penalty method and the distance function. No boundary conditions for pressure are imposed on the fluid/structure interface. Oscillations in pressure are prevented as done in [16, 30]. The scheme is consistent and therefore mass conservation is granted in the limit of the approximation accuracy, like in any other projection scheme.

3 Numerical results

In this section, we first evaluate the order of accuracy of the solver (Subsection 3.1). Then we test it on several reference cases. We will consider both steady (Subsection 3.2) and moving cylinders (Subsection 3.3). We will compare the aerodynamic coefficients to reference results for several values of the Reynolds number. While the steady cylinder is defined by $\boldsymbol{u}_B = \mathbf{0}$ and $\boldsymbol{U}_\infty \neq \mathbf{0}$, the moving cylinder is defined by $\boldsymbol{u}_B \neq \mathbf{0}$ and $\boldsymbol{U}_\infty = \mathbf{0}$. We will thus define the Reynolds number as $Re = \|\boldsymbol{U}_\infty - \boldsymbol{u}_B\|_2 D/\nu$. Finally, we will model and simulate a three-dimensional jellyfish (Subsection 3.4).

The forces exerted by the fluid on the body can be computed by:

$$\boldsymbol{F} = \oint_{\partial\Omega_S} \boldsymbol{n}\Big[-pI + \frac{1}{Re}\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right)\Big]ds, \tag{3.1}$$

where $\partial\Omega_S$ is the boundary of the body and $\boldsymbol{n}$ is the external unit normal vector of the body boundary. The aerodynamic coefficient are $\boldsymbol{C} = \boldsymbol{F}/(0.5\rho_0 U_\infty^2 0.5D)$. The drag and lift coefficients are respectively defined as $C_D = \boldsymbol{C}\cdot\boldsymbol{e}_x$ and $C_L = \boldsymbol{C}\cdot\boldsymbol{e}_y$. In order to evaluate the pressure and the velocity gradients on the interface we use a $\mathbb{Q}_1$ interpolation on the pressure and the velocity. The $\mathbb{Q}_1$ interpolation consists in approximating the flow variable φ (here the pressure and the velocity) by a quadratic interpolation. The body is described by Lagrangian markers, and we then identify the four closest grid nodes to each marker. Then, the flow variable value is determined on this marker thanks to the formulation:

$$\varphi(x,y) = c_1 xy + c_2 x + c_3 y + c_4, \tag{3.2}$$

with (x,y) the position of the current marker. The four coefficients c_i are determined thanks to the values of φ on the four closest neighbors, previously identified:

$$\{\underline{C}\} = [\mathcal{V}]^{-1}\{\underline{\varphi}\}, \tag{3.3}$$

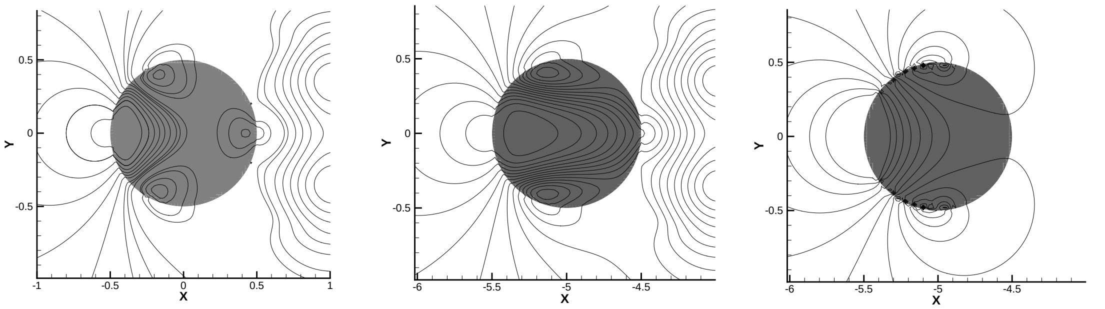
where $\{\underline{C}\}^T = (c_1,\cdots,c_4)$ is the vector containing the four unknown coefficients, $\{\underline{\varphi}\}^T = (\varphi_1,\cdots,\varphi_4)$ is the vector containing the values of the flow variable on the four grid nodes, $[\mathcal{V}]$ is the Vandermonde Matrix. As a consequence, $[\mathcal{V}]$ can be written as:

$$[\mathcal{V}] = \begin{pmatrix} xy|_1 & x|_1 & y|_1 & 1 \\ xy|_2 & x|_2 & y|_2 & 1 \\ xy|_3 & x|_3 & y|_3 & 1 \\ xy|_4 & x|_4 & y|_4 & 1 \end{pmatrix}.$$

Thus, the pressure can be determined and, differentiating this last expression by x and y, we can determine the velocity gradient.

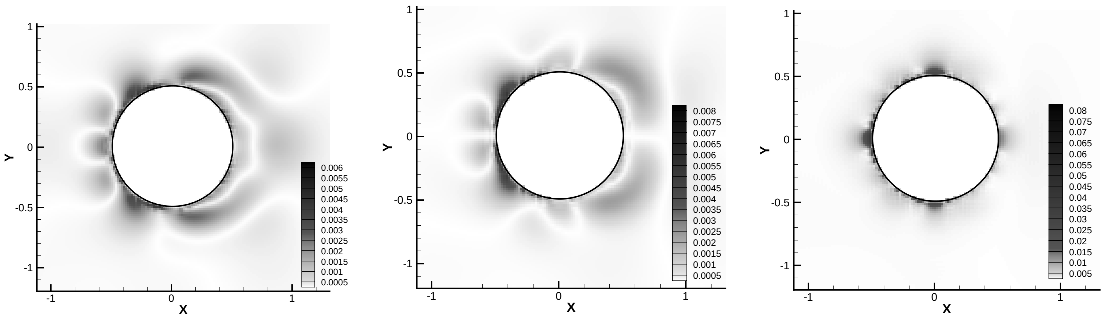
3.1 Grid convergence study

Here we test the solver for a two dimensional flow past a cylinder at $Re = 45$. The size of the domain is $6D \times 4D$ with D the diameter of the cylinder. Its center is located in $(0,0)$. We used Dirichlet boundary at the inflow ($\boldsymbol{u} = \boldsymbol{U}_\infty$), non-reflecting boundary conditions at the outflow [14] and streamline boundary conditions ($v = 0$ and $\partial u/\partial y = 0$) on others boundaries. The boundary conditions for the pressure are also similar to those developed in [14]. A grid convergence evaluation consists in comparing the errors between several



(a) steady cylinder (b) moving cylinder with penalty step (c) moving cylinder without penalty step

Figure 6: Distribution of pressure contours at $t=5$ for $Re=550$.

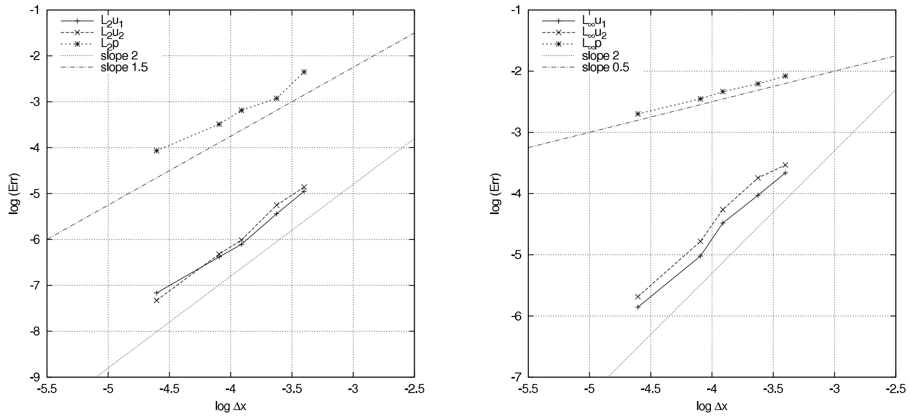


(a) u (b) v (c) p

Figure 7: Error distribution for the numerical solution for the steady cylinder with (360×240) grid.

grids and an analytical solution. Nevertheless, as there is no existing analytical solution for this case, a refined solution is considered as the analytical one, corresponding to a grid (1800×1200). This highly resolved solution is compared to five other solutions corresponding to the following grids: (600×400), (360×240), (300×200), (225×150) and (180×120). The time step chosen is $10^{-5}D/U_\infty$, thus respecting the CFL condition for every grid, and the different solutions are compared after a time of $t=0.5$, i.e., 5×10^4 iterations. The distribution of the error magnitude in velocity and pressure for the steady cylinder using grid (360×240) can be seen in Fig. 7. These figures show that the error is mainly concentrated around the immersed boundary. Fig. 8 shows that we globally obtain a second-order accuracy, measured with the L_2-norm, for the velocity and order 1.5 for the pressure. Concerning the local accuracy evaluated using the L_∞-norm, we have a second order too on the velocity and order 0.5 for the pressure.

As already mentioned in the previous section, we can see in Fig. 6 that the classical penalty step applies the right pressure conditions on the interface, without any special treatment.



(a) L_2

(b) L_∞

Figure 8: L_2 and L_∞ norms of the error for the u and v velocity components and the pressure, versus the computational grid size.

3.2 Steady cylinder

In this section we consider steady circular cylinders ($\boldsymbol{u}_B=0$) with an inflow velocity $U_\infty=1$. We will first study the flow past a single cylinder (Subsection 3.2.1) and then we will study the flow past two cylinder in a tandem arrangement (Subsection 3.2.2).

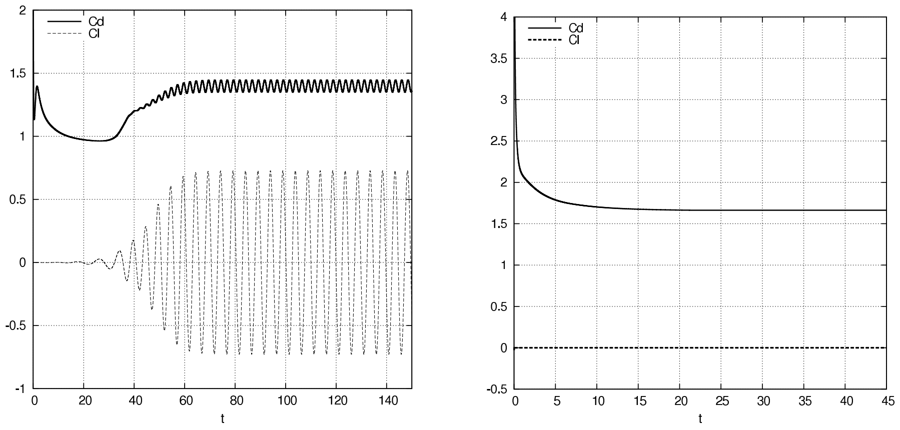
3.2.1 Flow past a steady cylinder

We have performed simulations for Reynolds numbers in a range $40-550$. The domain size is $[-8;16]\times[-8;8]$. The space step is $\Delta x=0.01$ and the time step is chosen to respect the CFL conditions.

We first consider the asymptotic regime (long time integration) at $Re=40$, $Re=100$, $Re=200$ and $Re=300$. The temporal evolutions of the drag and lift coefficients at $Re=40$ and $Re=200$ is presented in Fig. 9. While the flow at $Re=40$ is steady, the flow at $Re=200$ is unsteady (drag and lift fluctuations). Comparisons with references results in presented in Table 1. Our results show good agreements with previous studies.

Table 1: Comparison of the mean drag coefficient for one steady cylinder, with previous studies.

Re	40	100	200	300
Present study	1.66	1.41	1.40	1.42
Wieselsberger [39]	1.70	-	-	1.22
Mittal [23]	1.53	1.35	-	1.36
Hendersen [13]	1.54	1.35	1.34	1.37
Braza [4]	-	1.36	1.40	-
Bergmann [2]	1.68	1.41	1.39	-



(a) $Re = 200$ (b) $Re = 40$

Figure 9: Drag and Lift coefficients for a steady cylinder.

We then consider a short time integration at $Re = 550$. The temporal evolution of the drag is shown in Fig. 10. These results also present good agreement between the present study and the numerical results in [28].

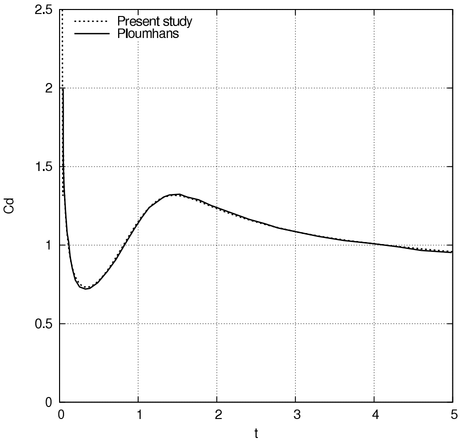


Figure 10: Temporal evolution of the drag coefficient for a steady cylinder at $Re = 550$. Comparison with results obtained by [28] for a fixed cylinder.

3.2.2 Flow past two steady cylinders in a tandem arrangement

After validating the method for a single steady cylinder, we now consider two steady cylinders in a tandem arrangement. In other words, let us consider two circular cylinders $\mathcal{C}_1$ and $\mathcal{C}_2$, with the same diameter D, at a distance L from each other. The size of the domain is $40D \times 20D$. The two solids are vertically centered in the domain and the cylinder $\mathcal{C}_1$ is located at $10D$ from the upstream boundary (see Fig. 11). The space step is

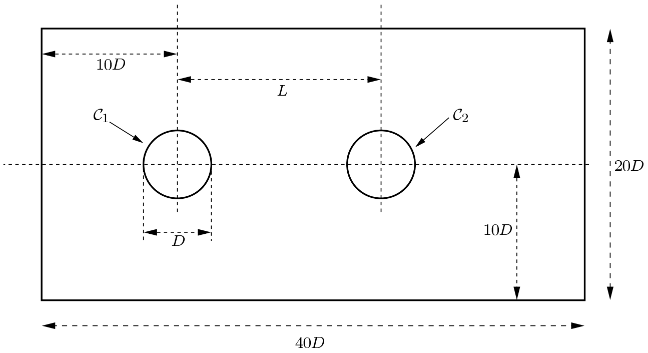


Figure 11: Sketch of the two steady cylinders in a tandem arrangement.

chosen to be $\Delta x = \Delta y = 1.6\times 10^{-2}$ and the time step is $\Delta t = 10^{-3}$. Then, we compare the mean drag coefficient and the Strouhal number, for several values of Re, with previous studies. The results are gathered in Tables 2 to 4 and show good agreements with the previous studies.

Table 2: Comparison of the mean drag coefficient and the Strouhal number for two steady cylinders in a tandem arrangement with previous studies for $Re=100$ and $L=2.5D$ or $L=5.5D$.

$Re=100$								
	$L=2.5D$				$L=5.5D$			
	C_d		St		C_d		St	
	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$
Present study	1.16	−0.057	0.12	0.12	1.31	0.90	0.167	0.167
Dehkordi [8]	1.24	−0.071	–	–	1.33	1.00	0.165	0.165
Mittal [25]	1.27	−0.075	–	–	1.43	0.86	0.168	0.168
Ding [9]	1.16	−0.09	–	–	1.33	0.95	0.160	0.160
Sharman [34]	1.17	−0.062	0.118	–	1.25	0.65	0.154	0.154

Table 3: Comparison of the mean drag coefficient and the Strouhal number for two steady cylinders in a tandem arrangement with previous studies for $Re=200$ and $L=1.5D$ or $L=2D$.

$Re=200$								
	$L=1.5D$				$L=2D$			
	C_d		St		C_d		St	
	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$
Present study	1.04	−0.17	0.185	0.185	1.00	−0.19	0.140	0.140
Dehkordi [8]	1.05	−0.15	0.175	0.175	1.03	−0.16	0.138	0.138
Meneghini [22]	1.06	−0.18	0.167	0.167	1.03	−0.17	0.130	0.130
Mahir [20]	–	–	–	–	1.06	−0.21	–	–
Slaouti [35]	–	–	–	–	0.89	−0.21	0.130	0.130

Table 4: Comparison of the mean drag coefficient and the Strouhal number for two steady cylinders in a tandem arrangement with previous studies for $Re = 200$ and $L = 3D$ or $L = 4D$.

	$Re=200$							
	$L=3D$				$L=4D$			
	C_d		St		C_d		St	
	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_2$
Present study	0.96	-0.18	0.126	0.126	1.23	0.89	0.185	0.185
Dehkordi [8]	1.00	-0.08	0.129	0.129	1.16	0.52	0.18	0.18
Meneghini [22]	1.00	-0.08	0.125	0.125	1.18	0.38	0.17	0.17
Mahir [20]	1.05	-0.56	0.130	0.130	1.34	0.56	0.18	0.18
Slaouti [35]	0.87	-0.16	–	–	1.11	0.88	0.19	0.19

3.3 Moving cylinder

We now consider moving cylinders ($\boldsymbol{u}_B \neq \mathbf{0}$ and $\boldsymbol{U}_\infty = \mathbf{0}$). We consider an impulsively started cylinder (Subsection 3.3.1) and an impulsively started then stopped cylinder (3.3.2). Finally the sedimentation of a cylinder under gravity is studied (Subsection 3.3.3).

3.3.1 An impulsively started cylinder

We consider an impulsively started cylinder at $Re = 550$. This test case is similar to that presented in Subsection 3.2.1, except that we now have $u_B = 1$ and $\boldsymbol{U}_\infty = \mathbf{0}$. As it is shown in Fig. 12, our method shows good results. Indeed, the global behavior of the two curves is very similar although the curve of the impulsively started cylinder shows some oscillations but these are decreasing when $\Delta x \to 0$. This is explained by the fact that, due to the motion of the cylinder, the number of points in the fluid at time t^n is not the same as at t^{n+1}.

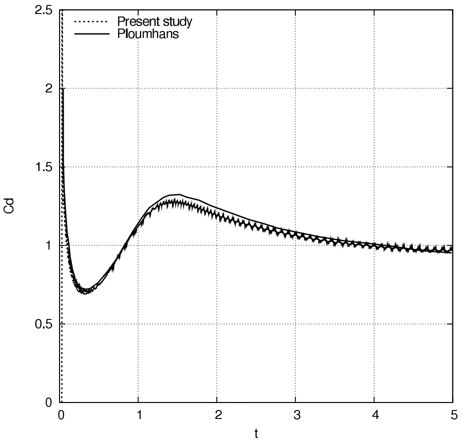


Figure 12: Temporal evolution of the drag coefficients for an impulsively started cylinder at Re=550. Comparison with results obtained by [28] for a fixed cylinder.

3.3.2 Impulsively started then stopped cylinder

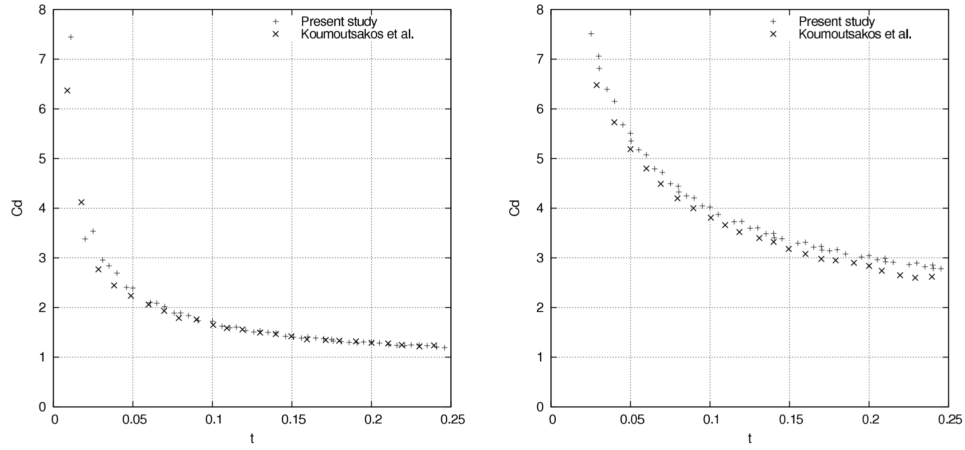
We also performed two-dimensional simulations of impulsively started cylinder that is impulsively stopped for several Reynolds numbers. These tests are presented in [15]. Let us note that in [15], time is non-dimensionalized using the radius of the cylinder:

$$T_0=\frac{\frac{1}{2}D}{U_\infty},$$

whereas we use the diameter of the cylinder: $t_0=D/U_\infty$, which explains the factor two between their non-dimensionalized times and ours. The following comparisons will be adapted to our non-dimensional time.

First of all, we compared drag coefficients for an impulsively started cylinder for $Re=40$ and $Re=200$, up to a time $t=0.25$ (Fig. 13) and observe that our results are in good agreements with [15]. Then, we considered an impulsively started cylinder for $Re=40$ that is stopped at time $t=5$. Fig. 14 shows a comparison of the evolutions of the drag and Fig. 15 represents the vorticity field computed with the present method which is in good agreement with the results in [15].

Finally, we perform a simulation at $Re=550$. An impulsively started cylinder is stopped at time $t=2.5$. Figs. 16 and 17 represent respectively the comparison of the drag evolutions and the vorticity field. All these results match those of Koumoutsakos et al. [15].

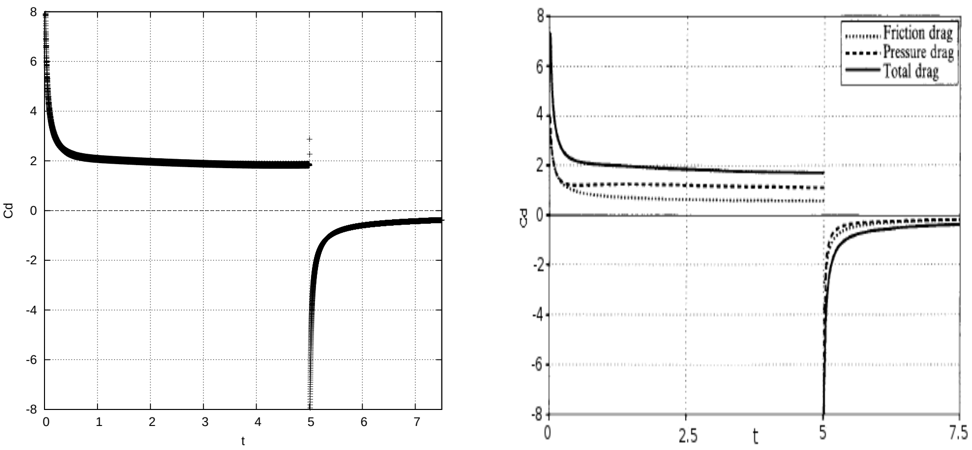


(a) $Re=200$ (b) $Re=40$

Figure 13: Comparison of the drag coefficients for an impulsively started cylinder up to a time $t=0.25$ with [15].

3.3.3 Sedimentation of cylinder

We study the sedimentation of a two-dimensional cylinder in a cavity. We consider the same configuration as Coquerelle and Cottet in [6]. The domain under consideration is



(a) Present study

(b) Koumoutsakos et al. [15]

Figure 14: Comparison of the Drag coefficients for an impulsively started cylinder stopped at $t=5$ for $Re=40$ (using our non-dimensionalized time).

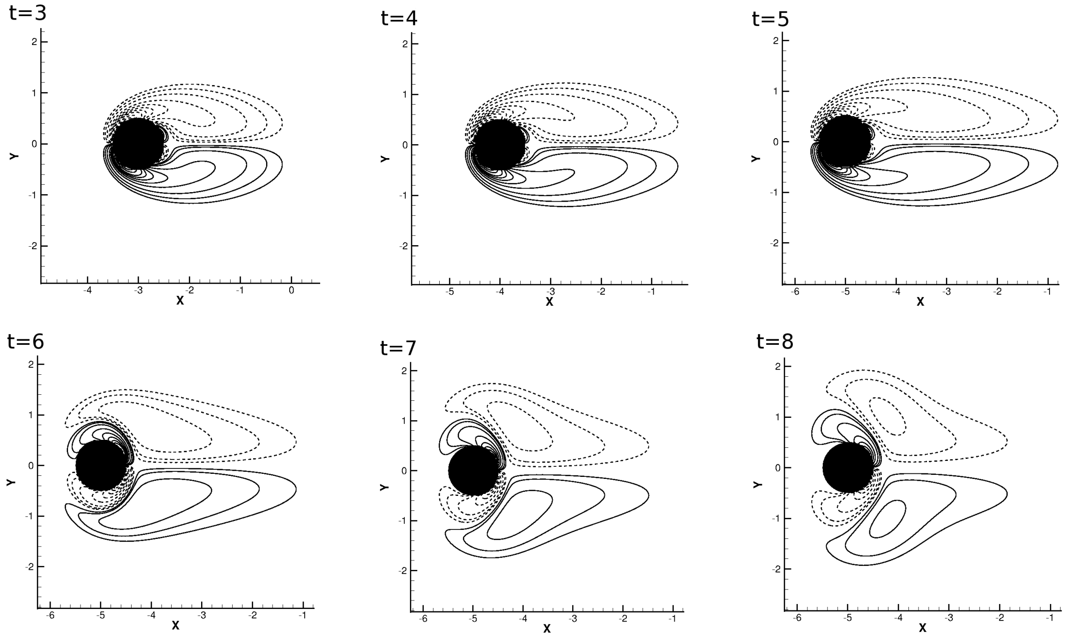
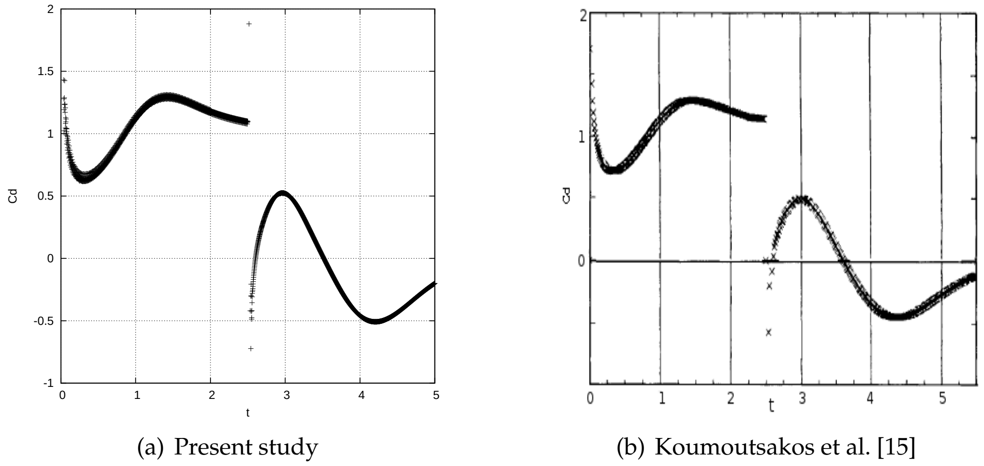


Figure 15: Vorticity field for an impulsively started cylinder stopped at $t=5$ for $Re=40$.

a box $[0;2]\times[0;6]$ where a two-dimensional circular cylinder is initially located in $(1;4)$ and falls under gravity. The cylinder's radius is $r=0.25$ and its density is $\rho_s=1.5$. The viscosity and density of the fluid are respectively $\nu=0.01$ and $\rho_f=1$. The gravity is chosen to be $g=-980$. The space and time steps are $\Delta x=\Delta y=1/256$ and $\Delta t=10^{-5}$ respectively.



(a) Present study

(b) Koumoutsakos et al. [15]

Figure 16: Comparison of the Drag coefficients for an impulsively started cylinder stopped at $t=2.5$ for $Re=550$ (using our non-dimensionalized time).

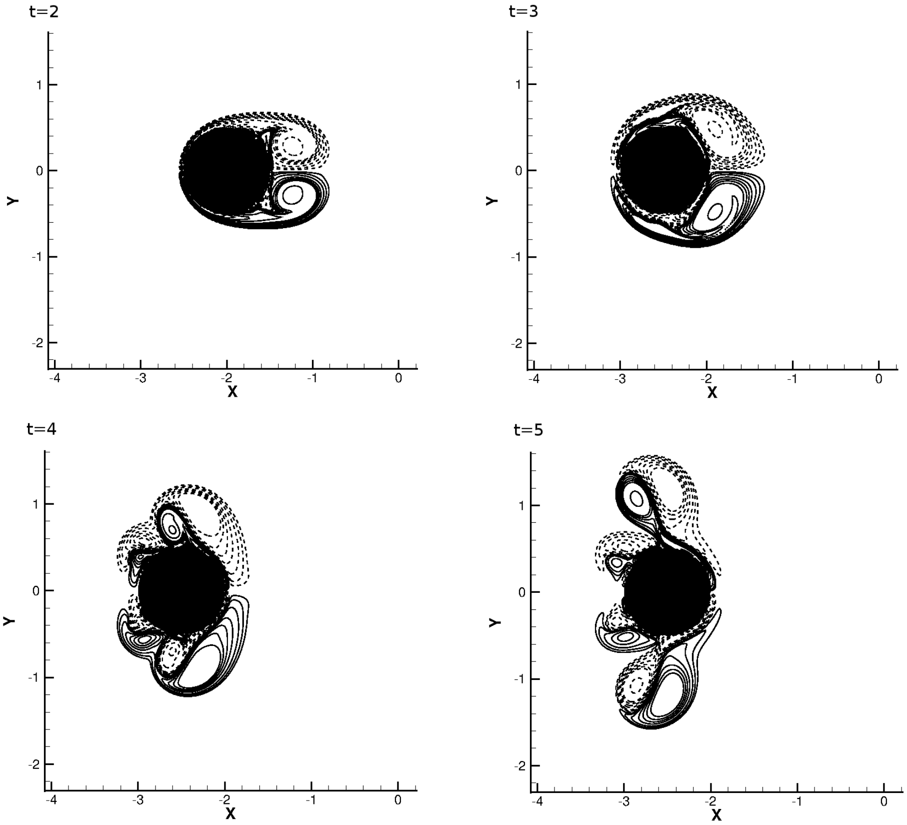


Figure 17: Vorticity field for an impulsively started cylinder stopped at $t=2.5$ for $Re=550$.

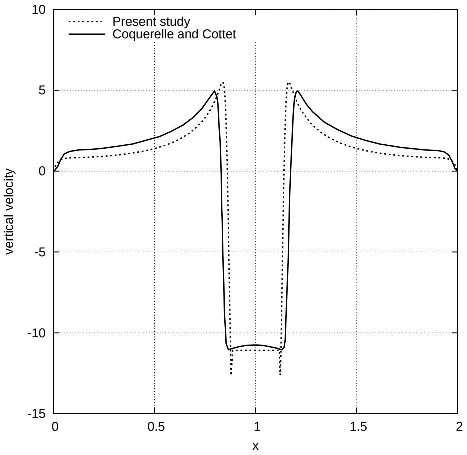


Figure 18: Comparison of the vertical velocity on an horizontal cut through the center of the cylinder falling under gravity at time $t=0.1$, between our current method and Coquerelle and Cottet's [6].

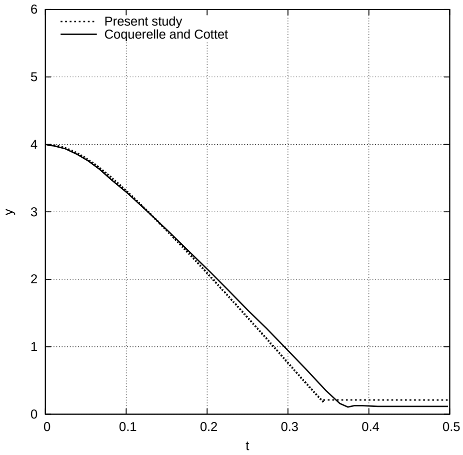


Figure 19: Comparison of the height evolution in time of the center of a cylinder falling under gravity, between our current method and Coquerelle and Cottet's [6].

Through the Figs. 18 and 19, we can see that our results match [6]. Furthermore, in Fig. 18, we can see that two points inside the cylinder have a vertical velocity lower than the others. These points are the ghost points and their lower velocity correspond to the corrected velocity induced by the IPC method.

3.4 A three-dimensional example: the jellyfish-like swimming

The numerical approach described in Section 2 can easily be extended to the modeling and numerical simulation of three-dimensional self-propelled swimmers. As an example, we study the flow generated by a moving and deformable body that is a simplified

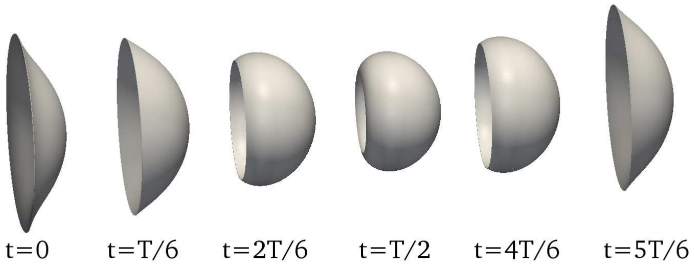


Figure 20: Sketch of the jellyfish shape during one stroke.

prototype for a jellyfish. The prototype has a lens type shape with diameter $D=4$cm and maximal thickness $\ell=D/10$. The swimming shape is obtained deforming the lens shape. We roll up this profiles onto a sphere with radius $r(t)$. In this study $r(t)$ oscillates periodically between $r_{\min}=1.7D/(2\pi)$ and $r_{\max}=3.3D/(2\pi)$ with frequency $f=1$Hz. We also slightly deform the extremity of that profile using frequency $f=1$Hz. This overall deformation does not preserve the volume of the jellyfish. We thus perform an homothetic transformation of the whole body to recover (with maximal error of 3%) the undeformed lens volume. These swimming parameters ($r_{\min}$, $r_{\max}$ and f) are chosen to mimic a realistic jellyfish. The jellyfish shape during one swimming stroke is presented in Fig. 20. The fluid and body density is $\rho=1000$ kg/m^3 and the fluid dynamic viscosity is $\mu=10^{-3}$Pa·s.

The mass center of the jellyfish is initially centered at $(x,y,z)=(0,0,0)$. The jellyfish moves under the effect of the force and torques integrated over its surface, according to Newton laws as explained in [3]. The jellyfish swims in the positive x-direction in an "aquarium" with computational domain $[-2D,6D]\times[-2D,2D]\times[-2D,2D]$. The mesh is composed by $600\times300\times300=54$ millions of nodes. The jellyfish swims in a fluid at rest ($\boldsymbol{u}=\boldsymbol{0}$ at $t=0$) and homogeneous Dirichlet boundary conditions ($\boldsymbol{u}=\boldsymbol{0}$) are used on all the boundaries.

It is known, see [16], that mass and momentum conservation perturbations near the moving bodies lead to considerable force oscillations and displacement inaccuracies when the immersed bodies incorporate new grid points or new grid points are released from the immersed body to the fluid. In Fig. 22 we show the plot of the force acting on the jellyfish. The plot shows that in our case these perturbations cause oscillations that are very small, provided that the largest possible CFL is taken. The pressure field in a 2D section close to the swimming body is shown in Fig. 23. It is seen that the pressure is continuous across the body boundaries. The pressure gradient is not necessarily orthogonal to the body, as it should be. No boundary conditions for pressure are applied on the immersed body, thanks to the penalization step. Also, we show a plot a the divergence after the projection step, Fig. 24. Of course the divergence is of the order of the residuals of the linear problem solved for projection, as expected. Moreover, the divergence residual is uniformly distributed near the boundary.

After a transient regime the jellyfish reaches an average velocity around 2.5cm/s (see

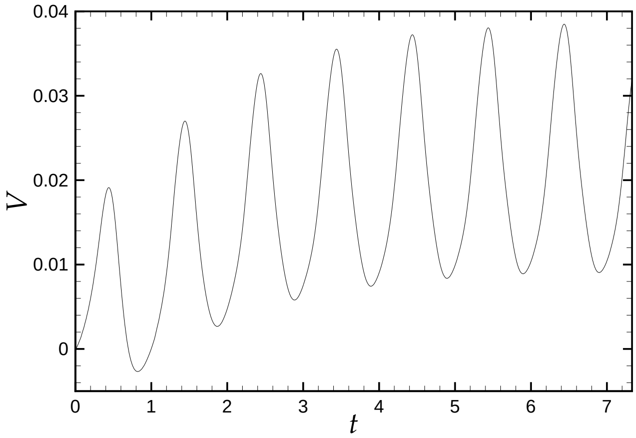


Figure 21: Temporal evolution of the jellyfish velocity.

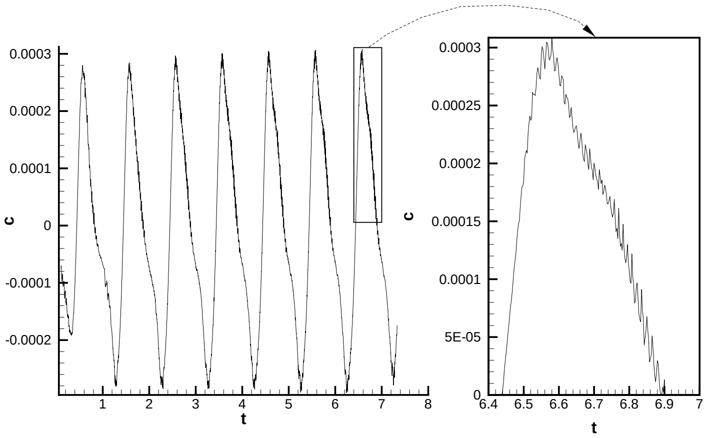


Figure 22: Temporal evolution of the forces exerted by the jellyfish.

Fig. 21). The associated Reynolds number is then $Re=\rho VD/\mu\approx 1000$. The wake generated by the jellyfish is presented in Fig. 25. It shows alternate counter-rotating vortex rings. Indeed, vortex pairing of counter-rotating vortices can also be observed in Fig. 26 where the z-vorticity is plotted in the plan $z=0$. This vortex pattern is similar to that generated by a two-dimensional jellyfish [3].

This coherent structure dynamics compares well to what is experimentally found in [7] by PIV analysis of the flow induced by an actual oblate medusian jellyfish (Aurelia Aurita) swimming in its natural marine habitat. Although the jellyfish swim simulated in this paper is realistic, quantitative comparisons are difficult because of the lack of details on the actual jellyfish geometry and swimming law. Nonetheless, the flow patterns are in remarkable qualitative accordance with the one observed in nature.

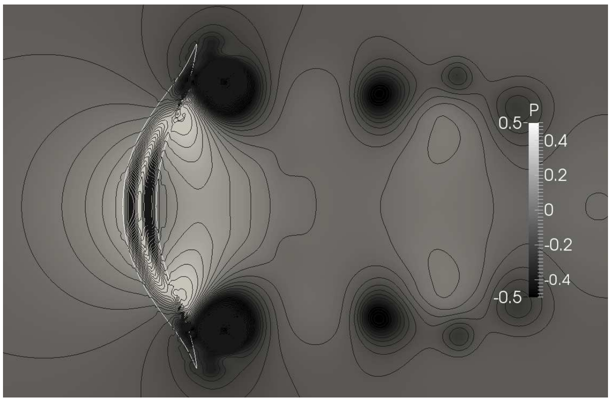


Figure 23: Snapshot of the pressure field generated by the jellyfish $(z=0)$.

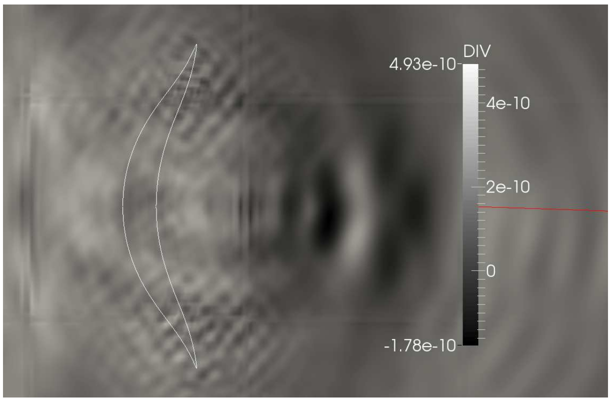
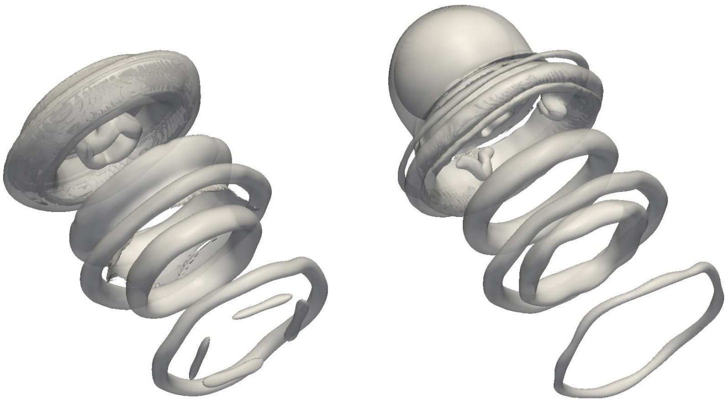
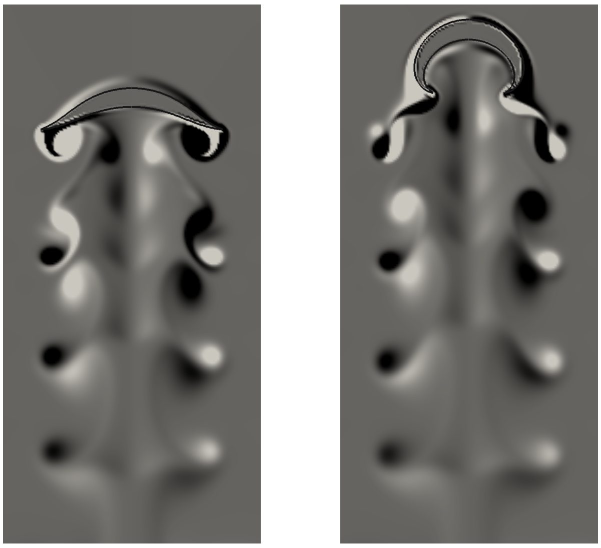


Figure 24: Snapshot of the divergence of the velocity field generated by the jellyfish $(z=0)$.



(a) $t=6$ (b) $t=6.5$

Figure 25: Vortices generated by a swimming jellyfish.



(a) $t=6$ (b) $t=6.5$

Figure 26: Vorticity visualization in the plan $z=0$. Black and white zones represent negative and positive vorticity, respectively.

4 Conclusions

We have presented an immersed boundary approach called *Image Point Correction* method (IPC). It is applied to the simulation of incompressible flow past complex moving boundaries captured by the level set function. This method is based on the Ghost-Cell approach and the penalty method. The solver is second order accurate and allows us to use cartesian grids instead of body-fitted meshes. The main advantage of this method is that the Navier-Stokes equations are solved on the entire domain without making difference between fluid nodes from the solid ones. Besides, the use of a classical penalty step enforces the right pressure boundary conditions on the interface without any special treatment and gives to the ″fresh cells″ a physical velocity.

In order to validate the model, a series of numerical simulations is exposed, covering a wide range of canonical tests. The computed results agreed with previous studies, whether it is for one or several steady cylinders, or a cylinder in motion. These cases demonstrate the ability of the solver to handle several, complex, moving objects. Also a three-dimensional self propelled jellyfish is simulated highlighting the ability of the method to simulate complex flows with non-trivial geometries.

Acknowledgments

This work has been supported by French National Research Agency (ANR) through COSINUS program (project CARPEINTER nANR-08-COSI-002).

Numerical experiments presented in this paper were carried out using the PLAFRIM experimental testbed, being developed under the Inria PlaFRIM development action with support from LABRI and IMB and other entities: Conseil Régional d'Aquitaine, FeDER, Université de Bordeaux and CNRS (see `https://plafrim.bordeaux.inria.fr/`).

References

[1] P. Angot, C. H. Bruneau and P. Fabrie, A penalization method to take into account obstacles in incompressible flows, Numer. Math., 81(4) (1999), 497–520.

[2] M. Bergmann, Optimisation Aérodynamique par Réduction de Modèle POD et Contrle Optimal, Application au Sillage Laminaire D'un Cylindre Circulaire, PhD thesis, Institut National Polytechnique de Lorraine, 2004.

[3] M. Bergmann and A. Iollo, Modeling and simulation of fish-like swimming, J. Comput. Phys., 230 (2011), 329–348.

[4] M. Braza, P. Chassaing and H. H. Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, J. Fluid. Mech., 165 (1986).

[5] A. Chorin, Numerical solution of the Navier Stokes equations, Math. Comput., 22 (1968), 746–762.

[6] M. Coquerelle and G. H. Cottet, A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies, J. Comput. Phys., 227(21) (2008), 9121–9137.

[7] J. Dabiri, S. Colin, J. Costello and M. Gharib, Flow patterns generated by oblate medusan jellyfish: field measurements and laboratory analyses, J. Experimental Bio., 208 (2005), 1257–1265.

[8] D. Dehkordi, H. Moghaddam and H. Jafari, Numerical simualtion of flow over two circular cylinders in tandem arrangement, J. Hydrodyn., 23 (2011), 114–126.

[9] H. Ding, C. Shu and K. Yeo, Numerical simulation of flows around two circular cylonders by mesh-free least square-based finite difference methods, Int. J. Numer. Meth. Fluids, 53 (2007), 305–332.

[10] F. Duarte, R. Gormaz and S. Natesan, Arbitrary lagrangian-eulerian method for navier stokes equations with moving boundaries, Comput. Methods Appl. Math. Eng., 193 (2004), 4819–4836.

[11] R. Ghias, R. Mittal and H. Dong, A sharp interface immersed boundary method for compressible viscous flows, J. Comput. Phys., 225 (2007), 528–553.

[12] F. Gibou, R. Fedkiw, L. Cheng and M. Kang, A second order accurate symmetric discretization of the poisson equation on irregular domains, J. Comput. Phys., 176 (2002), 205–227.

[13] R. Henderson, Details of the drag curve near the onset of vortex shedding, Phys. Fluids, 7 (1995), 2102–2104.

[14] G. Jin and M. Braza, A nonreflecting outlet boundary condition for incompressible unsteady Navier-Stokes calculations, J. Comput Phys., 107(2) (1993), 239–253.

[15] P. Koumoutsakos and A. Leonard, High-resolution simulations of the flow around an impulsively started cylinder using vortex methods, J. Fluid Mech., 296 (1995), 1–38.

[16] J. Lee, J. Kim, H. Choi and K. S. Yang, Sources of spurious force oscillations from an immersed boundary method for moving-body problems, J. Comput. Phys., 230(7) (2011), 2677–2695.

[17] Z. Li and M. Lai, The immersed interface method for the navier stokes equations with singular forces, J. Comput. Phys., 171 (2001), 822–842.

[18] C. C. Liao, Y. W. Chang, C. A. Lin and J. M. McDonough, Simulating flows with moving rigid boundary using immersed-boundary method, Comput. Fluids, 39(1) (2010), 152–167.

[19] H. Liu, S. Krishnan, S. Marella and H. Udaykumar, Sharp interface castesian grid method ii: a technique fir simulationg droplet interactions with surfaces of arbitrary shape, J. Comput. Phys., 210 (2005), 32–54.

[20] N. Mahir and Z. Altac, Numerical investigation of convective heat transfer in unsteady flow past two cylinders in tandem arrangements, Int. J. Heat Fluid Flow, 29 (2008), 1309–1318.

[21] S. Marella, S. Krishnan, H. Liu and H. Udaykumar, Sharp interface cartesian grid method i: an easily implemented technique for 3d moving boundary computations, J. Comput. Phys., 210 (2005), 1–31.

[22] J. Meneghini and F. Satara, Numerical simulation of flow interference between two cylinders in tandem and side-by-side arrangements, J. Fluids Struct., 15 (2001), 327–350.

[23] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas and A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comput. Phys., 227 (2008), 4825–4852.

[24] R. Mittal and G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid. Mech., (2005), 1–27.

[25] S. Mittal, V. Kumar and A. Raghuvanshi, Unsteady incompressible flows past two cylinders in tandem and staggered arrangements, Int. J. Numer. Meth. Fluids, 25 (1997), 1315–1344.

[26] S. Osher and R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2003.

[27] S. Osher and J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations, J. Comput. Phys., 79(12) (1988).

[28] P. G. W. Ploumhans, Vortex methods for high-resolution simulations of viscous flow past bluff bodies in general geometry, J. Comput. Phys., 165 (2000), 354–406.

[29] C. Peskin, Flow patterns around the heart valves, J. Comput. Phys., 10 (1972), 252–271.

[30] J. H. Seo and R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, J. Comput. Phys., 230(19) (2011), 7347–7363.

[31] J. Sethian, A fast marching level set method for monotonically advancing fronts, Appl. Math., 93 (1996), 1591–1595.

[32] J. A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, Cambridge, UK, 1999.

[33] J. A. Sethian, Evolution, implementation, and application of level set and fast marching methods for advancing fronts, J. Comput. Phys., 169 (2001), 503–555.

[34] B. Sharman, F. Lien and L. Davidson, Numerical predictions of low reynolds number flows over two tandem circular cylinders, Int. J. Numer. Meth. Fluids, 47(5) (2005), 423–447.

[35] A. Slaouti and P. Stansby, Flow around two circular cylinders by random-vortex method, J. Fluids Struct., 6(6) (1992), 641–670.

[36] M. Sussman, P. Smereka and S. Osher, A level set approach for computing solutions to incompressible 2-phade flow, J. Comput. Phys., 114 (1994), 146–159.

[37] 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.

[38] G. Tryggvason, B. Bunner, A. Esmaeeli and N. Al-Rawahi, Computational of multiphase flows, Adv. Appl. Mech., 39 (2003), 91–120.

[39] C. Wieselsberger, New data on the laws of fluid resistance, NACA TN, 84 (1922).

[40] Y. Yang and H. Udaykumar, Sharp interface castesian grid method iii: solidification of pure materials and binary solutions, J. Comput. Phys., 210 (2005), 55–74.