CILAMCE 2017
Florianópolis - Santa Catarina - Brazil

A SEMI-IMPLICIT IMMERSED BOUNDARY METHOD FOR FLOW OVER SHARP BODIES

João Rodrigo Andrade

Aristeu Silveira Neto

João Marcelo Vedovoto

joao.andrade@ufu.br

aristeus@ufu.br

vedovoto@ufu.br

Fluid Mechanics Laboratory, Dep. of Mechanical Engineering, Federal University of Uberlândia, Av. João Naves de Ávila, 2121, Campus Santa Mônica, Uberlândia, Minas Gerais, Brazil

Ramon Silva Martins

ramon.martins@ufes.br

Research Group on Air Quality, Graduate Program on Environmental Engineering, Federal University of Espírito Santo, Av. Fernando Ferrari, 514, Vitória, ES, Brazil

Abstract. *In the present work, the local ghost cell method (LGCM) from Berthelsen and Faltinsen [J. of Comp. Phys., 227, 4354 (2007)] is extended and applied to a semi-implicit temporal integration strategy (SILGCM). The SILGCM is an immersed boundary method for which the boundary conditions are performed directly, it means, by a change in the stencil of the discretized transport equations. The numerical method proposed is applied to the computation of uniform flow over blunt and sharp bodies. This method is investigated by applying the formulation for incompressible laminar flows. In the present paper, the salient features of the methodology is described with special emphasis on the capacity of computation of flow past sharp bodies. The transport equations are discretized by using the finite differences method, the immersed boundary is represented with a finite number of Lagrangian points, distributed over the solid-fluid interface. A Cartesian grid is used to solve the fluid flow equations. To test the accuracy of this approach, four different benchmark cases are applied. It was shown that the semi-implicit local ghost cell method presents good agreement with analytic, numerical benchmark and experimental data of all simulated cases.*

Keywords: *Immersed boundary method, ghost cell method, Cartesian grid method, thin geometry body, sharp geometry body*

1 INTRODUCTION

1.1 General aspects

The computational fluid dynamics (CFD) of flows past solid irregular bodies is a major topic of research due to its importance for industrial applications. A major difficulty for such numerical computation is the simulation of the complex geometries of different situation, such as, for instance, moving bodies and fluid structure interaction problems, e.g., flutter-instable oscillations, limit cycle oscillations (LCO), and vortex induced vibration (VIV): this has been one of the main issues in computational fluid dynamics. There are basically two main approaches to simulate complex flows with immersed boundaries: using unstructured body-conforming meshes, and the immersed boundary method (IBM), the latter of which is the main subject of the present paper. In the IBM, a body in the flow field is considered as an external entity to the fluid itself, therefore allowing the use of simpler orthogonal grids where only the balance equations will be solved. A complete disconnected Lagrangian set of points representing the immersed object can be positioned anywhere in the computational domain, regardless of the centers, vertex, or face positions of a Cartesian grid. The main advantages of the immersed boundary method are savings in memory and CPU use, and easy grid generation compared to the unstructured grid method.

The immersed boundary method (IBM) was first proposed by Peskin (1997) to simulate blood flow in heart valves by using the finite difference method, a highly complex problem that involves moving boundaries and complex geometries. The central idea related to Peskin's method is the use of two independent meshes: the Eulerian one, which is used to solve the fluid flow equations, and an independent Lagrangian mesh, which represents the immersed body. This procedure allows modeling complex geometries immersed in a fluid flow, avoiding the use of complex meshes to fit the grid to the immersed body. Since there is no geometric dependence between these two meshes, the IBM can easily handle moving boundary problems without regenerating the grids during the time-marching.

According to Ji at al. (2012), the correct calculation of the body force, which represents the fluid–solid boundary conditions, is a key issue for all type of IB approaches. The main issue in the methods based on the concept of an immersed boundary is how to compute the force term. Based on this, distinct classification systems can be used to categorize the different kinds of IBM. Mittal at al. (2005) classified the IBM using two classes of methods, depending on how the source terms describing the presence of the boundary are imposed: the continuous forcing approach, for which a modification is performed directly in the equations that model the flows, and the discrete forcing approach, where the resulting system of linear equations is changed to take into account the presence of the body. Mittal at al. (2005) also categorized the discrete forcing approaches into those that are formulated so as to impose the boundary condition on the immersed boundary through indirect means, and those that directly impose the boundary conditions on the IB. More details can be found in Mittal at al. (2005).

Many researchers have used the IBM successfully in various applications. These applications cover a broad spectrum of fluid dynamics problems, from flow through heart devices (Peskin,1972,1977), to the interaction of body movements and fluid flow (Cai et al., 2017; Mittal et al., 2005), and supersonic flows (Chi et al., 2017). Some examples may be given: modeling anguilliform swimming (Tyson et al., 2008) and 3-D parachute simulation (Kim et al., 2009).

Uhlmann (2005) presented a direct-forcing immersed boundary (IB) approach using the high-order regularized delta function suggested by Peskin (1972) in the velocity interpolation and in the force distribution. Cai et al. (2017) proposed an implicit immersed boundary method for the simulation of incompressible viscous flow over stationary or moving solid boundaries. It combines the implicit immersed boundary projection method and the explicit direct forcing IBM. Chi et al. (2017) used a ghost-cell immersed boundary approach to represent a solid body in compressible flow simulations in supersonic flows. Also following the line of the ghost-cell method, Tseng et al. (2003) showed good results by computing turbulent flows and Gao et al. (2007) proposed an improved hybrid Cartesian/immersed boundary approach based on a ghost point treatment. Another advantage of the immersed boundary method is the possibility of its application within other discretization methods besides finite difference and finite volume, e.g., Ilinca et al. (2011) and Boffi et al. (2003) used a finite element implementation of the IB method. Cheng et al. (2010) and Feng et al. (2004) applied a combination the most desirable features of the lattice Boltzmann and the IBM. Kinoshita et al. (2016) proposed a novel IBM for flows with thermal effects by using the Fourier pseudospectral method.

1.2 The issue concerning thin/sharp bodies

Especially for the geometries of thin plates and sharp edges, the interpolation function linking the Lagrangian and Eulerian meshes in the IBM can generate some inconsistencies in the discretized transport equations. As a result, some IBM cannot correctly predict the flow over these kinds of geometries. For instance, the so-called diffuse methods have a direct disadvantage of smearing out discontinuities across the boundary. This smearing has an unfavorable effect on the accuracy of the numerical scheme (Berthelsen et al., 2003) and its consequences can be noticed when treating flow over sharp-geometries, e.g flat plate, thin airfoil or streamlined bodies. This undesirable force-spreading behavior is better illustrated in Fig. 1, where the force distribution on a sharp corner is shown. One can notice that the region on which the surface force of a Lagrangian point acts has a larger dimension than the geometry represented and, consequently, it deforms the effective shape of the immersed boundary.

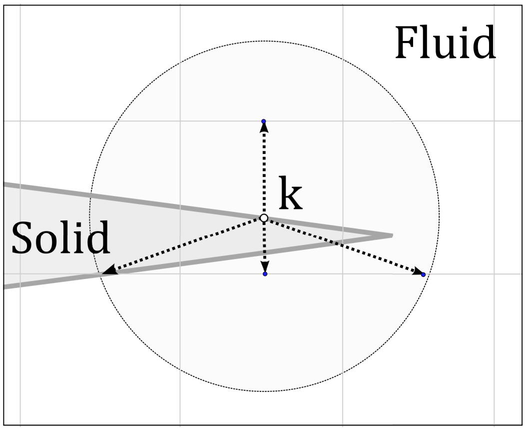

Figure 1: Sharp edge on a staggered grid showing a diffuse IB spreading strategy, where *k* is the Lagrangian point.

Seeking to deal with the problem of discontinuity due to very thin immersed bodies, the so-called sharp methods have been developed (Udaykumar et al., 2001). In this category of IBM, the communication between the immersed boundary and the flow solver is usually accomplished directly by modifying the computational stencil near the immersed boundary. Different kinds of sharp IBM have been developed, e.g., the cut-cell method (CCM) and the immersed interface method (IIM). In the CCM (De Zeeuw et al., 1993; Meinke et al., 2013; Quirket al., 1994; Tucker et al., 1999; Yang et al., 1999; Yee et al., 1999), the immersed body is cut out of the Cartesian mesh and the grid cell is then locally refined to better model the body's curvature. This kind of method is generally used with the finite-volume method. In the IIM (Berthelsen et al., 2004; Le Veque et al., 1997; Li et al., 2001, Wiegmann et al., 2000), a correction term is introduced to the standard differential equation so that the numerical discretization is well-defined across a discontinuous interface. Despite good results, the sharp IB method is limited due to the difficulty of its implementation in three-dimensional geometries.

1.3 Objective

In the present paper, we apply a direct immersed boundary method in order to analyze its ability to properly compute the flow past thin plates and sharp edges, especially those leading to discontinuous regions. An extended version of the local ghost cell method (Berthelsen et al., 2003) with a semi-implicit temporal integration is presented. The method is validated and applied to several benchmark cases: a two-dimensional flow past a stationary circular cylinder; the flow past a circular cylinder between parallel walls; an impulsively started flow past a flat plate; and, lastly, the flow through two parallel channels aligned in the x-direction with different heights separated by a very thin rigid interface.

2 BASIC MATHEMATICAL MODEL AND NUMERICAL METHOD

In this section, the differential mathematical model is presented as well as the numerical procedure to solve the resulting transport equations. In order to solve for the flow around an arbitrary body with a complex geometry, incompressible two-dimensional laminar flows are modeled by using the Navier–Stokes and continuity equations. The numerical methods used are based on a finite-difference approach on a staggered grid. A semi-implicit scheme is adopted. The central difference scheme (CDS) is applied to express both the diffusive and advective contributions of the transport equations. The fractional-step (Kim et al., 1985) algorithm is used for pressure–velocity coupling.

2.1 Mathematical model

A rectangular domain is used and discretized with an Eulerian grid in a Cartesian reference frame. The momentum and continuity equations for a viscous incompressible fluid flow with constant specific mass and viscosity can be written as Eqs. 1 and 2, respectively,

$$\rho\left[\frac{\partial u_i}{\partial t}+\frac{\partial(u_iu_j)}{\partial x_j}\right]=-\frac{\partial p}{\partial x_i}+\mu\frac{\partial^2 u_i}{\partial x_j^2}+f_i, \tag{1}$$

$$\frac{\partial u_i}{\partial x_i}=0, \tag{2}$$

where ρ is the specific mass, the u_i are the components of the fluid velocity, p is the pressure, μ is the dynamic viscosity, and the f_i are the Eulerian body force components.

It is necessary to provide boundary conditions to the equations above. In the context of IBM, these boundary conditions can be divided into two major groups: the boundary conditions related to the limits of the domain, and the boundary conditions at the interface between the fluid and solid domains. The first ones are specific to each case and will be treated in Section 4. The boundary conditions for the fluid–solid interface of the IB are generally given by Eq. 3,

$$u_i = u_{i,\Gamma} \text{ in } \Omega_\Gamma, \tag{3}$$

where $u_{i,\Gamma}$ are the velocity components of the immersed body.

The whole domain is denoted by Ω. The subdomain defined by the immersed solid body is Ω_Γ and the fluid surrounding it is Ω_f. The immersed boundary is represented by Γ. These equations are solved in the Eulerian domain and the coupling between both grids (Eulerian and Lagrangian) is accomplished by the force source term, which is defined in the whole domain Ω, but presents values different from zeros only at the points that coincide with the immersed geometry, enabling the Eulerian field to detect the presence of a solid interface. Such behavior is expressed by Eq. 4,

$$\vec{f}(\vec{x},t) = \int_\Gamma \vec{F}(\vec{X}_k,t)\delta(\vec{X}_k - \vec{x})d\vec{X}_k, \tag{4}$$

where $\vec{F}(\vec{X}_k,t)$ is calculated over the domain Γ. This integral transfers the Lagrangian field $\vec{F}(\vec{X}_k,t)$ to the Eulerian field $\vec{f}(\vec{x},t)$, where $\vec{X}_k$ describes the interface Γ, and $\vec{x}$ describes the domain Ω. It is worth noting that a direct method which directly imposes the boundary conditions on the IB can also be expressed by a force term, even if it plays a role only in the discretization of the transport equations.

2.2 Numerical method

In this section, the numerical procedures chosen to solve the transport equations presented above are described. The continuity and Navier–Stokes equations are discretized employing the finite-difference method with staggered arrangements of cells for the velocities and a cell-centered arrangement for the pressure. The equations are integrated in time using the fractional step method based on the semi-implicit first order Euler method, treating the advective term explicitly and the diffusive term implicitly, which eliminates the viscous stability constraint.

Time-stepping procedure

The equations are integrated in time by using the fractional step method (Kim et al., 1985; Van Kan et al., 1986), which consists of three substeps. Firstly, a modified momentum equation is solved and an intermediate velocity, u_i^*, is obtained. A fully implicit scheme is employed for the diffusive terms, while the advective terms are discretized using an explicit scheme. In this substep, the following modified momentum equation is integrated with a semi-implicit first order Euler scheme:

$$\frac{u_i^* - u_i^n}{\Delta t} = -N\left(u_i^n\right) + L\left(u_i^*\right) + F_i^n, \tag{5}$$

where $N\left(u_i^n\right)$ is the advective term, $L\left(u_i^*\right)$ is the diffusive term and F_i^n is the Lagrangian body force components at the present time step (n). For implicit as well as for semi-implicit methods, a linear system of algebraic equations needs to be solved for the velocity.

The second substep requires the solution of the pressure correction equation:

$$\frac{u_i^{n+1} - u_i^*}{\Delta t} = -\frac{1}{\rho}\frac{\delta p}{\delta x_i}, \tag{6}$$

which is solved with the constraint of having a divergence-free field for the velocity at the next time step, u_i^{n+1}. This gives the following Poisson's equation for the pressure correction:

$$\frac{\partial}{\partial x_j}\left(\frac{\partial p}{\partial x_i}\right) = \frac{\rho}{\Delta t}\frac{\partial u_i^*}{\partial x_j}. \tag{7}$$

Once the pressure correction is obtained, the velocity is updated by

$$u_i^{n+1} = u_i^* - \frac{\Delta t}{\rho}\frac{\partial p}{\partial x_i}. \tag{8}$$

This procedure completes the time step loop. Normally, one loop is sufficient to obtain a divergence-free velocity field.

Spatial discretization of the transport equations

As already mentioned at the beginning of the present section, the central difference scheme (CDS) is applied to express both the diffusive and advective terms of the transport equations. To illustrate the discretization of a derivative using the CDS approach, consider the grid cell of lengths $\Delta x_i \times \Delta y_j$ shown in Fig. 2. The indices (i, j) denote the node at which the partial differential equation is approximated. Note that the pressure, $p_{i,j}$, is located at the center of the cell, whereas $u_{i,j}$ and $v_{i,j}$ represent the velocity components at its faces.

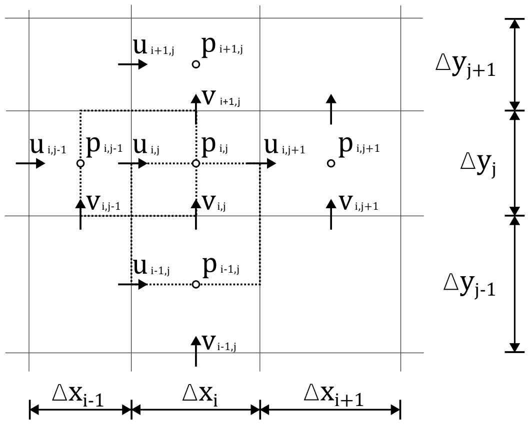

Figure 2: Sketch describing the naming convention and location of velocity components employed in the spatial discretization of the transport equations.

The structure adopted for the computational grid can be uniform or non-uniform. Therefore, it is necessary to perform interpolations to discretize any spatial derivatives. In the case of non-uniform grids, it is important that such interpolations are obtained by using distance-weighted rules (Ferziger et al., 1996). Boundary conditions must be given in order to solve the set of transport equations. In the numerical code developed for the present research, it is possible to choose between periodic, Dirichlet, Neumann, or advective boundary conditions (Orlanski, 1976) for each boundary of the computational domain.

Employing the temporal integration method described above and the discretization strategy chosen, the result of the numerical form of Eq. 5 far from the immersed body is a system of linear algebraic equations of the form

$$A_{i,j}u_{i,j} + A_{i+1,j}u_{i+1,j} + A_{i-1,j}u_{i-1,j} + A_{i,j+1}u_{i,j+1} + A_{i,j-1}u_{i,j-1} = Q_{i,j}, \tag{9}$$

where $Q_{i,j}$ is the source term depending on the velocity at the preceding time step and $A_{i,j}$ the coefficients. The approximations for the y-momentum equation can be formed in a similar manner.

3 IMMERSED BOUNDARY METHOD

How the boundary conditions are imposed on the immersed boundary determines, for the most part, how an IB algorithm is performed, and distinguishes one IB method from another. In general, the modification takes the form of a forcing function in the transport equations that reproduces the effect of the boundary. According to Mittal at al. (2005), introducing a forcing function leads to a division of IB methods into two groups, namely direct and indirect boundary condition imposition. Another common classification is in terms of continuous or discrete forcing. In this section, the mathematical and computational procedure of the immersed boundary method proposed in the present paper will be presented.

3.1 The semi-implicit local ghost cell method

The semi-implicit local ghost cell method (SILGC) was conceived to solve for flows around thin and blunt bodies in an IBM approach. The basic idea in this method is to calculate the flow variables adjacent to the immersed boundary with no dependence on the cells inside the body, such that the boundary conditions in the vicinity of the immersed boundary are satisfied. This method was introduced by Berthelsen et al. (2007) in an explicit approach and it showed good agreement with the results in the literature.

In order to apply the SILGC, a special method was developed to identify the nearest surrounding fluid cells with respect to the Lagrangian points on the suface of the body. The Eulerian cells near to the external surface of the body need to be identified and the computational stencil must be modified, so that it satisfies the boundary conditions. In order to identify the influence of the immersed boundary, it is necessary to establish the grid–interface relation with a given description of the immersed boundary, such as a parametrized curve/surface or a triangulation. In this step, based on Berthelsen et al. (2007), all Cartesian grid center nodes are split into four kinds of grid cells, namely:

- solid cells – as the name suggests, these are cells whose nodes lie inside the solid body;
- neighboring-cells – these are the cells in the fluid domain making up a stencil of one or more neighboring cells in the solid phase;
- fluid-cells – these are the cells whose nodes lie outside the body, in the fluid domain and far from the immersed boundary;
- forcing faces (for velocity faces) – since a staggered grid is being employed, if a velocity face lies in the fluid domain with one pressure cell inside the solid domain as a neighbor, the pressure equation (Eq. 7) cannot be applied and a special treatment is needed. Then, this velocity face has its value determined by an interpolation strategy.

Solid cells and forcing faces are considered “inactive cells” while the others are called “active cells.” Only the active cells inside the fluid domain (Ω_f) are included in the linear system for the momentum equation described in the previous section (Eq. 9).

Velocity determination in the forcing faces

As depicted above, the forcing faces are velocity faces that have an absence of pressure nodes in the composition of its stencil, so that Poisson’s equation cannot be applied and, consequently, the balance of momentum equation becomes impractical for this face. To circumvent this limitation, the velocity $u_{i,j}$ at the forcing face must be determined by considering the influence of the IB and the velocity field. For example, considering Fig. 3, the calculation of the pressure gradient on face $u_{i,j}$ requires the pressure nodes $p_{i,j}$ and $p_{i,j-1}$. However, since $p_{i,j}$ lies inside the solid domain and is separated from $u_{i,j}$ by an interface, this pressure gradient cannot be computed. Consequently, Poisson’s equation cannot be applied in this situation and the velocity $u_{i,j}$ must be imposed depending on the velocity in the IB and the velocity field in Ω_f.

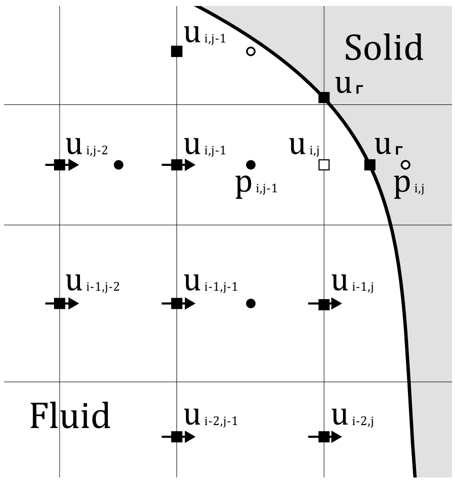

Figure 3: Sketch describing the stencil of the velocity face $u_{i,j}$ where pressure node $p_{i,j}$ lies inside the immersed boundary.

To impose the velocity on a forcing face, an interpolation function must be applied in order to calculate its value. Here, the general form of the interpolation function is a simple 2-D polynomial approach and the calculation of the interpolated velocity is accomplished with the nearest Lagrangian point and the four nearest fluid nodes. The flow variables are assumed to vary in a quadratic manner in both the x and y directions. By expressing the local flow variables as ϕ, the complete form of the interpolation is given by

$$\phi_{i,j} = \sum_{m=1}^{5} a_m \zeta_{i,j}^{(m)} = a_0 + a_1 x_{i,j} + a_2 x_{i,j}^2 + a_3 y_{i,j} + a_4 y_{i,j}^2. \tag{10}$$

where $a_0, a_1, \cdots, a_4$ are the coefficients of the polynomial and $\zeta_{i,j}^{(m)}$ represent the indeterminates of the polynomial ($x_{i,j}$, $x_{i,j}^2$, $y_{i,j}$ and $y_{i,j}^2$) at the face (i, j). The imposed forcing face velocity value is a weighted combination of the values at the neighboring nodes. The five constant coefficients of the assumed polynomial are evaluated from the four neighboring fluid cells and the IB point, and are given by

$$a = B^{-1}\phi, \tag{11}$$

where, for the chosen 2-D interpolation, B is a 5×5 matrix whose elements can be computed from the coordinates of the four neighboring cells and the IB point. B is given by

$$B = \begin{bmatrix} 1 & x_{i-1,j} & x_{i-1,j}^2 & y_{i-1,j} & y_{i-1,j}^2 \\ 1 & x_{i+1,j} & x_{i+1,j}^2 & y_{i+1,j} & y_{i+1,j}^2 \\ 1 & x_{i,j-1} & x_{i,j-1}^2 & y_{i,j-1} & y_{i,j-1}^2 \\ 1 & x_{i,j+1} & x_{i,j+1}^2 & y_{i,j+1} & y_{i,j+1}^2 \\ 1 & x_\Gamma & x_\Gamma^2 & y_\Gamma & y_\Gamma^2 \end{bmatrix} = \begin{bmatrix} \zeta_{i-1,j}^{(1)} & \zeta_{i-1,j}^{(2)} & \cdots & \zeta_{i-1,j}^{(5)} \\ \zeta_{i+1,j}^{(1)} & \zeta_{i+1,j}^{(2)} & \cdots & \zeta_{i+1,j}^{(5)} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_{i,j+1}^{(1)} & \zeta_{i,j+1}^{(2)} & \cdots & \zeta_{i,j+1}^{(5)} \end{bmatrix}. \tag{12}$$

Majumdar et al. (2001) and Tseng et al. (2003) tested this kind of interpolation function for ghost-cell immersed boundaries by using second-order bilinear and quadratic interpolation schemes and obtained good results.

Now let ϕ assume the value of the velocity u. The 2-D polynomial interpolation is evaluated based on the velocity faces neighboring $u_{i,j}$ and ϕ is computed by

$$\phi = \{u_{i-1,j}, u_{i+1,j}, u_{i,j-1}, u_{i,j+1}, u_\Gamma\}. \tag{13}$$

Then the imposed velocity evaluated in the face position (i, j) is given by

$$\begin{aligned} u_{i,j} = &+ \left[B_{11}\zeta_{i,j}^{(1)} + B_{21}\zeta_{i,j}^{(2)} + \cdots B_{51} + \zeta_{i,j}^{(5)}\right] u_{i-1,j} + \left[B_{12}\zeta_{i,j}^{(1)} + B_{22}\zeta_{i,j}^{(2)} + \cdots B_{52} + \zeta_{i,j}^{(5)}\right] u_{i+1,j} \\ &+ \left[B_{13}\zeta_{i,j}^{(1)} + B_{23}\zeta_{i,j}^{(2)} + \cdots B_{53} + \zeta_{i,j}^{(5)}\right] u_{i,j-1} + \left[B_{14}\zeta_{i,j}^{(1)} + B_{24}\zeta_{i,j}^{(2)} + \cdots B_{54} + \zeta_{i,j}^{(5)}\right] u_{i,j+1} \\ &+ \left[B_{15}\zeta_{i,j}^{(1)} + B_{25}\zeta_{i,j}^{(2)} + \cdots B_{55} + \zeta_{i,j}^{(5)}\right] u_\Gamma. \end{aligned} \tag{14}$$

Since in the present paper a semi-implicit temporal integration method is proposed, it is convenient to compute the terms of the interpolation function relative to the velocity faces in an implicit manner. Then Eq. 14 must be included in the linear system given by Eq. 9. Thus, for the imposed velocity in the forcing face, the coefficients from Eq. 9 are replaced by

$$\begin{aligned} A_{i,j} &= 1, \\ A_{i-1,j} &= -\left[B_{11}\zeta_{i,j}^{(1)} + B_{21}\zeta_{i,j}^{(2)} + \cdots B_{51}\zeta_{i,j}^{(5)}\right], \\ A_{i+1,j} &= -\left[B_{12}\zeta_{i,j}^{(1)} + B_{22}\zeta_{i,j}^{(2)} + \cdots B_{52}\zeta_{i,j}^{(5)}\right], \\ A_{i,j-1} &= -\left[B_{13}\zeta_{i,j}^{(1)} + B_{23}\zeta_{i,j}^{(2)} + \cdots B_{53}\zeta_{i,j}^{(5)}\right], \\ A_{i,j+1} &= -\left[B_{14}\zeta_{i,j}^{(1)} + B_{24}\zeta_{i,j}^{(2)} + \cdots B_{54}\zeta_{i,j}^{(5)}\right], \\ Q_{i,j} &= \left[B_{15}\zeta_{i,j}^{(1)} + B_{25}\zeta_{i,j}^{(2)} + \cdots B_{55}\zeta_{i,j}^{(5)}\right] u_\Gamma. \end{aligned} \tag{15}$$

Since forcing faces always have at least one neighboring cell separated by an interface, the general interpolation form given by Eq. 14 cannot be applied, because not all neighboring velocity faces are in the fluid domain. Considering this restriction, the interpolation function must use only the velocity faces inside the fluid domain, Ω_f, and the coefficients of the linear system must be updated. For instance, considering the face (i,j) in Fig. 3, the imposed velocity $u_{i,j}$ can only be evaluated using the velocity faces $u_{i-1,j}$ and $u_{i,j-1}$. So its interpolated value is given by

$$
\begin{aligned}
u_{i,j} = &+ \left[B_{11}\zeta^{(1)}_{i,j} + B_{31}\zeta^{(3)}_{i,j} + B_{51}\zeta^{(5)}_{i,j}\right] u_{i-1,j} \\
&+ \left[B_{13}\zeta^{(1)}_{i,j} + B_{33}\zeta^{(3)}_{i,j} + B_{53}\zeta^{(5)}_{i,j}\right] u_{i,j-1} \\
&+ \left[B_{15}\zeta^{(1)}_{i,j} + B_{35}\zeta^{(3)}_{i,j} + B_{55}\zeta^{(5)}_{i,j}\right] u_{\Gamma},
\end{aligned}
\tag{16}
$$

and the coefficients from Eq. 9 are replaced by

$$
\begin{aligned}
A_{i,j} &= 1, \\
A_{i-1,j} &= -\left[B_{11}\zeta^{(1)}_{i,j} + B_{31}\zeta^{(3)}_{i,j} + B_{51}\zeta^{(5)}_{i,j}\right], \\
A_{i+1,j} &= 0, \\
A_{i,j-1} &= -\left[B_{13}\zeta^{(1)}_{i,j} + B_{33}\zeta^{(3)}_{i,j} + B_{53}\zeta^{(5)}_{i,j}\right], \\
A_{i,j+1} &= 0, \\
Q_{i,j} &= \left[B_{15}\zeta^{(1)}_{i,j} + B_{35}\zeta^{(3)}_{i,j} + B_{55}\zeta^{(5)}_{i,j}\right] u_{\Gamma}.
\end{aligned}
\tag{17}
$$

From the coefficients of the linear system (Eq. 17) it can be seen that there is no link between regions separated by an interface, since there is no dependence between these cells.

Discretization of the momentum equation over neighboring cells

In the presence of a discontinuity due to a sharp obstacle, methods like finite differences or finite volume present regions of discontinuity. According to Linnick et al. (2005) and Berthelsen et al. (2007), a Taylor series expansion can only be applied to a continuous and smooth function, making its use impossible for flows over thin bodies. However, if the function to be considered is piecewise smooth, a remedy to solve this problem can be stated and its higher derivatives are computed. For simplicity, consider the example given by Berthelsen et al. (2007): a generic, one-dimensional function $f(x)$ with $x \in [x_{\text{min}}, x_{\text{max}}]$, which is continuous everywhere except at the point $x = x_\alpha$ ($x_{\text{min}} < x_\alpha < x_{\text{max}}$),

$$
f(x) = \begin{cases} f^-(x) & \text{if } x_{\text{min}} \le x \le x_\alpha \\ f^+(x) & \text{if } x_\alpha < x \le x_{\text{max}} \end{cases}.
\tag{18}
$$

The grid equations are discretized using a uniform grid spacing Δx. For any region in which the relevant variables are smooth, the numerical approximation to the nth derivative of $f(x_i)$ can be written as

$$
f^{(n)}_i = \mathcal{L}^{(n)}(f_l, f_{l+1}, ..., f_i, ..., f_{r-1}, f_r) + \mathcal{O}(\Delta x^p),
\tag{19}
$$

in which $\mathcal{L}$ stands for the discrete finite difference operator and the order of accuracy of the numerical discretization method is given by p. The indices $l, l+1, i, r-1, r$ denote the nodes composing the stencil while i indicates the node at which the differential equation is approximated.

If there is a discontinuity located at x_α, the finite difference must be modified, because it involves grid cells on both sides of the discontinuity point. A procedure to deal with this issue is the following. Since $f(x)$ is piecewise continuous, the derivative operator can be applied considering only the cell values at one side of the discontinuity by considering a smooth extrapolation beyond the interface and its values at the positions of the solid cells in the grid. So, by changing the operator $\mathcal{L}^{(n)}$ to compute the approximate nth derivative of $f(x_i)$, we have

$$f_i^{(n)} = \mathcal{L}^{(n)}(f_l, f_{l+1}, ..., f_i, ..., f_j, f_{j+1}^g, ..., f_{r-1}^g, f_r^g) + \mathcal{O}(\Delta x^p), \tag{20}$$

where $f_{j+1}^g, ..., f_r^g$ are the extrapolated values, referred to as fictitious ghost cell values. The extrapolated values can be determined by any interpolation function.

Based on the above equations (Eqs. 19 and 20), a relation between the grid and the immersed boundary needs to be established for the momentum equation. In this way, for the cells located close to an immersed boundary, Eq. 9 is no longer valid and may be modified by replacing any point separated by an immersed boundary by a temporary fictitious value. This smoothly extends the solution across the boundary and the effect of the body inside the flow can be represented well. For instance, in Fig. 4, if on the stencil of a cell velocity point $u_{i,j}$ the velocity points $u_{i,j+1}$ and $u_{i+1,j}$ lie inside the immersed boundary, they cannot be used to evaluate the velocity at cell (i, j). Using the strategy described above, these velocity points inside the solid domain will be replaced by the values obtained by extrapolating the solution along the x- and/or y-direction. Equations 21 and 22 show the extrapolated velocities for cells $u_{i,j+1}$ and $u_{i+1,j}$, respectively:

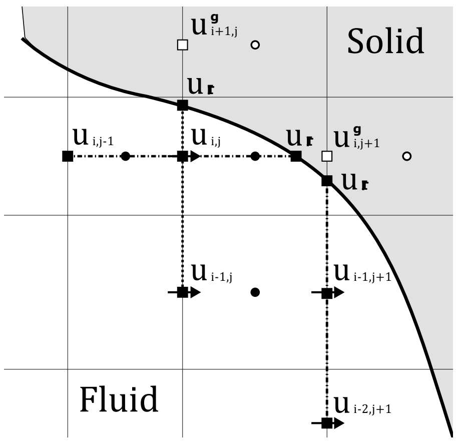

Figure 4: Sketch describing the stencil of the velocity face $u_{i,j}$ where velocities $u_{i,j+1}$ and $u_{i+1,j}$ lie inside the immersed boundary and are replaced by ghost cells.

$$u_{i+1,j}^g = \alpha_0 + \alpha_1 u_{i,j} + \alpha_2 u_{i-1,j} + \alpha_3 u_\Gamma, \tag{21}$$

and

$$u_{i,j+1}^g = \beta_0 + \beta_1 u_{i,j} + \beta_2 u_{i,j-1} + \beta_3 u_{i-1,j+1} + \beta_4 u_{i-2,j+1} + \beta_5 u_\Gamma, \tag{22}$$

where α and β are the coefficients of the extrapolation function depending on the geometric dimensions. The extrapolation function applied here is given by Eq. 10. For this example (Fig. 4), the coefficients from Eq. 9 are replaced by

$$A^g_{i,j}u_{i,j} + A^g_{i-1,j}u_{i-1,j} + A^g_{i,j-1}u_{i,j-1} + A^g_{i-1,j+1}u_{i-1,j+1} + A^g_{i-2,j+1}u_{i-2,j+1} = Q^g_{i,j}, \quad (23)$$

where

$$\begin{aligned}
A^g_{i,j} &= A_{i,j} + A_{i+1,j}\alpha_1 + A_{i,j+1}\beta_1,\\
A^g_{i,j-1} &= A_{i,j-1} + A_{i,j+1}\beta_2,\\
A^g_{i-1,j} &= A_{i-1,j} + A_{i+1,j}\alpha_2,\\
A^g_{i-1,j+1} &= A_{i,j+1}\beta_3,\\
A^g_{i-2,j+1} &= A_{i,j+1}\beta_4,
\end{aligned} \quad (24)$$

and $Q^g_{i,j}$ is similar to $Q_{i,j}$ (Eq. 9) with the velocities $u_{i+1,j}$ and $u_{i,j+1}$ being replaced by $u^g_{i+1,j}$ and $u^g_{i,j+1}$, respectively. Note that a part of the calculation of the interface source term, given by u_Γ, is also incorporated into the implicit term. In the present paper, to obtain the ghost cell values, a third-order Lagrange extrapolation polynomial scheme is used.

In Poisson's equation for pressure (Eq. 7) at cells close to the immersed boundary, a similar procedure to that for the velocity equation is used to compute its derivatives: inactive cells are replaced by ghost values depending on the fluid domain and on the IB. The approach applied in the present paper is the same as that used by Berthelsen et al. (2007) and it is not detailed here.

4 RESULTS

In the present paper, the semi-implicit local ghost cell method (SILGC) have been applied to several benchmark cases at various Reynolds numbers in order to validate the method and the numerical procedure. The cases are: the flow past a circular cylinder, the flow past a circular cylinder between parallel walls, an impulsively started flow past a flat plate and a separated plane channel flow. All the calculation were compared with results reported by the literature.

A rectangular domain was used for all the simulated cases and a stretched Cartesian grid with refined cells near the immersed boundary was applied. The grid was chosen for each case based on a grid refinement study, omitted in the present paper for conciseness. The boundary conditions were imposed in such a way that the flow was from the left face toward the right face of the domain. The fluid properties ρ and μ were considered constants. The dimensions of the domain were set so as to minimize the boundary effects on the development of the flow.

Dirichlet boundary conditions ($u/U = 1; v = 0$, where U is the inflow velocity) was used at the inflow boundary. On the body's surface, non-slip conditions were imposed ($u = 0, v = 0$). A Neumann-type boundary condition was adopted at the outflow and lateral boundaries for open flows. The Reynolds number was defined as

$$Re = \frac{\rho U D}{\mu}, \quad (25)$$

where D is the characteristic length. The dimensionless time was defined as

$$T = \frac{Ut}{D}. \quad (26)$$

Once the velocity and pressure fields are calculated, the drag and lift coefficients and the Strouhal number can also be determined, using the force field directly. As demonstrated by Lai et al. (2000), the drag force on an immersed body in a stream arises from two sources: the shear stress and the pressure distribution along the body. The drag coefficient can be defined as

$$C_D = \frac{F_D}{(1/2)\rho U^2 D}, \tag{27}$$

where F_D is the drag force. Here, it can be calculated using the Eulerian force or the Lagrangian force:

$$F_D = -\int_\Gamma F_x dx = -\int_0^L -f_x ds, \tag{28}$$

where F_x and f_x are the x components of the Lagrangian and Eulerian forces, respectively, and L is the length of the interface.

Similar to the drag coefficient, the lift coefficient can be defined as

$$C_L = \frac{F_L}{(1/2)\rho U^2 D}, \tag{29}$$

where F_L is the lift force, which can be calculated using the y component of the Eulerian force or the Lagrangian force:

$$F_L = -\int_\Gamma F_y dy = -\int_0^L -f_y ds, \tag{30}$$

where F_y and f_y are the y components of the Eulerian and Lagrangian forces, respectively.

The Strouhal number, St, is defined as the dimensionless frequency at which the vortices are shed behind the body:

$$St = \frac{fD}{U}, \tag{31}$$

where f is the vortex shedding frequency. This frequency can be obtained using the Fast Fourier Transform of the lift coefficient time distribution.

4.1 Laminar flow past 2D stationary circular cylinder

A laminar flow past a two-dimensional stationary circular cylinder is calculated to verify the methods developed in this paper. The domain has a length of $120D$ and a width of $60D$, where D is the diameter of the cylinder. Simulations were performed for Reynolds numbers ranging from 10 to 300. A computational mesh of $\approx 500\,000$ volumes was used. The center of the cylinder was placed at $x/D = 50$ and $y/D = 30$.

Figure 5 shows the streamlines neighboring the cylinder and the recirculation bubble behind the cylinder at $Re = 40$ in the steady flow regime obtained through the semi-implicit local ghost cell method. The bubble increases with the Reynolds number. At Reynolds numbers equal to 10, 20 and 40, the wake formed behind the cylinder attains a steady symmetric state and at higher Reynolds numbers it becomes unstable, which is in agreement with the well-established

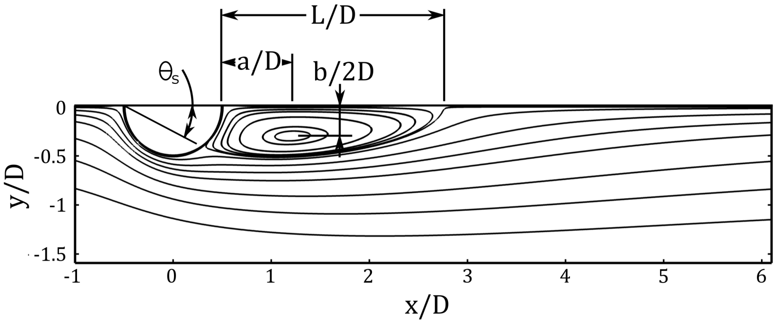

Figure 5: Circular cylinder streamlines at $Re = 40$. Nomenclature used in Tab. 1.

Table 1: Steady uniform flow past a circular cylinder for $Re = 40$: length L of recirculation zone, location (a, b) of vortex center and separation angle θ_S (nomenclature given in Fig. 5).

Reference	L/D	a/D	b/D	θ_S
Coutanceau et al. (1977)	2.13	0.76	0.59	53.5
Calhoun (2002)	2.18	-	-	54.2
Le et al. (2006)	2.22	-	-	53.6
Linnick et al. (2005)	2.28	0.72	0.60	53.6
Xu et al. (2006)	2.21	-	-	53.5
SILGC	2.23	0.73	0.60	53.0

result of the linear stability theory. The cylinder wake instabilities rise for $Re \geq 47$. The length of the recirculation zone, the location of the centers of the vortex, and the angle of separation are given in Tab. 1, where they are compared with experimental and other numerical results. This indicates that the boundary condition is satisfied.

Figure 6-(a) shows the pressure coefficient on the cylinder surface, defined as $Cp = 2(p - p_0)/(\rho U^2)$; Fig. 6-(b) shows the vorticity coefficient $\omega D/U$ on the cylinder surface at $Re = 40$. Both variables are functions of the angle θ and are taken at the Lagrangian points $\vec{X}_k$.

The comparison of the drag coefficient obtained in the present paper with other numerical and experimental results Park et al. (1998), Yee et al. (1999) and Triton (1959) is presented in Tab. 2. Very good agreement has been obtained.

4.2 Suddenly accelerated normal plate

A laminar flow past a suddenly accelerated normal plate has been studied. This geometry illustrates a flow around a thin body in a position that is normal to the flow. The domain has a length of $120D$ and a width of $60D$, where D is the height of the plate. The center of the plate was placed at $x/D = 50$ and $y/D = 30$. The simulation was performed at Reynolds number $Re = 126$ (based on the height of the plate and the uniform free stream velocity), in order to compare it with results of Koumoutsakos et al. (1996) and Taneda et al. (1971). A computational mesh of $\approx 700\ 000$ volumes was used. The plate is considered as an infinitesimally thin finite flat plate accelerating normal to its surface.

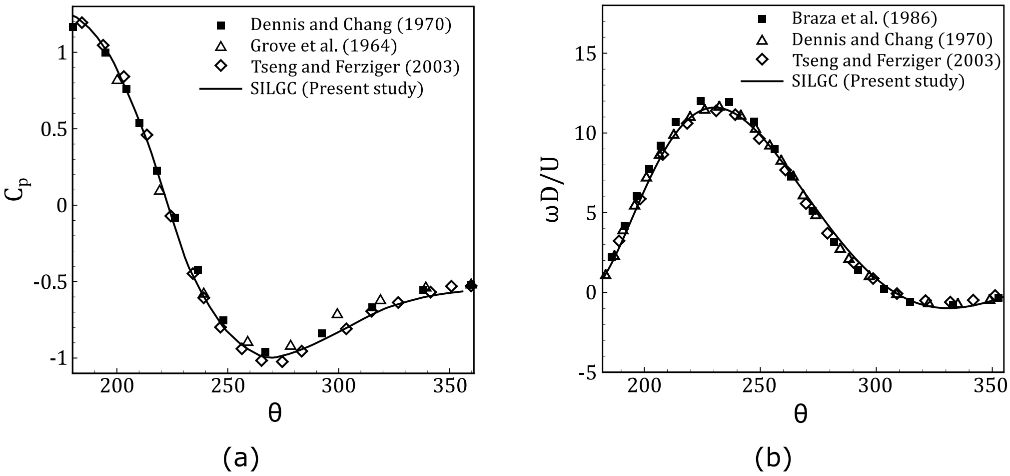

Figure 6: Pressure (C_p) and vorticity ($\omega D/U$) coefficients distribution on the cylinder surface at $Re = 40$.

Table 2: Comparison of mean drag coefficient (C_D) with those in the literature.

Re	SILGC	Berthelsen et al. (2007)	Yee et al. (1999)	Triton (1959)
10	2.68	2.78	-	-
20	2.04	2.01	2.03	2.22
40	1.53	1.51	1.52	1.48
80	1.38	1.35	1.37	1.29
100	1.34	1.33	-	-
300	1.24	1.37	1.38	-

Figure 7 shows the streamlines neighboring the flat plate and the evolution of the separation bubble behind the plate at two different time-steps. The bubble increases with time and does not reach a steady flow.

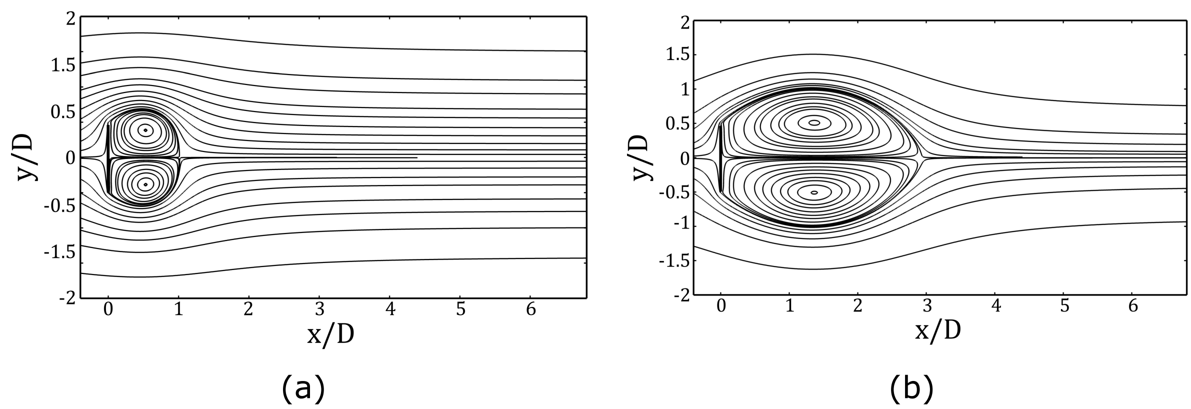

Figure 7: Instantaneous streamlines showing evolution of the separation bubble at non-dimensional times tU/D (a) 1.2 and (b) 7.5.

Figure 8 presents a comparison of the pressure coefficient distribution on the plate surface found with the method proposed here and the numerical results of Berthelsen et al. (2007) at $tU/D = 1.0$. The SILGC has a good agreement with the results of Berthelsen et al. (2007),

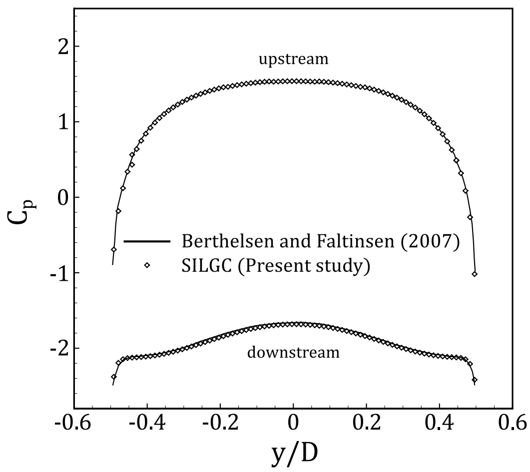

Figure 8: Distribution of pressure coefficient (C_p) along the flat plate.

which applied the same method explicitly using a second order Runge–Kutta method.

4.3 Two parallel laminar channel flow

An other example that illustrates the importance of the pressure and velocity decoupling constraint in the simulation of flow past thin body is the case shown in Fig. 9. Two parallel channels (also known as Poiseuille flow) aligned in the x-direction with different heights, separated by a very thin rigid interface (notionally with zero thickness) with steady laminar flows in the same direction.

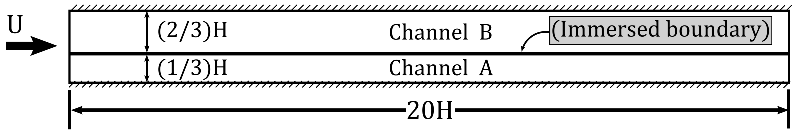

Figure 9: Schematics of the two parallel plane channels with different height.

The lower channel (channel A) has a height of $(1/3)H$ and a normalized pressure difference of $2\Delta P/U^2\rho = 2000$ between the inlet and outlet faces, and the upper channel (channel B) has a height of $(2/3)H$ with $2\Delta P/U^2\rho = 1200$. A characteristic mean velocity $U = 1$ is used for the normalization, H is the total (sum) height of the channels and the length of the domain is $20H$. Both the decay rate of the pressure in the streamwise direction and the pressure at the inlet face are different in the two channels in order to create a greater discontinuity. This case seeks to create a flow with discontinuities in the velocity and pressure fields, and thus the ability to treat cases involving thin bodies is tested. The results will be compared with the exact solution given by Eq. 32:

$$u(y) = -\frac{1}{2}\frac{h^2}{\mu}\frac{dP}{dx}\left[1 - \left(\frac{y}{h}\right)^2\right], \tag{32}$$

where the $h = 2H$ is the half-height of sum of the heights of the two channels and the pressure gradient is considered constant.

Since the pressure and velocity are different in each channel, their profile normal to the interface shows a discontinuity across the thin wall, as seen in Fig. 10. One can see that the

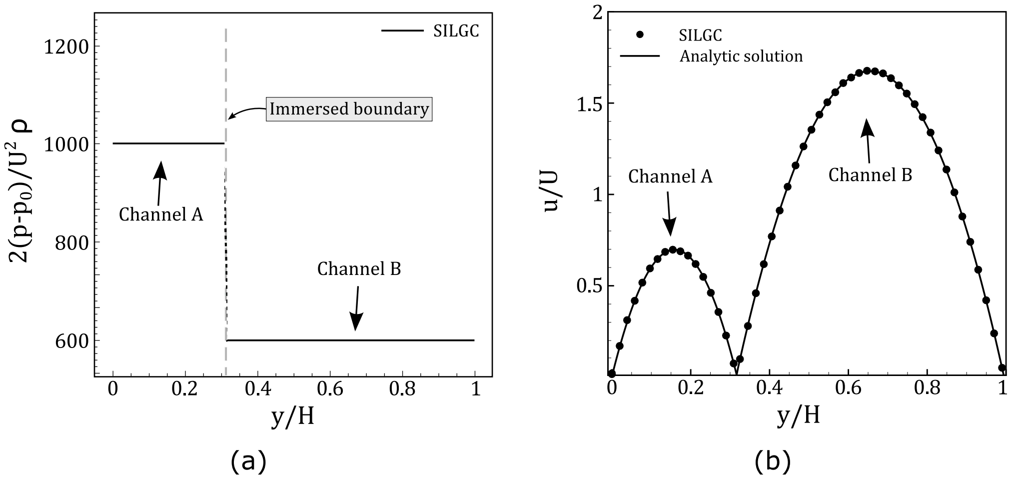

Figure 10: Profiles at $x/H = 10$ of the (a) normalized pressure coefficient $2(p - p_0)/U^2\rho$ and (b) normalized horizontal velocity u/U compared with the exact solution.

semi-implicit local ghost cell method maintains a discontinuous pressure profile (Fig. 10-a) and a parabolic x-velocity profile (Fig. 10-b) across the thin wall and has a good agreement with the exact solution. The pressure decoupling constraint is satisfied, under the condition that the derivation of the velocity in one domain is determined exclusively from the velocity field in that fluid region and does not create a pressure and velocity link between both channel domains.

5 CONCLUSION

A IBM for imposing the boundary conditions on the solid–fluid interface, namely the semi-implicit local ghost cell method, were presented and tested to calculate the uniform flow over immersed bodies with a special focus on sharp and thin IB. In order to verify the presented IBM, different benchmark cases have been tested: the laminar flow past a circular cylinder, the flow past a circular cylinder between parallel walls, an impulsively started flow past a flat plate and a separated plane channel flow. Flow parameters including drag and lift coefficients, Strouhal number, wall pressure, velocity streamlines, and the length of the bubble recirculation, have been presented for several values of the Reynolds number.

The SILGC, a semi-implicit IBM approach, proposed in the present paper, showed good results when dealing with a flow over blunt bodies, thin plane and sharp edges. This behavior can be explained by the fact that its force distribution does not smear the immersed boundary over several grid cells as diffuse methods and then, deals properly with sharp geometries.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the financial technical support from the Petróleo Brasileiro S.A. (Petrobras) and the Brazilian funding agencies Coordenadoria de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) and the School of Mechanical Engineering of the Federal University of Uberlândia (FEMEC).

REFERENCES

Peskin, C., 1977. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, vol. 25, pp. 220–252.

Ji, C., Munjiza, A. & Williams, J. J. R., 2012. A novel iterative direct-forcing immersed boundary method and its finite volume applications. *Journal of Computational Physics*, vol. 231, pp. 1797–1821.

Mittal, R. & Iaccarino, G., 2005. Immersed boundary methods. *Annual Reviews of Fluid Mechanics*, vol. 37, pp. 239–261.

Peskin, C., 1972. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, vol. 10, pp. 252–271.

Cai, S. G., Ouahsine, A., Favier, J. & Hoarau, Y., 2017. Moving immersed boundary method. *International Journal for Numerical Methods in Fluids*, pp. 1–36.

Chi, C., Lee, B. J. & Im, H. G., 2017. An improved ghost-cell immersed boundary method for compressible flow simulations. *International Journal for Numerical Methods in Fluids*, vol. 83, pp. 132–148.

Tyson, R., Jordan, C. E. & Hebert, J., 2008. Modelling anguilliform swimming at intermediate Reynolds number: a review and a novel extension of immersed boundary method applications. *Computational Methods for Applied Mechanical Engineering*, vol. 197, pp. 2105–2118.

Kim, Y. & Peskin, C. S., 2009. 3-d parachute simulation by the immersed boundary method. *Computers & Fluids*, vol. 38, pp. 1080–1090.

Uhlmann, M., 2005. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, vol. 209, pp. 448–476.

Tseng, Y. H. & Ferziger, J. H., 2003. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, vol. 192, pp. 593–623.

Gao, T., Tseng, Y. H. & Lu, X. Y., 2007. An improved hybrid Cartesian/immersed boundary method for fluid–solid flows. *International Journal for Numerical Methods in Fluids*, vol. 55, pp. 1189–1211.

Ilinca, F. & Hétu, J. F., 2011, A finite element immersed boundary method for fluid flow around rigid objects. *International Journal for Numerical Methods in Fluids*, vol. 65, pp. 856–875.

Boffi, D. & Gastaldi, L., 2003. A finite element approach for the immersed boundary method. *Computers and Structures*, vol. 81, pp. 491–501.

Cheng, Y. & Zhang, H., 2010. Immersed boundary method and lattice Boltzmann method coupled FSI simulation of mitral leaflet flow. *Computers & Fluids*, vol. 39, pp. 871–881.

Feng, Z. G. & Michaelides, E. E., 2004. The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. *Journal of Computational Physics*, vol. 195, 602–628.

Kinoshita, D., da Silveira Neto, A., Mariano, F. P., da Silva, R. A. P. & Serfaty, R., 2016, A novel immersed boundary/Fourier pseudospectral method for flows with thermal effects. *Numerical Heat Transfer, Part B: Fundamentals*, vol. 69, pp. 1–22.

Berthelsen, P. A. & Faltinsen, O. M., 2007. A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries. *Journal of Computational Physics*, vol. 227, pp. 4354–4397.

Udaykumar, H. S., Mittal, R., Rampunggoon, P. & Khanna, A., 2001. A sharp interface Cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics*, vol. 174, pp. 345–380.

De Zeeuw, D. & Powell, K. G., 1993. An adaptively refined Cartesian mesh solver for the Euler equations. *Journal of Computational Physics*, vol. 104, pp. 56–68.

Meinke, M., Schneiders, L., Günther, C. & Schroder, W., 2013. A cut–cell method for sharp moving boundaries in Cartesian grids. *Computers & Fluids*, vol. 85, pp. 135–142.

Quirk, J. J., 1994. An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies. *Computers & Fluids*, vol. 23, 125–142.

Tucker, P. G. & Pan, Z., 1999. A Cartesian cut cell method for incompressible viscous flow. *AIAA Journal*, vol. 37, pp. 905–911.

Yang, G., Causon, D. M. & Ingram, D. M., 1999. Cartesian cut-cell method for axisymmetric separating body flows. *AIAA Journal*, vol. 37, pp. 905–911.

Yee, T., Mittal, R., Udaykumar, H. & Shyy, W., 1999. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*, vol. 156, pp. 209–240.

Berthelsen, P. A., 2004. A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions. *Journal of Computational Physics*, vol. 197, pp. 364–386.

LeVeque, R. J. & Zhilin, L., 1997, Immersed interface methods for stokes flow with elastic boundaries or surface tension. *SIAM Journal on Scientific Computing*, vol. 18, pp. 709–735.

Li, Z. & Lai, M. C., 2001. The immersed interface method for the Navier–Stokes equations with singular forces. *Journal of Computational Physics*, vol. 171, pp. 822–822.

Wiegmann, A. & Bube, K. P., 2000. The explicit-jump immersed interface method: Finite difference methods for PDEs with piecewise smooth solutions. *SIAM Journal on Numerical Analysis*, vol. 37, pp. 827–862.

Kim, J. & Moin, P., 1985. Application of a fractional step method to incompressible Navier–Stokes equations. *Journal of Computational Physics*, vol. 59, pp. 308.

van Kan, J., 1986. A second-order accurate pressure correction scheme for incompressible viscous flow. *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 870–891.

Ferziger, J. & Perić, M., 1996. Computational Methods for Fluid Dynamics, Springer,

Orlanski, I., 1976. A simple boundary condition for unbounded hyperbolic flows. *Journal of Computational Physics*, vol. 21, pp. 251–269.

Mittal, R., Dong, H., Bozkurttas, M., Najjar, F. M., Vargas, A. & von Loebbecke, A., 2008. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, vol. 227, pp. 4825–4852.

Majumdar, S., Iaccarino, G. & Durbin, P., 2001. RANS solvers with adaptive structured boundary non-conforming grid. *Annual Research Briefs, NASA Ames Research Center/Stanford University Center for Turbulence Research*, Stanford, CA, pp. 353–366, 0000.

Linnick, M. N. & Fasel, H. F., 2005. A high-order immersed interface method for simulating unsteady compressible flows on irregular domains. *Journal of Computational Physics*, vol. 204, pp. 157–192.

Lai, M. C. & Peskin, C. S., 2000. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, vol. 160, pp. 705–719.

Coutanceau, M. & Bouard, R., 1977. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation part 1: Steady flow. *Journal of Fluid Mechanics*, vol. 79, pp. 231–256.

Calhoun, D., 2002. A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions. *Journal of Computational Physics*, vol. 176, 231–275.

Le, D. V, C. Khoo, B. & Peraire, J., 2006. An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries. *Journal of Computational Physics*, vol. 220, pp. 109–138.

Russel, D. & Wang, Z. J., 2003. A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow. *Journal of Computational Physics*, vol. 191, pp. 177–205.

Xu, S. & Wang, Z. J., 2006. An immersed interface method for simulating the interaction of a fluid with moving boundaries. *Journal of Computational Physics*, vol. 216, pp. 454–493.

Nishioka, M. & Sato, H., 1978. Mechanism of determination of the shedding frequency of vortices behind a cylinder at low Reynolds numbers. *Journal of Fluid Mechanics*, vol. 89, pp. 49–60.

Park, J., Kwon, K. & Choi, H., 1998. Numerical solutions of flow past a circular cylinder at Reynolds number up to 160. *KSME International Journal*, vol. 12, pp. 1200–1205.

Roshko, A., 1955. On the wake and drag of bluff bodies. *Journal of the Aeronautical Sciences*, vol. 22, pp. 124–132.

Williamson, C. H., 1996. Vortex dynamics in the cylinder wake. *Annual Reviews of Fluid Mechanics*, vol. 28, pp. 477–539.

Triton, D. J., 1959. Experiments on the flow past a circular cylinder at low Reynolds number. *Journal of Fluid Mechanics*, vol. 6, pp. 547.

Singha, S. & Sinhamahapatra, K. P., 2010. Flow past a circular cylinder between parallel walls at low Reynolds number. *Ocean Engineering*, vol. 37, pp. 757–769.

Koumoutsakos, P. & Shiels, D., 1996. Simulations of the viscous flow normal to an impulsively started and uniformly accelerated flat plate. *Journal of Fluid Mechanics*, vol. 328, pp. 177–227.

Taneda, S. & Honji, H., 1971. Unsteady flow past a flat plate normal to the direction of motion. *Journal of the Physical Society of Japan*, vol. 30, pp. 262–272.