

Journal Pre-proof

A directional ghost-cell immersed boundary method for incompressible flows

Cheng Chi, Abouelmagd Abdelsamie, Dominique Thévenin

PII: S0021-9991(19)30827-7

DOI: <https://doi.org/10.1016/j.jcp.2019.109122>

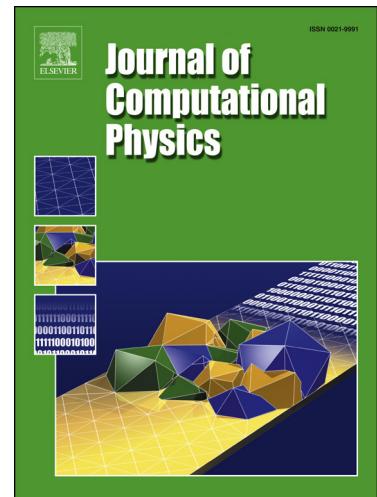
Reference: YJCPH 109122

To appear in: *Journal of Computational Physics*

Received date: 17 August 2018

Revised date: 6 September 2019

Accepted date: 10 November 2019



Please cite this article as: C. Chi et al., A directional ghost-cell immersed boundary method for incompressible flows, *J. Comput. Phys.* (2019), 109122, doi: <https://doi.org/10.1016/j.jcp.2019.109122>.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2019 Published by Elsevier.

A directional ghost-cell immersed boundary method for incompressible flows

Cheng Chi^{a,b,*}, Abouelmagd Abdelsamie^{a,c}, Dominique Thévenin^a

^a*Lab. of Fluid Dynamics and Technical Flows, University of Magdeburg “Otto von Guericke”, Germany*

^b*International Max Planck Research School (IMPRS) for Advanced Methods in Process and Systems Engineering, Magdeburg, Germany*

^c*Lab. of Fluid Mechanics, Faculty of Engineering, Helwan University, Cairo, Egypt*

Abstract

This paper describes an efficient ghost-cell immersed boundary method for incompressible flow simulations. The method employs a locally directional extrapolation scheme along all discretization directions for the ghost values. Additionally, it involves fictitious discrete boundary forcing terms instead of the ghost values in the governing equations. In this way, the boundary is represented more accurately than in prior IBMs and it is possible to fulfill the divergence-free condition. When combined with high-order spatial discretization schemes, the IBM order is reduced locally near the immersed boundary in a step-wise manner. In this way, the method delivers more compact stencils and is able to deal with sharp interfaces. By handling the distance from the ghost point to the boundary carefully, the proposed method delivers lower truncation errors than standard IBM, with clean (persistent) convergence rate and enhanced stability. The parallel implementation of this

*Corresponding author

Email address: cheng.chi@ovgu.de (Cheng Chi)

approach is straightforward. Its accuracy has been checked by considering a variety of test cases, including irregular, three-dimensional, and moving boundaries. The local accuracy for the proposed method is formally second order, and is measured to be close to this value using numerical tests.

Keywords: incompressible flow, complex geometry, immersed boundary, ghost-cell method

1. Introduction

For computational fluid dynamics (CFD) simulations with complex geometries, Cartesian grids show many advantages over body-fitted methods, in particular easy mesh generation, memory and CPU savings, easy parallelization and possible benefit from adaptive mesh refinement. These advantages open the door for high-fidelity CFD simulations, explaining the numerous investigations on Cartesian grids in recent years.

As one of the most popular Cartesian grid methods, the immersed boundary method (IBM) has been introduced at the turn of the century [1, 2] and has been considerably extended since then [3, 4]. The underlying physics of IBM is to enforce the boundary conditions by modifying the momentum forces near the solid boundary, either continuously or discretely. The immersed boundary method is thus categorized as continuous forcing approach or discrete forcing approach [3, 5].

Compared to the continuous forcing approach [6, 7], where the forcing function is computed on the boundary and then spread over the vicinity of the immersed surface [8], the discrete forcing approach is more flexible and allows for a sharper representation of the immersed boundary [9]. In earlier discrete

forcing methods [10, 11, 12], the discrete momentum forcing was computed on the body surface and inside the body; then, an explicit mass source/sink was required to maintain the conservation of mass near the boundary. To improve this point, an alternative discrete forcing method implicitly involves 5 the momentum forcing in the discrete spatial operators, by introducing ghost cells. This approach is called ghost-cell immersed boundary method, and is considered in the present work.

Ghost-cell IBM was first introduced in [13, 14]. It has been later widely used and extended, see for instance [9, 15, 16]. Compared to other immersed 10 boundary methods, such as cut-cell methods [17, 18, 19, 20], ghost-cell IBM is more efficient and easier to implement in parallel, while showing comparable accuracy [21]. The ghost cell is identified as the grid cell (for a finite-volume discretization) or grid point (for a finite-difference discretization) in the solid phase near the solid boundary with at least one neighbor grid cell (respectively point) in the fluid. The conventional idea for ghost-cell IBM is to impose implicitly the boundary condition at the interface by including the variable values at the ghost cells in the governing equations. The ghost cell values are usually extrapolated from the values at the nearby fluid cells and at the boundary points.

20 As the boundary interface is only implicitly enforced, the accuracy of the ghost-cell IBM is a critical issue. Tseng et al. [15] computed the values of the ghost cells with a quadratic extrapolation and obtained second-order accuracy for their ghost-cell IBM. Gao et al. [22] used a second-order Taylor series expansion for the ghost values, later eliminating numerical instabilities 25 by matrix inversion [15]. Then, Mittal et al. [9] and Ghias et al. [16]

used well-conditioned extrapolation/interpolation stencils for the values of the variables at the ghost cells and also obtained second-order accuracy. Pan et al. [23] and Chi et al. [21] introduced a simplified but still stable extrapolation/interpolation scheme involving a farther image point for the 5 ghost cell values, delivering second-order accuracy for L_2 and L_1 norms.

All the previous studies tried to improve the accuracy of the ghost-cell IBM by introducing higher-order extrapolation/interpolation schemes for the ghost cell values. However, the convergence rate of the truncation error with the conventional ghost-cell IBM is not persistent, for reasons that we will 10 describe in Section 2.3. Additionally, high-order extrapolation/interpolation methods are always sensitive to numerical instabilities [24]. Finally, for IBM coupled with a finite-difference discretization, as considered in the present work, high-order methods often lead to discrete conservation issues [12, 16, 20, 25, 26]. To solve these problems regarding convergence, stability, and 15 conservation, an improved ghost-cell IBM is introduced in the present work.

First, some limitations of the conventional ghost-cell IBM are analyzed. Then, a stable, second-order ghost-cell IBM is introduced and implemented in a finite-difference solver. The method involves multiple ghost values at the ghost cells, one for each discretization direction. The developed approach 20 is compatible with different discretization schemes in the interior of the fluid domain. In the present case, the IBM is implemented in a high-order finite-difference solver; the order is reduced progressively near the immersed boundary. In contrast to conventional ghost-cell methods, where ghost-cell variables are involved in the Navier-Stokes equations and boundary forces 25 are implicitly enforced, the present approach involves instead explicit dis-

crete forces near the boundary. The fictitious ghost cell values are only used to compute these discrete forces and are not directly considered in the continuity equation any more, resulting in improved mass conservation at the boundary. The present approach leads to a straightforward parallel implementation, and the following test cases have been computed in parallel.

The paper is organized as follows: Section 2 introduces the governing equations and the numerical methods, including discretization. In a second step, limitations of the conventional ghost-cell methods and proposed improvements are discussed. Then, simulation results from various test cases are shown in Section 3, followed by the final conclusion in Section 4.

2. Formulation and Numerical Method

In this part, the governing equations for incompressible flows and the numerical method are explained. The main drawbacks of the conventional ghost-cell immersed boundary method are analyzed. In an effort to solve these issues, a novel ghost-cell IBM is proposed.

2.1. Governing equations

The incompressible formulation of the Navier-Stokes equations is considered in this study. The conservation equations for mass and momentum are thus written as:

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} - \nu \nabla^2 \mathbf{u} + \nabla P \mathbf{I} = \mathbf{f} \quad (2)$$

Here, \mathbf{f} is the boundary forcing term, P is the pressure divided by (constant) density ρ , $\nu = \mu/\rho$ is the kinematic viscosity with μ the dynamic viscosity, and \mathbf{u} is the velocity vector with components u , v and w , i.e. $\mathbf{u} = u\mathbf{i} + v\mathbf{j} + w\mathbf{k}$.

⁵ *2.2. Discretization scheme*

Following the numerical schemes in the in-house direct numerical simulation (DNS) solver DINO [27], the momentum equation (Eq. 2) is discretized spatially using a centered, sixth-order finite difference scheme, progressively reduced down to third order near real domain boundaries (meaning here the ¹⁰ actual boundaries of the numerical domain, and not the additional immersed boundaries). For the example shown in what follows, the iteration in time uses a fully explicit, low-storage fourth-order Runge-Kutta method [28]. As explained in [27], implicit solvers are available as well, and could be also combined with the proposed IBM approach after minor changes.

¹⁵ For pressure/velocity coupling, the fractional step reads

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t \mathbf{R}^n, \quad (3)$$

$$\nabla^2 P^{n+1/2} = \frac{1}{\Delta t} \nabla \mathbf{u}^*, \quad (4)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla P^{n+1/2}, \quad (5)$$

with \mathbf{R} the right-hand side term including convective term, diffusive term, and the boundary forcing term

$$\mathbf{R} = -(\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} + \mathbf{f}. \quad (6)$$

Here, \mathbf{u}^* is the intermediate velocity. In this fractional step, the continuity and momentum equation are coupled together using the pressure-free projection method [29, 30]. The continuity constraint is ensured in Eq. (4), which is obtained from Eq. (5) by applying the divergence operator.

5 2.3. Conventional ghost-cell IBM

In the conventional ghost-cell immersed boundary method the boundary forcing term \mathbf{f} is implemented implicitly in the momentum equation in order to enforce the boundary condition at each time step [15]. Hence, \mathbf{f} is not evaluated explicitly and is set to 0 in the computation. Instead, the boundary 10 forcing is implicitly coupled to the convective and diffusive terms in Eq. (6) and to the pressure gradient term in Eq. (5). By extrapolating the velocity and pressure at the ghost cells from the nearby fluid cells, and incorporating the boundary conditions, the convective, diffusive and pressure gradient terms at the fluid points near the boundary are modified. Equations (3) and 15 (5) are transformed into:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \mathbf{R}^n + (\mathbf{R}^n|_{ghost} - \mathbf{R}^n), \quad (7)$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\nabla P^{n+1/2} + (\nabla P^{n+1/2} - \nabla P^{n+1/2}|_{ghost}), \quad (8)$$

where \mathbf{R}^n and $\nabla P^{n+1/2}$ are the terms computed when assuming that there is no immersed boundary in the computational domain, while $\mathbf{R}^n|_{ghost}$ and $\nabla P^{n+1/2}|_{ghost}$ are the terms obtained when the values at the ghost cells 20 are included in the computations. Thus, $(\mathbf{R}^n|_{ghost} - \mathbf{R}^n)$ and $(\nabla P^{n+1/2} - \nabla P^{n+1/2}|_{ghost})$ represent the action of the boundary forces on the fluid points. If $(\mathbf{R}^n|_{ghost} - \mathbf{R}^n)$ and $(\nabla P^{n+1/2} - \nabla P^{n+1/2}|_{ghost})$ are evaluated at the fluid

points which are far away from the immersed boundary, these two values are equal to 0, recovering the standard solution. The range of influence of the immersed boundary at a fixed time step is determined by the discretization scheme employed to compute the partial derivatives.

- 5 In the present study, a sixth-order central difference scheme is used to discretize the spatial derivatives in the internal fluid domain. This high-order approach is particularly useful for simple turbulence studies, for instance in a periodic box. But it must be compatible with the developed IBM method.
- 10 The discretization scheme (denoted here as $f_c(a, b, n)$ where a is the discretized variable, b is the discretization direction, and n is the order of the derivative) is as follows:

$$f_c(u_i, x, 1) = \frac{\partial u_i}{\partial x} = \frac{1}{60} (u_{i+3} - u_{i-3} - 9(u_{i+2} - u_{i-2}) + 45(u_{i+1} - u_{i-1})) / \Delta x; \quad (9)$$

$$f_c(u_i, x, 2) = \frac{\partial^2 u_i}{\partial x^2} = \frac{1}{180} (2(u_{i+3} + u_{i-3}) - 27(u_{i+2} + u_{i-2}) + 270(u_{i+1} + u_{i-1}) - 490u_i) / \Delta x^2. \quad (10)$$

- As there are 3 points involved on either side of the discretization point i , this specific discretization involves three layers of fluid points near the immersed boundary, as shown in Fig. 1. As a consequence, three layers of ghost points are necessary to get correct derivatives at the fluid points near the immersed boundary. For instance, for the 1D case shown exemplarily in Fig. 1, the values at ghost points $i + 1$, $i + 2$ and $i + 3$ are extrapolated from the fluid points and the boundary point P in the conventional ghost-cell IBM. The boundary force is discretely and implicitly applied at the fluid points i , $i - 1$ and $i - 2$.

In the conventional approach, the extrapolation for the ghost values takes

place along the direction normal to the solid boundary, leading to the image point (IP) shown in Fig. 2. This procedure is straightforward and has been used to develop different high-order reconstruction schemes. However, some problems concerning accuracy, stability, and conservation are often encountered using the conventional approach. The main bottlenecks of the

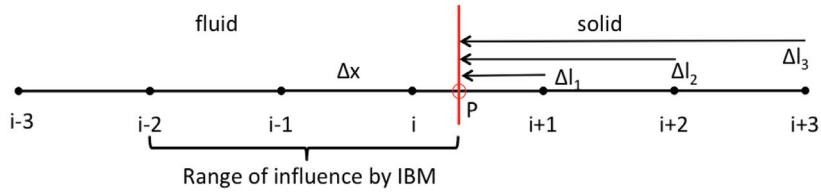


FIGURE 1. Conventional ghost-cell IBM: sketch for a one-dimensional case.

5

conventional ghost-cell IBM are as follows:

Drawback 1: In previous studies, different attempts have been documented concerning a higher order computation of the ghost cell values, in order to improve the order of accuracy of the IBM representation. However,
10 most of those computations rely on the assumption that the order of Δl is of the same order as the grid size Δx , i.e. $\Delta l = O(\Delta x)$, where Δl is the distance from the ghost point to the boundary [5, 9, 22, 31]. Since Δl is not linearly related to Δx , the error term based on $O(\Delta l)$ does not necessarily converge like Δx . The convergence rate of the truncation error is fluctuating and not
15 persistent. It is believed that the error term based on $O(\Delta l)$ will statistically converge at the desired order. However, to ensure a clean (persistent) convergence rate is still important, especially when a coarser simulation result is not good enough and a better result is expected by refining the mesh resolution.

Drawback 2: For multi-dimensional simulations, as explained in Fig. 2, the intersection point P with the solid body along the normal direction is the implicitly enforced boundary point when extrapolating the values at the ghost point $(i+1, j)$. However, the points naturally relevant for the employed numerical discretization stencil are the boundary points X and Y (Fig. 2); those are not enforced at all.

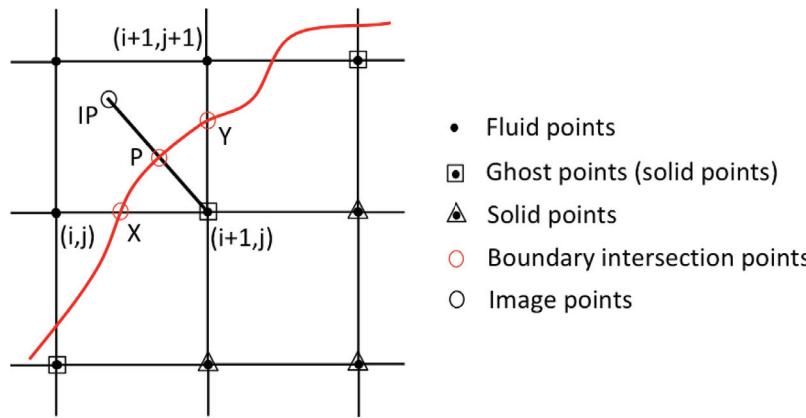


FIGURE 2. Conventional ghost-cell IBM: sketch for a two-dimensional case.

Drawback 3: When considering high-order spatial discretization schemes in the fluid solver, as is here the case, multiple layers of ghost points become necessary. As shown in Eqs. (9) and (10), to compute the spatial derivatives at fluid point i , the ghost values at points $i+1$, $i+2$ and $i+3$ are required, which means that the distances Δl_1 , Δl_2 and Δl_3 must be taken into account simultaneously to enforce the boundary conditions. Looking back at Drawback 1, the relationship between Δl_1 , Δl_2 , Δl_3 , and Δx is even more complex. Practically, this leads to a complex implementation. Multiple layers of ghost points hinder parallelization efficiency and make it more difficult

to accurately capture the boundary layer at sharp corners.

2.4. A novel ghost-cell IBM implementation

In order to release the previously discussed bottlenecks, an improved ghost-cell IBM method is now introduced.

5 Drawback 2 is related to the representation of the immersed boundary, impacting the magnitude of the absolute error. Drawbacks 1 and 3 concern the convergence rate of the truncation error, determining the order of accuracy of the ghost-cell IBM. The magnitude of the absolute error is most important for DNS at higher Reynolds numbers, so that Drawback 2 is considered first.

10 In conventional methods, for instance in a two-dimensional case (Fig. 2), the values computed at the ghost point $(i+1, j)$ will be used in both x and y directions for reconstructing the fictitious boundary force at fluid points (i, j) and $(i+1, j+1)$, respectively. Following the previous discussion, the fictitious 15 momentum force at point (i, j) in x direction and at point $(i+1, j+1)$ in y direction can be formulated from Eq. (6) and Eq. (7) respectively as:

$$f_x(i, j) = - \left(\frac{\partial(u_{i,j}u_{i,j})}{\partial x} + \frac{\partial(u_{i,j}v_{i,j})}{\partial y} \right)_{ghost} + \nu \left(\frac{\partial^2 u_{i,j}}{\partial y^2} + \frac{\partial^2 u_{i,j}}{\partial x^2} \right)_{ghost} \quad (11) \\ + \left(\frac{\partial(u_{i,j}u_{i,j})}{\partial x} + \frac{\partial(u_{i,j}v_{i,j})}{\partial y} \right) - \nu \left(\frac{\partial^2 u_{i,j}}{\partial y^2} + \frac{\partial^2 u_{i,j}}{\partial x^2} \right),$$

$$\begin{aligned}
f_y(i+1, j+1) = & - \left(\frac{\partial(v_{i+1,j+1} u_{i+1,j+1})}{\partial x} + \frac{\partial(v_{i+1,j+1} v_{i+1,j+1})}{\partial y} \right)_{ghost} \\
& + \nu \left(\frac{\partial^2 v_{i+1,j+1}}{\partial y^2} + \frac{\partial^2 v_{i+1,j+1}}{\partial x^2} \right)_{ghost} \\
& + \left(\frac{\partial(v_{i+1,j+1} u_{i+1,j+1})}{\partial x} + \frac{\partial(v_{i+1,j+1} v_{i+1,j+1})}{\partial y} \right) \\
& - \nu \left(\frac{\partial^2 v_{i+1,j+1}}{\partial y^2} + \frac{\partial^2 v_{i+1,j+1}}{\partial x^2} \right).
\end{aligned} \tag{12}$$

In Eq. (11) and Eq. (12), the partial derivatives with notation "ghost" are functions of the variables at the nearby fluid points and ghost points, depending on the employed discretization schemes (DCT):

$$\left(\frac{\partial \mathbf{u}_{i,j}}{\partial x} \right)_{ghost} = \text{DCT}(\mathbf{u}_{i+1,j}, \mathbf{u}_{i,j}, \mathbf{u}_{i-1,j} \dots), \tag{13}$$

5

$$\left(\frac{\partial \mathbf{u}_{i+1,j+1}}{\partial y} \right)_{ghost} = \text{DCT}(\mathbf{u}_{i+1,j}, \mathbf{u}_{i+1,j+1}, \mathbf{u}_{i+1,j+2} \dots). \tag{14}$$

To implicitly enforce the immersed boundary condition and get a proper boundary force, most conventional methods compute the ghost values $\mathbf{u}_{i+1,j}$ by bilinear interpolation using the surrounding fluid points or boundary interception points. It is obvious that the two boundary points X and Y cannot
10 be accurately represented simultaneously by using just one ghost value, no matter which reconstruction method is used for the ghost values. Therefore, two different sets of independent ghost values ($\mathbf{u}_{i+1,j}|_x$ and $\mathbf{u}_{i+1,j}|_y$) are necessary at the same ghost point $(i+1, j)$ in order to fulfill the boundary conditions in both computational directions; similarly, three ghost values are
15 employed for a 3D problem. Then, Equations (13) and (14) become

$$\left(\frac{\partial \mathbf{u}_{i,j}}{\partial x} \right)_{ghost} = \text{DCT}(\mathbf{u}_{i+1,j}|_x, \mathbf{u}_{i,j}, \mathbf{u}_{i-1,j} \dots), \tag{15}$$

$$\left(\frac{\partial \mathbf{u}_{i+1,j+1}}{\partial y} \right)_{ghost} = \text{DCT}(\mathbf{u}_{i+1,j}|_y, \mathbf{u}_{i+1,j+1}, \mathbf{u}_{i+1,j+2} \dots). \quad (16)$$

In this manner, both fictitious momentum forces $f_x(i, j)$ and $f_y(i + 1, j + 1)$ are computed correctly and account for the presence of the two boundary points X and Y simultaneously. In this way, the boundary is represented
 5 more accurately from the numerical point of view. A similar treatment has been proposed for incompressible viscous flows in [24] and for compressible Euler flows in [32]. These different ghost values are only used to reconstruct the fictitious boundary force in a discrete sense, and will not be used in the continuity equation.

10 For multi-dimensional simulations, the ghost values are now computed along the discretization directions instead of the boundary-normal direction. One disadvantage of this treatment would be for slip-wall boundary conditions, where curvature information of the boundary at the point where it is intersected by the grid is required. However, all the cases considered in
 15 what follows and in our research involve no-slip wall boundary conditions, leading to a straightforward computation of the ghost values along the grid directions. The algorithm for extrapolating the ghost values is introduced later.

Concerning now drawback 3, reducing progressively the discretization scheme near the immersed boundary reduces the number of required layers of ghost points. The same is already done at the real boundaries. In the present ghost-cell IBM, the original sixth-order finite-difference algorithm is reduced step-by-step to 1) fifth order at the second layer of fluid points near the immersed boundary, and 2) fourth order at the first layer, leading

to following equations for the first derivatives in the one-dimensional case exemplified in Fig. 1:

$$\frac{\partial u_{i-1}}{\partial x}|_{5th} = \frac{1}{60} (2(u_{i-3} - u_{i-4}) - 13(u_{i-2} - u_{i-3}) + 20(u_{i-1} - u_{i-2}) + 27(u_i - u_{i-2}) - 3(u_{i+1} - u_i)) / \Delta x; \quad (17)$$

$$\frac{\partial u_i}{\partial x}|_{4th} = \frac{1}{12} ((u_{i+1} - u_{i-3}) + 2(u_{i+1} - u_{i-1}) + 10(u_i - u_{i-1}) - 6(u_{i-1} - u_{i-2})) / \Delta x. \quad (18)$$

Similar to the original discretization function f_c , the above discretization with a step-wise order reduction is denoted as f_b . The employed stencils are no longer centered (as was the case for the sixth-order finite difference), but are forward/backward finite-difference schemes depending on the position of

- 5 the immersed boundary. Therefore, only one layer of ghost points is finally needed to reconstruct the boundary forces. This treatment is not specific for the sixth-order central finite difference scheme employed in this study; it can be easily adjusted for other high-order finite-difference algorithms.

Provided everything has been implemented correctly, the truncation error
 10 would converge at fourth-order with the grid size. The final order of accuracy will be limited by the accuracy of the computation of the ghost point values. The truncation errors in the computations of the ghost point values involve both Δl and Δx . As there is no direct linear relationship between Δl and Δx , as discussed previously, the persistent convergence rate of the truncation
 15 error can be ensured if Δl is treated properly.

In the current ghost-cell IBM implementation, a third-order extrapolation scheme based on a Taylor series expansion on a regular grid is applied for

the ghost point velocities,

$$\begin{aligned} u_{i+1} = & u_P + \Delta l \left(\frac{u_{i-1} - u_P}{2\Delta x - \Delta l} - \frac{2(u_i - u_P)}{\Delta x - \Delta l} \right) + O(\Delta l \Delta x^2) \\ & + O(\Delta x^4) + O\left(\frac{\Delta x^5}{2\Delta x - \Delta l}\right) + O\left(\frac{\Delta x^2(\Delta x - \Delta l)^3}{2\Delta x - \Delta l}\right), \end{aligned} \quad (19)$$

where u_P denotes the corresponding fluid variable at the boundary. This solution is equivalent to quadratic polynomial interpolation and is retained to eliminate the error terms which are dominated by $O(\Delta l^n \Delta x^m)$, with $n \geq 1$
⁵ and $m \leq 1$.

From Eq. (19), it is clear that the total truncation error is dominated by $O(\Delta l \Delta x^2)$, which should converge in a third-order manner with grid size Δx (at least second-order persistent convergence rate).

To avoid stability issue of the above extrapolation scheme when the fluid
¹⁰ point is too close to the boundary (i.e., when $\Delta x - \Delta l$ is approaching 0), the higher order scheme is reduced down to a lower order when $1 - \frac{\Delta l}{\Delta x} < r$. In that case,

$$u_{i+1} = u_P + \Delta l \frac{u_P - u_{i-1}}{2\Delta x - \Delta l}. \quad (20)$$

Here, r is the stability threshold. Based on our experience with the test cases shown in what follows, it is recommended to choose r between 0.1 and 0.2,
¹⁵ the solution being more stable for a higher value of r but more accurate for a lower value of r . In the examples shown in the next section, $r = 0.2$ has been systematically retained. The 1st order fallback scheme described by Eq. (20) will inevitably impact in a negative manner the overall order of the proposed IBM method whenever it is activated. This is the reason why the
²⁰ formally second-order approach will deliver a real overall order lower than two (especially for coarser grids) in the numerical tests discussed in what follows.

As suggested by one reviewer, an alternative way to solve the above stability issue would be to shift the stencil in Eq. (19) by Δx if $(\Delta x - \Delta l) < \frac{\Delta x}{2}$. This approach would keep the order. However, it would also involve farther fluid points in the stencil, which may be a problem, for instance for very thin boundary layers.

For conventional ghost-cell IBM, a Neumann boundary condition for pressure is usually enforced [9, 14], resulting in zero pressure force over the boundary. However, an unphysical mass flux still might exist at the boundary. The drawback of the Neumann boundary condition for pressure at the immersed boundary has been discussed in [33, 34]. To explain this issue, Equations (4) and (5) are written as follows near the immersed boundary:

$$\nabla^2 P^{n+1/2} = \frac{1}{\Delta t} \nabla \mathbf{u}_{ghost}^*, \quad (21)$$

$$\mathbf{u}^{n+1} = \mathbf{u}_{ghost}^* - \Delta t \nabla P_{ghost}^{n+1/2}. \quad (22)$$

$\nabla \mathbf{u}_{ghost}^*$ is computed by involving the ghost velocity field \mathbf{u}_{ghost}^* , which contains the interpolated (fictitious) values at the ghost points. The quantity $\nabla P_{ghost}^{n+1/2}$ takes into account the Neumann condition at the boundary. By applying the divergence operator to Eq. (22), and using Eq. (21),

$$\begin{aligned} \nabla \mathbf{u}^{n+1} &= \nabla \mathbf{u}_{ghost}^* - \Delta t \nabla^2 P_{ghost}^{n+1/2} \\ &= \Delta t \nabla^2 P^{n+1/2} - \Delta t \nabla^2 P_{ghost}^{n+1/2}. \end{aligned} \quad (23)$$

The pressure field P_{ghost} is modified from the field P using:

$$P_{ghost}(\mathbf{x}) = \begin{cases} P(\mathbf{x}), & \text{if } \mathbf{x} \in \text{fluid and solid region}, \\ P^*(\mathbf{x}), & \text{if } \mathbf{x} \in \text{ghost region}, \end{cases} \quad (24)$$

where \mathbf{x} denotes the location of the computed grid point. $P^*(\mathbf{x})$ is usually interpolated or extrapolated from $P(\mathbf{x})$ by incorporating the Neumann boundary condition for pressure on the IB. By involving the ghost (unphysical) values \mathbf{u}_{ghost}^* in the above computation, the pressure correction equation 5 (Eq. (22)) would generate a driving pressure force if the Neumann boundary condition is not enforced. Thus, the inter/extrapolated $P^*(\mathbf{x})$ is not the same as $P(\mathbf{x})$ at the ghost points. It is also obvious from Eq. (23) and Eq. (24) that $\nabla \mathbf{u}^{n+1}$ is not equal to 0 at the fluid points near the IB, leading to an unphysical mass source or sink term there. This discussion 10 shows that there would be an unphysical mass flux across the boundary if the Neumann boundary condition would be simply enforced for the pressure near the boundary. Therefore, as already done in [33, 34], the physical solution, instead of the fictitious velocity field, is applied at the ghost points in Eq. (21) and Eq. (22). More numerical details can be found in the method 15 summary section (Section 2.5). In this manner, there is no driving pressure force along the IB and a Neumann boundary condition for pressure is not needed. As a result, local continuity is preserved, without any additional mass source or sink. This pressure treatment is also simpler compared to the implementation of a Neumann boundary condition.

20 After implementing all these features, the resulting ghost-cell IBM should deliver accurate and stable results while fulfilling the divergence-free condition, as will be shown in what follows for different test-cases of increasing complexity.

2.5. Method summary

The numerical procedure for the proposed IBM approach can be summarized as follows:

1. Compute the level set function, which is time-dependent for a moving geometry.
2. Determine the locations of the ghost points (i.e., one layer of grid points within the solid body).
3. Substitute the velocity at the ghost points and solid points with the boundary (solid body) velocity and store it as pre-extrapolation velocity field $\mathbf{u} = (u, v, w)$. Extrapolate the ghost values at the ghost points using Eqs. (19) and (20). For multi-dimensional cases, extrapolate in each direction separately, and store the scalar ghost values in vectors ($\mathbf{u}_G = (u_{gx}, u_{gy}, u_{gz})$, $\mathbf{v}_G = (v_{gx}, v_{gy}, v_{gz})$ and $\mathbf{w}_G = (w_{gx}, w_{gy}, w_{gz})$ where the indices gx, gy, gz represent the values in x, y, z directions.) After extrapolation, there are three (for 3D cases) newly generated velocity fields, denoted as $\mathbf{u}_{gx} = (u_{gx}, v_{gx}, w_{gx})$, $\mathbf{u}_{gy} = (u_{gy}, v_{gy}, w_{gy})$, and $\mathbf{u}_{gz} = (u_{gz}, v_{gz}, w_{gz})$. These velocity fields keep the same values at the fluid points as the pre-extrapolation velocity field $\mathbf{u} = (u, v, w)$.
4. Compute the fictitious force based on the directional ghost values, using

$$\mathbf{F}(\mathbf{x}) = \begin{cases} \mathbf{R}_{b,\mathbf{x}}(\mathbf{u}_{gx}, \mathbf{u}_{gy}, \mathbf{u}_{gz}) - \mathbf{R}_{c,\mathbf{x}}(\mathbf{u}), & \text{if } \mathbf{x} \in \text{IBM influence range} \\ 0, & \text{otherwise.} \end{cases} \quad (25)$$

The range of IBM influence has been discussed in the previous section, and depends on the actual discretization scheme, as shown in Fig. 1. It

is smaller if the scheme is more compact and if the discretization order is lower. $\mathbf{R}_{c,\mathbf{x}}(\mathbf{u})$ is the right-hand-side of the momentum equation:

$$\mathbf{R}_{c,\mathbf{x}} = \mathbf{L}(f_c(u|v|w, x|y|z, 1|2)), \quad (26)$$

The term $\mathbf{R}_{b,\mathbf{x}}(\mathbf{u}_{gx}, \mathbf{u}_{gy}, \mathbf{u}_{gz})$ is obtained from

$$\begin{aligned} \mathbf{R}_{b,\mathbf{x}} = & \mathbf{L}(f_b(u_{gx}|v_{gx}|w_{gx}, x, 1|2), \\ & f_b(u_{gy}|v_{gy}|w_{gy}, y, 1|2), f_b(u_{gz}|v_{gz}|w_{gz}, z, 1|2)), \end{aligned} \quad (27)$$

where function f_c has been presented in Eqs. (9) and (10), function f_b has been defined in Eqs. (17) and (18). The symbol | represents "or". $\mathbf{L}()$ is the vector function, whose output components are computed by the scalar inputs separately.

5. Add the fictitious force to the right-hand-side of the momentum equation (Eq. (3)) as:

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t (\mathbf{R}^n + \mathbf{F}(\mathbf{x})^n). \quad (28)$$

10. Equation (28) is computed only in the fluid region. Then, solve the Poisson equation (Eq. (4)), for instance with a fast spectral approach [27, 35]:

$$\nabla^2 P^{n+1/2} = \frac{1}{\Delta t} \nabla \mathbf{u}^*, \quad (29)$$

15. Finally, update the velocity field \mathbf{u}^{n+1} using the pressure correction equation (Eq. (5)) and the pressure field computed from Eq. (29). During this process, physical solutions remain at the ghost points and solid points.

6. Continue the next Runge-Kutta sub-time step. (Go back to Step (3).)

7. Continue the next time step. (Go back to Step (3) for a stationary boundary or to Step (1) for a moving boundary.)

As the algorithm only needs to replace the partial derivatives concerning convection and diffusion terms, it is fairly independent of the actual procedures
⁵ employed for spatial discretization and temporal iteration. Concerning for instance time integration, the IBM algorithm has been already successfully combined with a fully explicit, low-storage 4th-order Runge-Kutta method [28], a 3rd-order explicit Williamson method [36], and a 3rd-order semi-implicit Williamson-Runge-Kutta method with analytic Jacobian inversion
¹⁰ (in case of reactive-flow) [37], without any major impact on the implementation.

2.6. Geometry representation and boundary motion

The present ghost-cell IBM is combined with a level-set method [38] to represent the complex geometry and the motion of the boundary. The level-set function ϕ is computed at every grid point and contains the signed distance from the grid point to the immersed boundary. Therefore, $\phi = 0$ is exactly where the boundary locates. For a moving boundary, the level-set function ϕ is varying with time. To compute ϕ for a given geometry, the explicit mathematical expression of the level-set function or the triangular
¹⁵ surface mesh of the geometry should be known; then, the level-set values can be directly computed from the explicit function, or reconstructed from the surface mesh. The values of ϕ at the ghost points are used to compute Δl and are thus necessary for an accurate reconstruction of the boundary. More
²⁰ details about the level-set method can be found in [39, 40, 41].

To deal with a moving boundary, one critical issue is the computation of the fluid variables such as velocity and pressure for the freshly generated fluid points that were solid points at the previous time step, i.e. for which ($\phi_i^{n+1}\phi_i^n < 0$). In the present study, the fluid values at these new 5 fluid points are interpolated from the surrounding fluid points and from the nearby boundary point, taking into account the moving boundary. Details concerning the implemented interpolation algorithm can be found in Mittal et al. [9].

2.7. Efficient implementation and parallelization

10 The implementation of the developed ghost-cell method is flexible. It can be combined with many different spatial discretization schemes and time integration approaches. As the extrapolation scheme always stick to the discretization directions, this method does not need to solve an interpolation matrix for the ghost values, as done for instance in [9, 12]. This avoids additional 15 iterations involved by the matrix computation, reducing computational time.

For most conventional ghost-cell IBM methods, parallelization on large distributed-memory computers is not easy. By involving the image point [9, 16, 21, 23], which is projected normally through the boundary from the 20 ghost point, the communication between different CPU/GPU processors becomes necessary, which is particularly challenging when the corresponding ghost point is near a domain-decomposition interface. Though it would be possible to derive a communication pattern containing directly the corner points when parallelizing the IBM procedure, this is not so straightforward 25 for parallelization algorithms relying on the pencil decomposition. This is-

sue disappears in the present ghost-cell IBM, since it does not involve any boundary-normal projected point. The parallelization of the new ghost-cell IBM is simple and straightforward. For the test cases shown in what follows, parallelization relies on a 2D pencil decomposition based on the open-
 5 source library 2DECOMP&FFT [42]. The communication between subdomain boundaries is handled by the halo-cell communication function integrated in the library.

3. Benchmark Simulations

In this section, the stability and accuracy of the new ghost-cell IBM
 10 method is checked in various benchmarks involving incompressible flow simulations of increasing complexity. Both stationary and moving, simple and complex geometries are taken into account. The rate of convergence is evaluated from the linear regression of the norm errors, which are denoted L_m , with $L_m = (\sum_i^N |e_i|^m / N)^{1/m}$. Here e_i is the error relative to the exact (base-
 15 line) solution at the i -th grid point, and N is the total number of grid points in the computational domain. The L_∞ error is defined as the largest local error ($\max |e_i|$) in the whole domain.

3.1. Tilted planar Poiseuille flow

This first case is simulated to check the magnitude of the error near the
 20 immersed boundary. An exact analytical solution exists for this case. In order to activate the IBM, the planar channel has been tilted on purpose by a specific angle, so that the channel boundaries in general do not align with the actual grids. The Reynolds number for the planar Poiseuille flow is defined as $\text{Re} = UD/\nu$, where D is the distance between the two channel

boundaries, U is the mean flow velocity, ν is the kinematic viscosity. A laminar flow with Reynolds number $Re = 120$ is simulated in the present case. The numerical setup of the tilted Poiseuille flow is shown in Fig. 3.

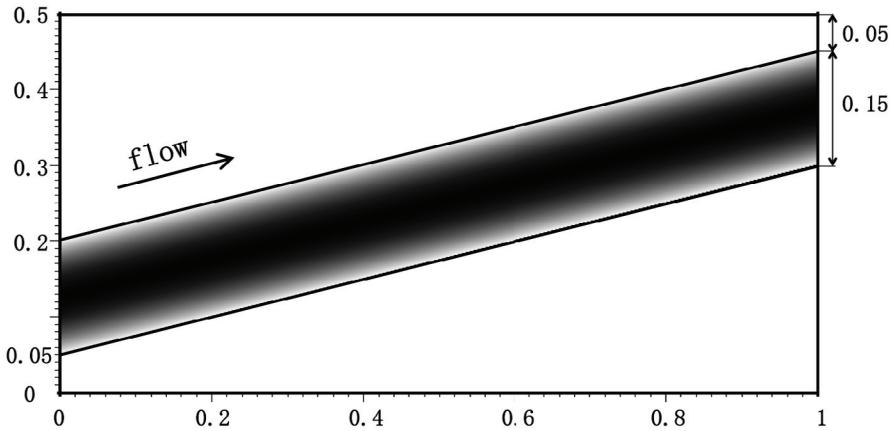


FIGURE 3. Computational domain for the tilted planar Poiseuille flow (all lengths are in cm).

The planar Poiseuille flow has been simulated twice using two different
 5 IBM approaches: 1) the ghost-cell IBM described in [9], and 2) the novel
 IBM approach described in this work. Both simulations are continued up
 to the same iteration with the same constant time step $\Delta t = 10^{-5}$ s. The
 absolute error is computed in a smaller domain $[0.1:0.9]\text{cm} \times [0:0.5]\text{cm}$, to
 exclude the possible influence of the inlet and outlet boundary. This choice
 10 gives a more reasonable comparison between these two methods, as we don't
 know the performance of the reference IBM [9] intersected with the real
 boundary. Table 1 shows the error norms computed for the two methods
 using the same uniform grid with 2048×1024 points. The magnitude of the

error norm is smaller using the proposed IBM method, which indicates that the representation of the domain boundaries is improved.

To quantify the convergence rate of the proposed IBM, the absolute error has been computed for different grid resolutions (2048×1024 , 1024×512 ,
5 512×256 , 256×128). Table 1 shows the errors and the estimated convergence orders at each refinement step. Similarly, figure 4 shows the convergence rates of the L_1 , L_2 and L_∞ norm errors for the two components (u and v) of velocity. It can be seen that the results are approaching second-order accuracy, deviations being due to the 1st order fallback scheme described by Eq. (20).

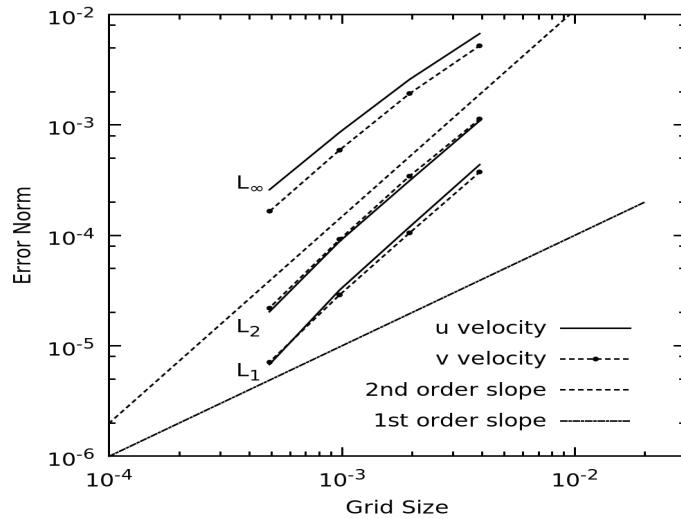


FIGURE 4. Convergence of the L_1 , L_2 and L_∞ norm errors of u and v velocity components when refining the computational grid.

10 3.2. Uniform flow past a fixed circular cylinder

This case is a standard benchmark for the validation of Navier-Stokes solvers and IBM algorithms. The resulting flow shows different behaviors as

| Grids | 256×128 | 512×256 | 1024×512 | 2048×1024 | 2048×1024 ([9]) |
|--------------------|-----------------------|-----------------------|-----------------------|-----------------------|--------------------------|
| L_1 for u | 4.37×10^{-4} | 1.19×10^{-4} | 3.20×10^{-5} | 6.78×10^{-6} | 5.67×10^{-5} |
| Est. order | - | 1.88 | 1.89 | 2.24 | |
| L_2 for u | 1.09×10^{-3} | 3.12×10^{-4} | 8.83×10^{-5} | 2.03×10^{-5} | 1.51×10^{-4} |
| Est. order | - | 1.80 | 1.82 | 2.12 | |
| L_∞ for u | 6.71×10^{-3} | 2.58×10^{-3} | 8.57×10^{-4} | 2.59×10^{-4} | 3.30×10^{-4} |
| Est. order | - | 1.38 | 1.59 | 1.73 | |
| L_1 for v | 3.74×10^{-4} | 1.05×10^{-4} | 2.89×10^{-5} | 7.12×10^{-6} | 1.40×10^{-5} |
| Est. order | - | 1.83 | 1.86 | 2.02 | |
| L_2 for v | 1.13×10^{-3} | 3.44×10^{-4} | 9.18×10^{-5} | 2.20×10^{-5} | 4.73×10^{-5} |
| Est. order | - | 1.72 | 1.91 | 2.06 | |
| L_∞ for v | 5.22×10^{-3} | 1.92×10^{-3} | 5.93×10^{-4} | 1.65×10^{-4} | 2.35×10^{-4} |
| Est. order | - | 1.44 | 1.70 | 1.85 | |

TABLE 1. Magnitude of the absolute error and computed convergence order for the planar Poiseuille flow simulation using the novel IBM compared to the simulation using the IBM from [9].

a function of the Reynolds number. The Reynolds number is $\text{Re} = U_\infty D / \nu$, where D is the diameter of the cylinder, U_∞ is the incoming velocity of the uniform flow. The flow remains steady and symmetrical for Reynolds number lower than about 47, then becomes unsteady and generates periodic Kármán vortex streets for higher Reynolds numbers. When the Reynolds number becomes higher than about 188.5, the flow begins to intrinsically turn into a three-dimensional flow [43, 44, 45]. For the present study, flows at Reynolds number of 40 and 100 have been investigated in two-dimensional simulations.

The simulation domain is a square with side length $40D$, which is large enough to minimize the domain confinement effects on the development of the wake. A uniform grid with 1024×1024 grid points is used to discretize the domain, which makes the grid resolution near the cylinder comparable to that of [15]. The cylinder is centered at $(16D, 20D)$ in the domain.

Figure 5 shows the streamlines around the cylinder at $\text{Re} = 40$. A steady solution is obtained, symmetrical about the wake centerline. The drag coefficient $C_D = F_x / (0.5\rho U_\infty^2 D)$ has been calculated, where F_x is the drag force calculated using the domain integral method described in [46]. Then, the length of the recirculation zone L_W has been extracted. As shown in Table 2, the comparison with previous studies, both numerical and experimental, is very good.

The convergence rate has been checked again in this case. The simulation results with a finer resolution 1024×1024 are taken as baseline solution for computing the truncation errors. The L_1 , L_2 and L_∞ norm errors using coarser resolutions (512×512 , 256×256 , 128×128) have been computed and tabulated in Table 3. The norm errors using the IBM from [9] with resolution

| | Re=40 | | Re=100 |
|------------------------|---------|-------|------------------|
| | L_W/D | C_D | $St=fD/U_\infty$ |
| Current study | 2.29 | 1.53 | 0.166 |
| Tseng et al. [15] | 2.21 | 1.53 | 0.164 |
| Ye et al. [19] | 2.27 | 1.52 | - |
| Lai et al. [46] | - | - | 0.165 |
| Kim et al. [11] | - | 1.51 | 0.165 |
| Baranyi [47] | - | - | 0.163 |
| Williamson (Exp.) [48] | - | - | 0.166 |

TABLE 2. Comparison between the current study and previous works concerning recirculation length L_W , drag coefficient C_D , and Strouhal number $St=fD/U_\infty$, where f is the frequency of vortex shedding.

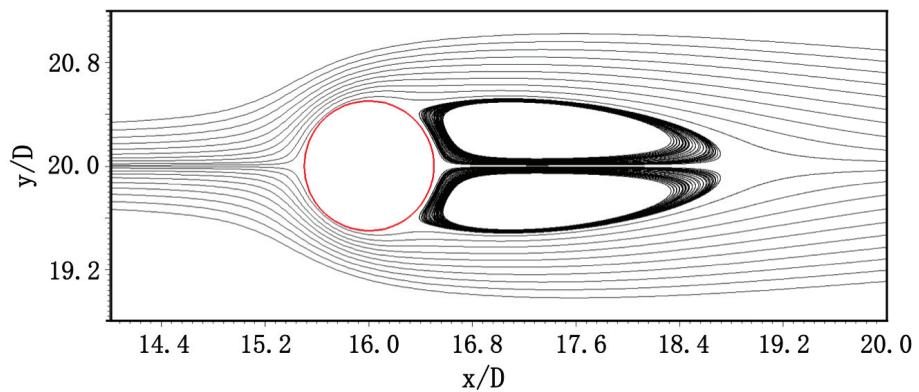


FIGURE 5. Streamlines of the flow around a fixed circular cylinder at $Re = 40$

512×512 is also tabulated for comparison. The proposed IBM systematically leads to a slightly smaller absolute error. Figure 6 shows the associated convergence rate. As it can be seen, the measured errors are again converging with the grid size at a rate close to the second-order slope.

| Grid size (mm) | 1/128 | 1/256 | 1/512 | 1/512 ([9]) |
|--------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| L_1 for u | 8.62×10^{-4} | 1.59×10^{-4} | 2.88×10^{-5} | 2.04×10^{-4} |
| Est. order | - | 2.44 | 2.46 | |
| L_2 for u | 2.45×10^{-3} | 4.43×10^{-4} | 7.41×10^{-5} | 5.84×10^{-4} |
| Est. order | - | 2.47 | 2.58 | |
| L_∞ for u | 5.17×10^{-2} | 1.41×10^{-2} | 4.12×10^{-3} | 8.14×10^{-3} |
| Est. order | - | 1.87 | 1.77 | |
| L_1 for v | 6.11×10^{-4} | 1.02×10^{-4} | 1.68×10^{-5} | 1.44×10^{-4} |
| Est. order | - | 2.58 | 2.60 | |
| L_2 for v | 1.54×10^{-3} | 2.96×10^{-4} | 5.49×10^{-5} | 4.01×10^{-4} |
| Est. order | - | 2.38 | 2.43 | |
| L_∞ for v | 3.84×10^{-2} | 9.81×10^{-3} | 2.63×10^{-3} | 4.81×10^{-3} |
| Est. order | - | 1.97 | 1.90 | |

TABLE 3. Magnitude of the absolute error and computed convergence order for the flow past a circular cylinder at $Re = 40$ using the novel IBM in comparison with the IBM from [9].

For the flow at $Re = 100$, the wake becomes unstable and a Kármán vortex street is generated, as shown in Fig. 7.

The vortex shedding frequency f can be obtained from the periodic variation of the flow. The Strouhal number $St = fD/U_\infty$ is then calculated,

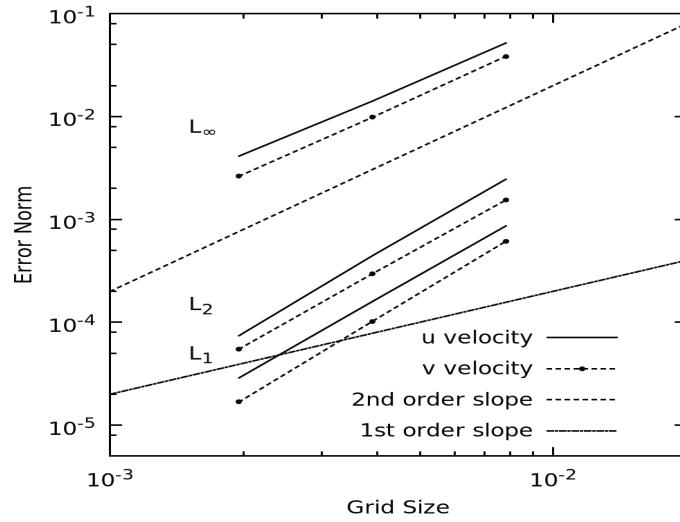


FIGURE 6. Convergence of the L_1 , L_2 and L_∞ norm errors of the streamwise (u) and spanwise (v) velocity with the computational grid size.

and is shown in Table 2. The obtained results match very well with previous numerical and experimental studies.

3.3. Oscillating circular cylinder in a fluid at rest

After checking the accuracy for a flow over a fixed geometry, the performance of the ghost-cell IBM for moving geometries must also be investigated. For this purpose, an oscillating circular cylinder in a fluid initially at rest is simulated using the proposed method.

The cylinder is initially located at the center of the domain. Its center oscillates in the streamwise direction as follows

$$x(t) = -A \sin(2\pi ft), \quad (30)$$

where A is the amplitude of the oscillation and f is the oscillation frequency.

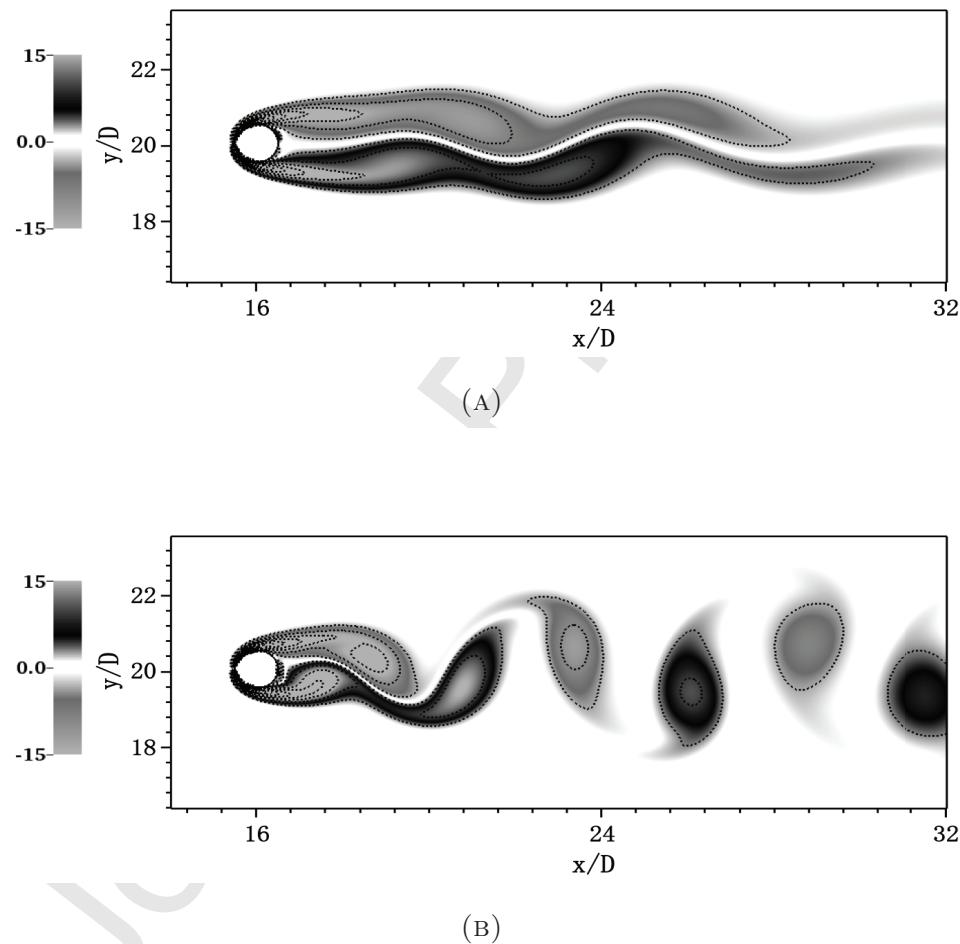


FIGURE 7. Instantaneous vorticity contours plot for the flow at $\text{Re} = 100$ around the fixed circular cylinder at (a) $t = 30T$ (b) $t = 60T$, with $T = D/U_\infty$.

The Reynolds number is defined as $\text{Re} = U_{max}D/\nu$, where U_{max} is the maximum velocity of the cylinder, D is the diameter of the cylinder, ν is the kinematic viscosity. Another important parameter is the Keulegan-Carpenter number, $\text{KC} = U_{max}/fD$. For the present study, $\text{Re} = 100$ and $\text{KC} = 5$,

- 5 keeping the values previously used by Dütsch et al. [49] for LDA experiments and a companion numerical study.

The computational domain is chosen to be $60D \times 30D$. All the boundaries are outflow boundary conditions with a velocity gradient equal to 0. The cylinder surface is associated to a no-slip velocity boundary condition, with

- 10 a surface velocity $u(t) = -2\pi f A \cos(2\pi ft)$. The fluid domain is discretized by a uniform grid with 2048×1024 points, resulting in a mesh resolution near the cylinder similar to that used in [50].

Figure 8 shows the normalized streamwise and spanwise velocity (u/u_{max} and v/v_{max}) at 4 different cross-sections ($x = -0.6D, 0D, 0.6D$ and $1.2D$) for

- 15 3 different phase positions ($\phi = 180^\circ, 210^\circ, 330^\circ$). The results are compared with the LDA measurements of [49]. Even if some discrepancies are observed, it is clear that the present results are overall in good agreement with the experimental data.

The in-line force F_x acting on the oscillating cylinder has been computed

- 20 as a function of time and is shown in Fig. 9. Here, the comparison with the experimental measurements is perfect, confirming the accuracy of the present ghost-cell IBM for describing the flow around a moving geometry.

3.4. Flow around a flapping wing

The previous case considered only the translational motion of a solid

- 25 body. In the present configuration, the flow induced by a flapping wing is

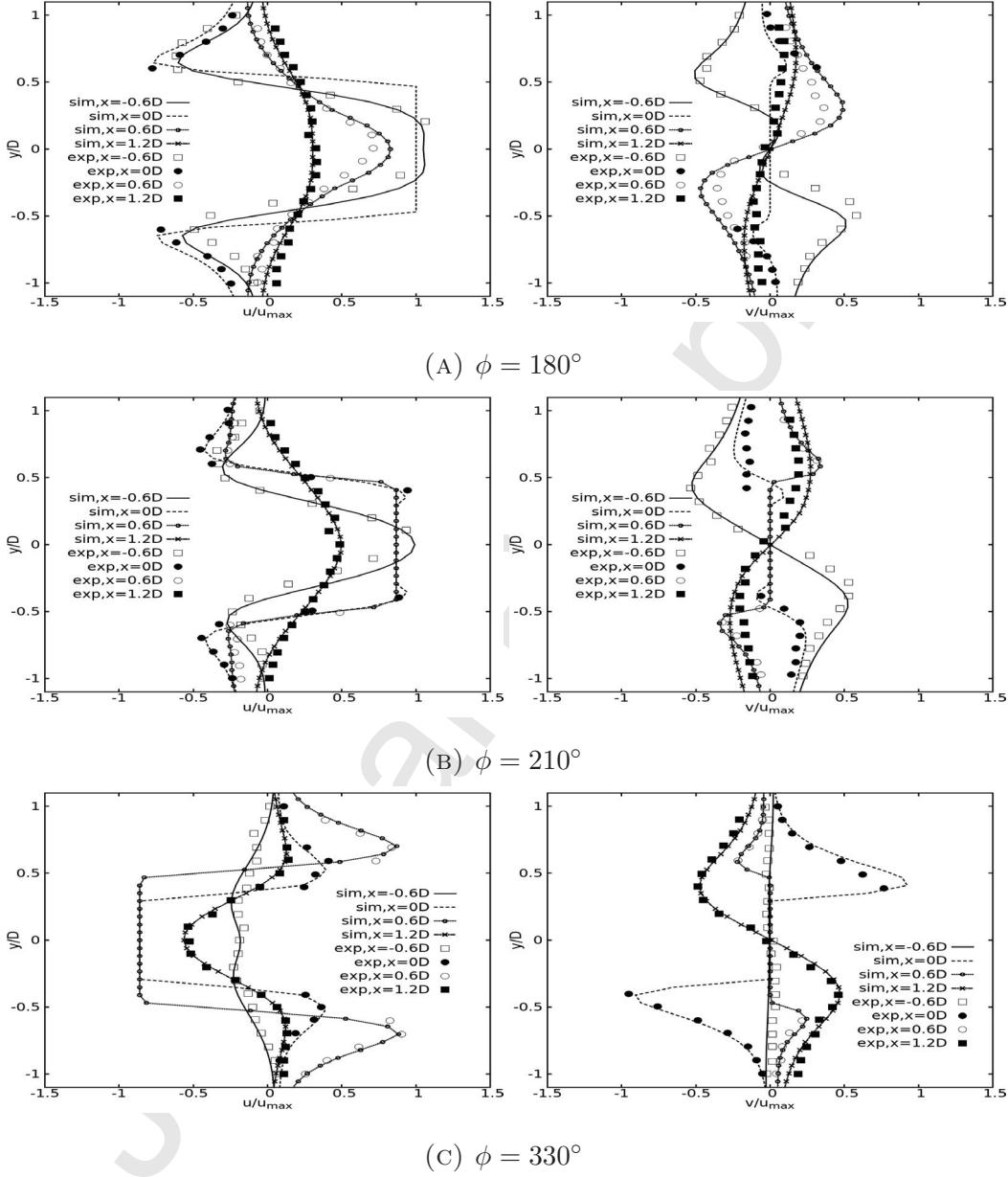


FIGURE 8. Velocity profiles in streamwise u (left), and spanwise direction v (right), at 4 different stream-wise cross-sections ($x = -0.6D, 0D, 0.6D$ and $1.2D$) for 3 different phase positions: (A) $\phi = 180^\circ$; (B) $\phi = 210^\circ$ and (C) $\phi = 330^\circ$. The symbols are experimental results from [49]. The lines denote the present results.

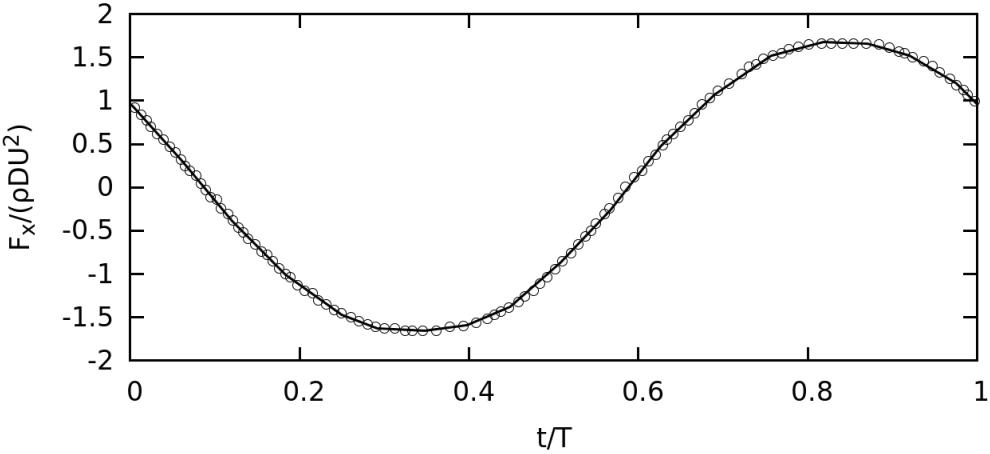


FIGURE 9. Time history of the in-line force acting on the cylinder. Symbols: experimental results from [49]; solid line: present simulation results.

investigated, which involves both translational and rotational motion. The velocity along the body surface points is not uniform any more, resulting in a far more complicated fluid-structure interaction. The configuration of this case follows the setup originally proposed by Wang [51]. The hovering wing
5 is a two-dimensional ellipse with major axis c (chord length) and aspect ratio (ratio between major axis and minor axis) e . The wing moves along a stroke plane inclined at an angle of β . The prescribed translational and rotational motion of the hovering wing is governed by

$$A(t) = 0.5A_0(\cos(2\pi ft) + 1), \quad (31)$$

$$\theta(t) = \theta_0(1 - \sin(2\pi ft + \phi_0)), \quad (32)$$

10 where $A(t)$ is the translational motion of the wing with an amplitude A_0 . The flapping frequency is chosen as $f = \frac{c}{\pi A_0}$. The rotational motion of the wing starts from an angle of attack of θ_0 with phase difference $\phi_0 = 0$. The

chord length c is retained as the length scale, the maximum translational velocity $\pi A_0 f$ is taken as the velocity scale. Thus, the Reynolds number is $\text{Re} = \pi A_0 f c / \nu$. In the following simulation, the parameters are kept as in previous studies [50, 51, 52, 53]: $c = 1$, $e = 4$, $A_0 = 2.5c$, $\beta = \pi/3$, $\theta_0 = \pi/4$,
⁵ leading to $\text{Re} = 157$.

The computational domain is a square with domain length $l = 24c$. A uniform grid of 1024×1024 is used to discretize the domain, resulting in a grid spacing of $0.0234c$. This resolution is slightly finer than the grid spacing used in [52, 53], and comparable to the one employed in [50].

¹⁰ Figure 10 shows four snapshots of the vorticity fields near the wing during one flapping period. A vortex dipole is generated from the leading and trailing edge vortices and sheds right-downward, resulting in lift force for the wing. The vorticity fields shown in Fig. 10 are visually in good agreement with results of previous studies [50, 51, 52, 53].

¹⁵ The drag and lift coefficients on the flapping wing have been calculated as a function of time and are plotted in Fig. 11. The results from the body-conforming mesh method in [51], the moving IBM in [50] and the immersed interface method in [52] are also plotted in this figure, allowing a direct comparison with the current ghost-cell IBM. Concerning the drag coefficient
²⁰ (Fig. 11a), the novel ghost-cell IBM delivers results very similar to the IBM approach described in [50]. Concerning now the lift coefficient (Fig. 11b), our approach is the closest one to the body-conforming mesh results documented in [51]. At the same time, the time step employed for the novel ghost-cell IBM is noticeably larger than that documented in [52], where small
²⁵ values are necessary to maintain the shape of the rigid body when using the

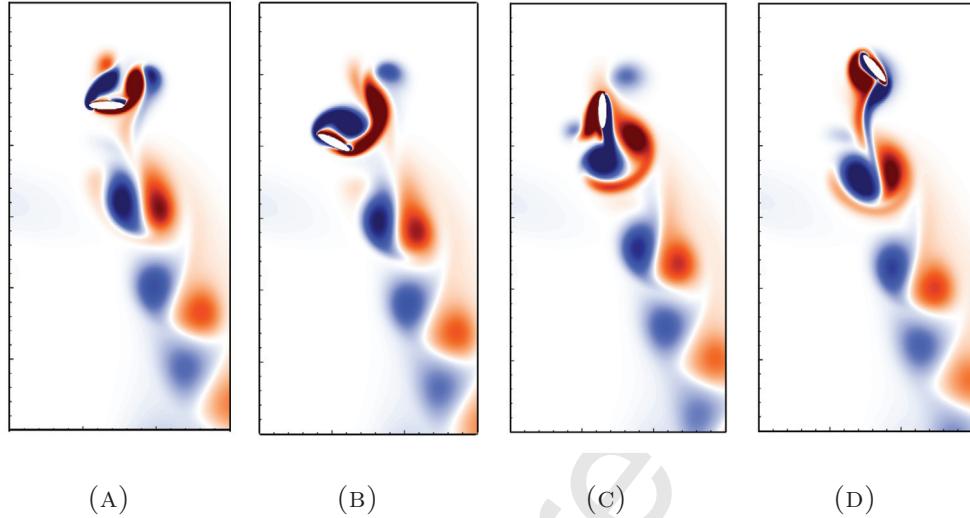


FIGURE 10. Snapshots of the vorticity fields near a flapping wing at $\text{Re} = 157$ at 4 different time instants during one flapping period T : (a) $t = 0.25T$; (b) $t = 0.44T$; (c) $t = 0.74T$; and (d) $t = 0.99T$.

immersed interface method. Overall, the performance of the novel ghost-cell IBM appears to be very satisfactory for such a complex fluid-structure interaction with a moving body.

3.5. Flow past a 3D sphere

All previous cases considered flows around 2D geometries. As a final step, the flow past a 3D sphere is now simulated to check the performance of the ghost-cell IBM in such configurations. This standard test case has been widely investigated in both experimental [54, 55, 56] and numerical studies [9, 57, 58], for low to moderate Reynolds numbers. It has been found that the flow is axisymmetric and steady below a Reynolds number of 210, becomes non-axisymmetric at a Reynolds number between 210 and 280, and finally unsteady at Reynolds numbers above 280 [5, 58].

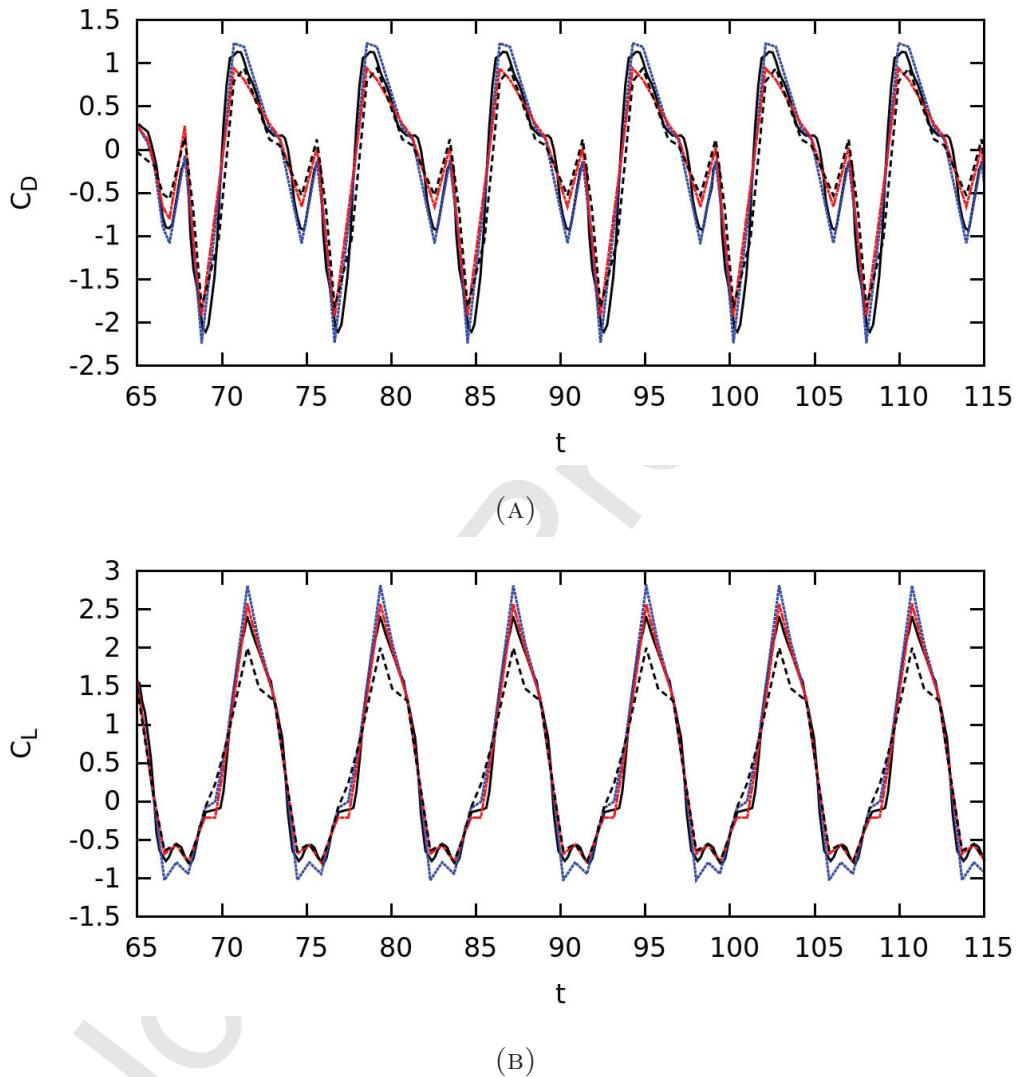


FIGURE 11. Time history of (a) drag coefficient and (b) lift coefficient for the flow around the flapping wing at $Re = 157$. — novel ghost-cell immersed boundary method; - - - body-conforming mesh method of [51]; ······ moving immersed boundary method of [50]; - - - - immersed interface method of [52].

In the present study, the flow past a fixed 3D sphere at a Reynolds number of 10 is first simulated to check that the novel ghost-cell IBM fulfills mass conservation (divergence-free condition). Using the same numerical setup as in [33], the computational domain has a size of $3.5d \times 4d \times 4d$ and 5 is discretized by a uniform Cartesian grid with $28 \times 32 \times 32$ points. The time step is controlled by a constant CFL number of 0.01. The L_2 norm error of the continuity equation has been computed for each iteration step and is compared with the results of [33] in Fig. 12. It can be seen that the total error of the novel IBM related to the continuity equation is kept below a 10 very small value (lower than 10^{-6}), proving that mass conservation is satisfactorily enforced. The error converged below 10^{-6} almost at the same time iteration step (iteration 14) as when using the IBM approach from [33] with ghost velocity excluded from the pressure correction equation.

Then, the same flows have been simulated at increasing Reynolds numbers of 100, 150, 300 and 350. The computational domain has a size of 15 $15d \times 15d \times 15d$ and is discretized by a uniform Cartesian grid with $256 \times 256 \times 256$ grid points.

Figure 13a shows instantaneous pseudo-streamlines around the sphere within the $x - y$ plane for the flow at $\text{Re}_d = 100$. In this configuration, 20 the flow is axisymmetric and steady, as documented in the literature. The absolute distance from the center of the flow recirculation pattern to the sphere center is denoted as x_c and y_c in x and y directions, respectively. The length of the recirculation zone is written L_W . These flow characteristics have been also calculated for flow at $\text{Re}_d = 150$ and compared with previous 25 studies, as documented in Table 4. A very good agreement is obtained.

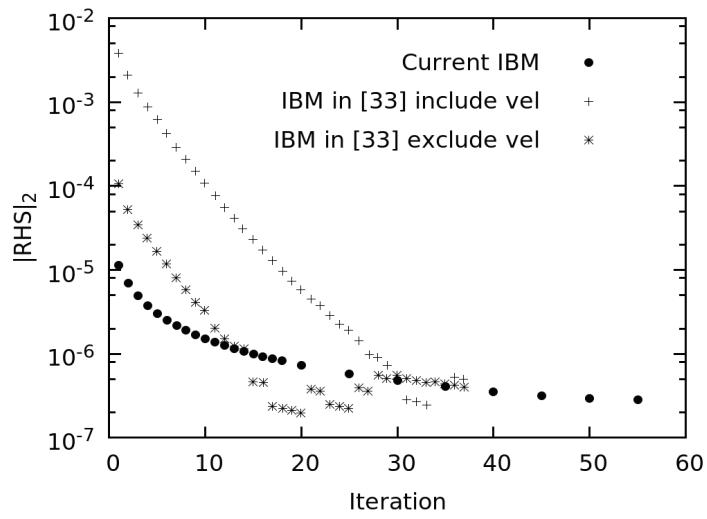


FIGURE 12. L_2 norm error for the continuity equation concerning the flow around a fixed 3D sphere at a Reynolds number of 10. The \bullet symbols represent results of the novel ghost-cell IBM. The $+$ symbols represent the IBM results in [33] with ghost velocity included in the pressure correction equation, while the $*$ represents the IBM results in [33] with ghost velocity excluded from the pressure correction equation.

| | Re _d =100 | | | Re _d =150 | | | Re _d =300 | Re _d =350 |
|---------------------|----------------------|-------------------|-------------------|----------------------|-------------------|-------------------|----------------------|----------------------|
| | x _c /d | y _c /d | L _W /d | x _c /d | y _c /d | L _W /d | St | St |
| Present study | 0.75 | 0.285 | 0.85 | 0.32 | 0.29 | 1.19 | 0.134 | 0.14 |
| Taneda [59] | 0.745 | 0.28 | 0.8 | 0.32 | 0.29 | 1.2 | - | - |
| Mittal [43] | - | - | 0.87 | - | - | - | - | 0.14 |
| Johnson et al. [58] | 0.75 | 0.29 | 0.88 | 0.32 | 0.29 | 1.2 | 0.137 | - |
| Marella et al. [5] | - | - | 0.88 | - | - | 1.19 | 0.133 | - |
| Mittal et al. [9] | 0.742 | 0.278 | 0.84 | 0.31 | 0.3 | 1.17 | 0.135 | 0.142 |

TABLE 4. Comparison of the computed flow characteristics with other experimental and numerical studies.

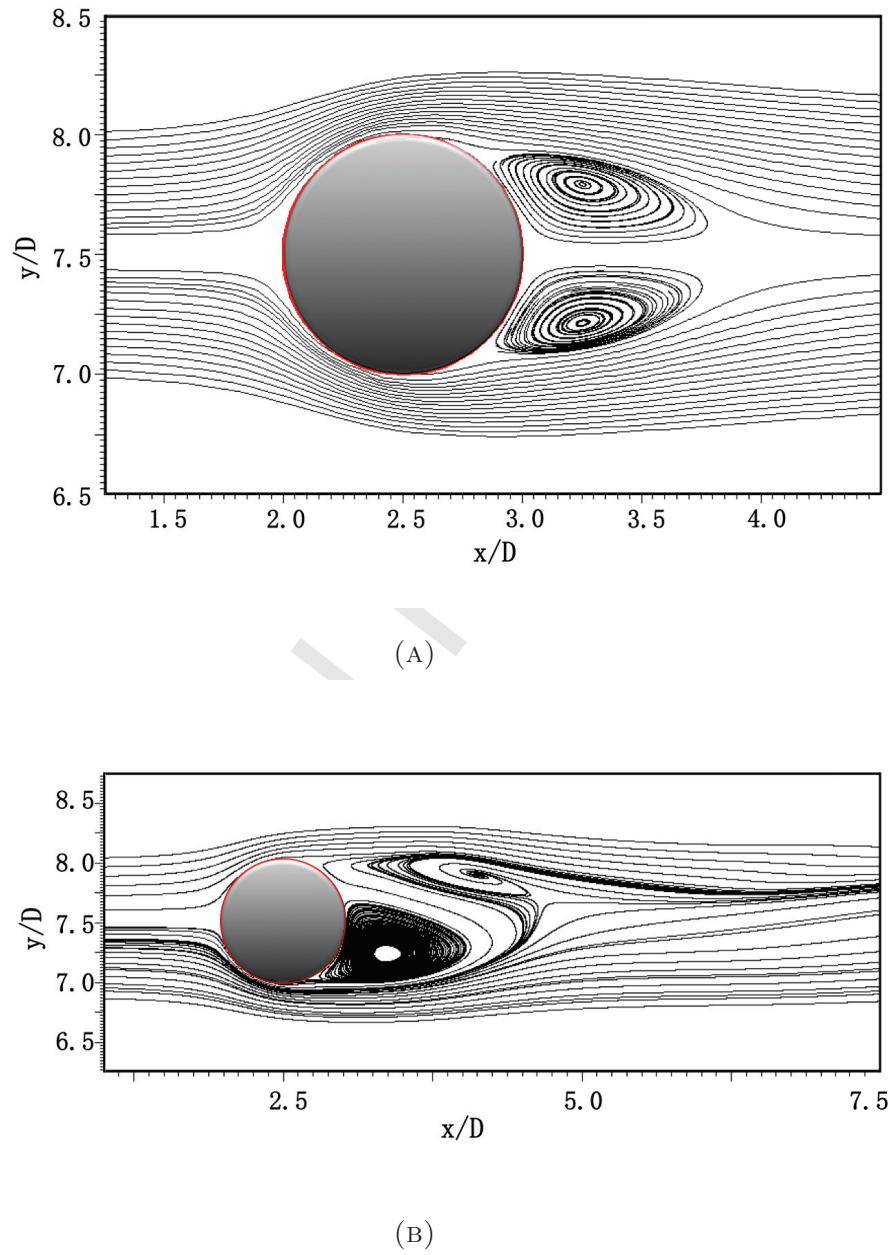


FIGURE 13. Instantaneous pseudo-streamlines of the velocity field around the sphere within the $x - y$ plane (A) for flow at $\text{Re}_d = 100$; (B) for flow at $\text{Re}_d = 300$.

At Reynolds numbers of 300 and 350, non-axisymmetric and unsteady flows are obtained. To reach a periodic state, the simulations have been appropriately continued until a periodic signal was found for the near-wake pressure. The frequency of this periodic signal is used to estimate the Strouhal number St , as shown in Table 4.

Figure 13b shows pseudo-streamlines around the sphere within the $x - y$ plane for the flow at $Re_d = 300$ at one time instant. This time-instant has been retained to allow comparison with the same streamlines (second flow phase) in [58]. It can be seen in Fig. 13b that the upper side of the near-wake vortex is becoming unstable and starts moving downstream, while the lower side of the vortex is becoming stable.

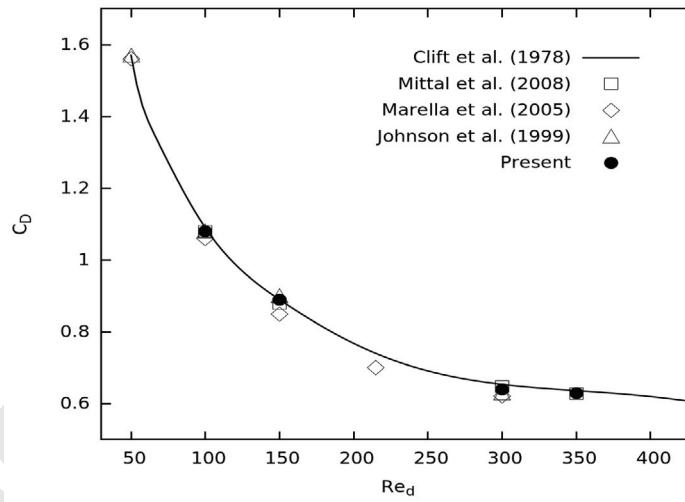


FIGURE 14. Comparison of the (averaged) drag coefficients with results from the literature.

The mean drag coefficient obtained for all these cases has been calculated and is summarized in Fig. 14. As the flows for $Re_d = 300$ and 350 are

unsteady, the drag coefficients for these two cases have been averaged after reaching periodic steady-state.

From the comparisons in Table 4 and Fig. 14, it is clear that the results obtained with the new method are in excellent agreement with previously published experimental and numerical studies, showing that the novel ghost-cell IBM method can be used for 3D cases as well.
5

3.6. Flow in an intracranial aneurysm

The last benchmark involves a medical flow simulation using a patient-specific intracranial aneurysm, since this topic has been very intensively considered in our group in the last decade [60]. This case is used to check the fidelity of the novel ghost-cell IBM for highly irregular 3D geometries, as well as to measure its parallel efficiency. The employed 3D aneurysm geometry has been derived from a female patient as described in [61]. Previous numerical and experimental studies considering this same geometry are discussed in particular in [62]. Following the guidelines discussed in [62], blood is described as an incompressible fluid with density $\rho = 1055 \text{ kg/m}^3$ and constant dynamic viscosity $\mu = 0.004 \text{ Pa.s}$. The period of each cardiac cycle for the patient is taken constant at 0.925 s. The patient-specific inflow velocity prescribed during each cardiac cycle is plotted in Fig. 15. Outflow boundaries are constant-pressure outlets. The grid resolution in this simulation is set to 40 μm . The simulation was carried out on SuperMUC (supercomputer cluster at Leibniz Supercomputing Center in Munich) using 1024, 2048, 4096, and 8192 CPU processors with 2.6 GHz, respectively. The CPU time needed for the IBM process has been measured separately by activating appropriate counters. Figure 16 shows the strong scaling plot of the overall CPU time
10
15
20
25

per time step. It can be seen that a good parallel efficiency (almost 80%) is obtained for this low-Mach solver. The efficiency of the proposed IBM procedure is shown separately in Fig. 16. Again, the parallel efficiency of the IBM method is very high, comparable to that of the overall code (70 to 80%).

- 5 It can also be seen that the numerical cost associated to IBM is roughly only 7 – 9% of the total computational time.

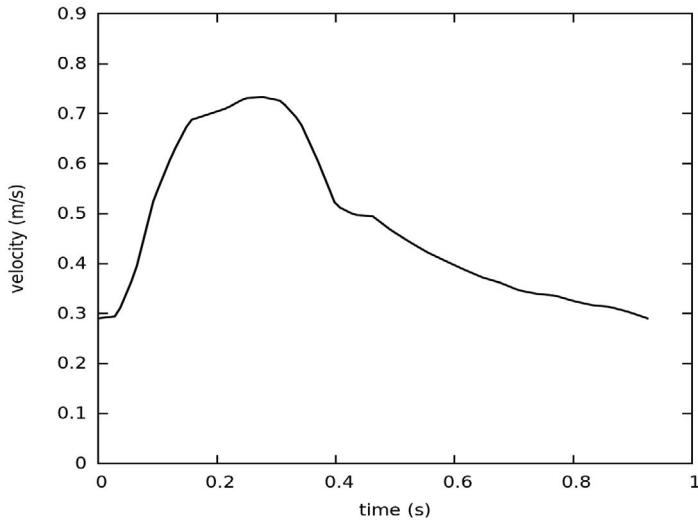


FIGURE 15. Patient-specific inflow velocity used during each cardiac cycle (duration of 0.925 s).

As found in [62], the simulation needs to be continued for at least 3 cardiac cycles in order to eliminate initial transients. Figure 17 shows the aneurysm geometry, represented as iso-surface of the level-set function, $\phi =$ 10 0. Two cut-planes have been selected for comparing the results with those presented in [62]. The peak-systole velocity magnitude obtained with the novel ghost-cell IBM on these two cut-planes is shown in Fig. 18. Selected simulation results documented in [62] for the velocity magnitude are also

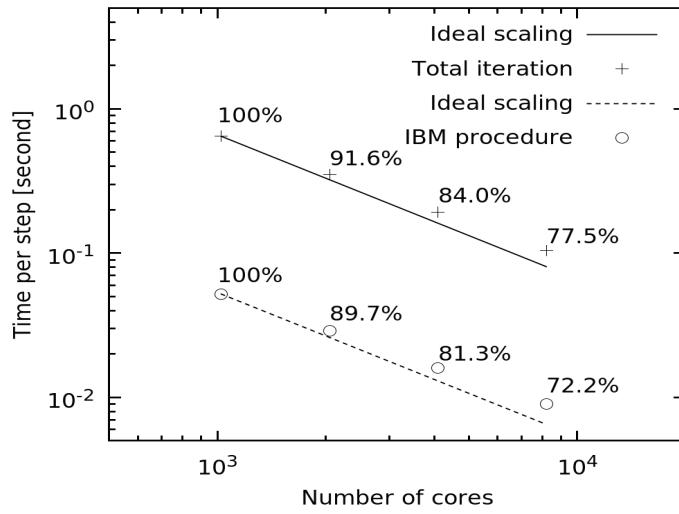


FIGURE 16. Measured parallel efficiency of the overall numerical solution and of the IBM procedure on SuperMUC.

shown in Fig. 18, allowing a direct comparison.

Compared with Cases C, E, I, K and X in [62], carried out using adapted unstructured meshes, the results obtained on a regular Cartesian grid with the novel ghost-cell IBM show an excellent agreement for plane P2. Slight
5 discrepancies are observed on plane P1, but deviations are below 10% when comparing the present results with the solution averaged from all Cases in [62]. This last test case confirms robustness and accuracy of the proposed method.

4. Conclusions

10 An efficient, robust and accurate ghost-cell immersed boundary method has been developed for incompressible flow simulations involving complex geometries. It has been implemented in a finite-difference DNS solver. In

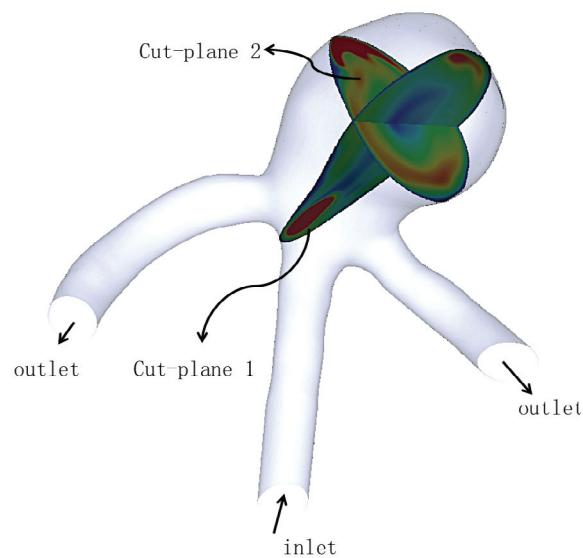


FIGURE 17. Geometry of the patient-specific cerebral aneurysm with inlet/outlets, and locations of two orthogonal cut-planes used to analyze the results. The color plots on the planes show the peak-systolic velocity magnitude.

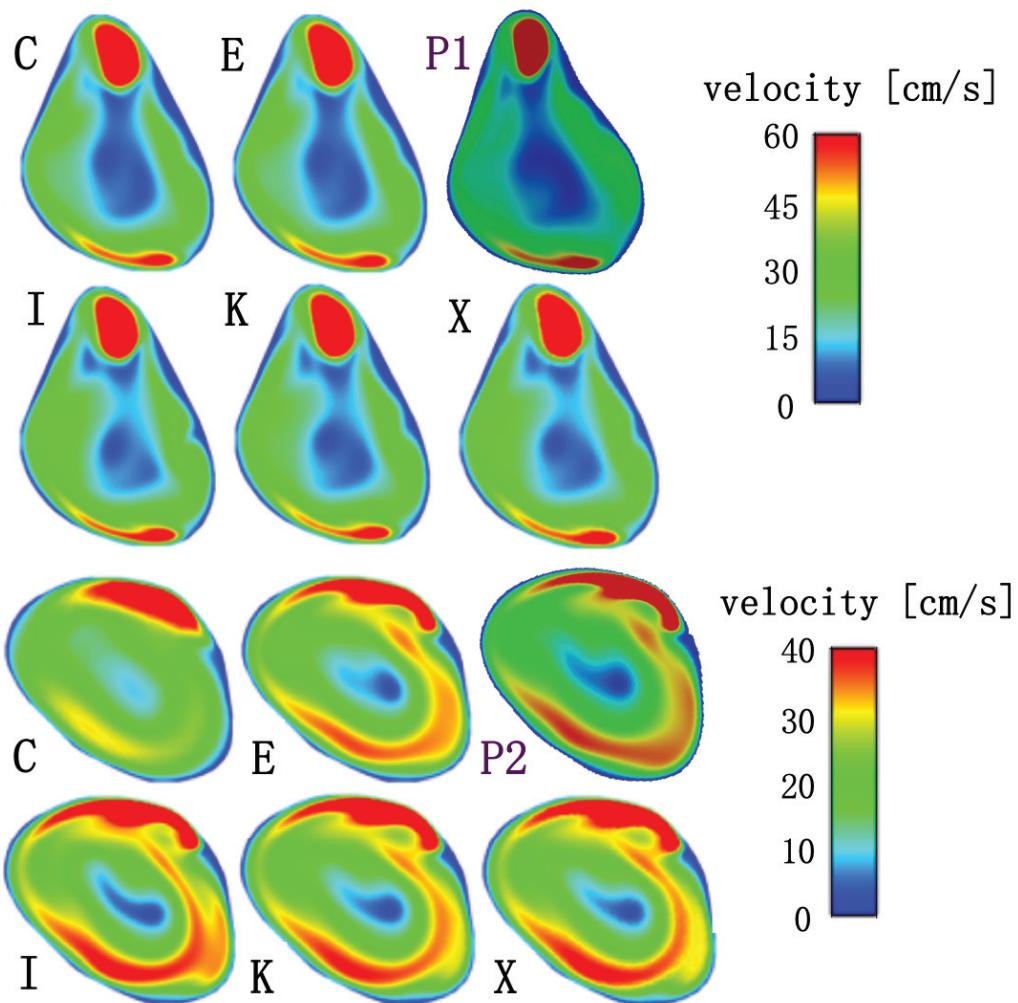


FIGURE 18. Comparison of the peak-systolic velocity magnitude along the two cut-planes (P1: top; P2: bottom) using the novel ghost-cell IBM and various adapted unstructured meshes (Cases C, E, I, K, and X in [62]).

contrast to conventional ghost-cell methods in which ghost values are extrapolated normally through the boundary, the present method always follows the discretization directions, but involving multiple ghost values at the same ghost points in multi-dimensional simulations (3 values at each ghost point in 3D). This leads to an easier and more accurate representation of the boundary on the employed Cartesian grid. The effect of the assumption $\Delta l = O(\Delta x)$ on the persistent convergence of the truncation error has been discussed and the error term involving Δl has been carefully treated. Finally, a second-order convergence is obtained for the truncation errors. Stability issues observed when one fluid point is close to the boundary have been solved by reducing progressively the local order of accuracy. To improve mass conservation near the boundary, the commonly used Neumann boundary condition for pressure is not used, avoiding an artificial pressure force over the boundary. The conservation constraint is implicitly enforced by the continuity equation, without involving the ghost velocities. Parallelization of the developed ghost-cell IBM is straightforward, reducing communication issues compared to conventional methods.

Various test cases ranging from the flow around a cylinder in two dimensions to the pulsating flow inside a patient-specific, three-dimensional cerebral aneurysm have been considered and discussed. Both steady boundaries and moving boundaries have been employed. The simulation results have been extensively compared to previous experimental and numerical studies, showing a very good agreement. These tests have shown that the novel ghost-cell IBM is second-order accurate near the boundary, robust even for moving geometries and complex boundaries, and ensures mass conservation.

Acknowledgement

This work has been financially supported by the EU-programme ERDF (European Regional Development Fund) within the research Center of Dynamic Systems (CDS) and the International Max Planck Research School 5 Magdeburg for Advanced Methods in Process and Systems Engineering (IMPRS ProEng). The computer resources provided by the Gauss Center for Supercomputing/Leibniz Supercomputing Center Munich are gratefully acknowledged. The authors thank Prof. Berend van Wachem and Prof. Bok Jik Lee for very interesting scientific discussions when preparing this article.

10 Reference

- [1] C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* 25 (1977) 220 – 252.
- [2] C. S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [3] R. Mittal, G. Iaccarino, Immersed boundary methods, *Annu. Rev. Fluid Mech.* 37 (2005) 239–261.
- [4] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, *Appl. Mech. Rev.* 56 (2003) 331–347.
- [5] S. Marella, S. Krishnan, H. Liu, H. Udaykumar, Sharp interface Cartesian grid method I: An easily implemented technique for 3D moving boundary computations, *J. Comput. Phys.* 210 (2005) 1–31.

- [6] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *J. Comput. Phys.* 209 (2005) 448 – 476.
- [7] T. Kempe, J. Fröhlich, An improved immersed boundary method with direct forcing for the simulation of particle laden flows, *J. Comput. Phys.* 231 (2012) 3663 – 3684.
- [8] C. S. Peskin, Flow patterns around heart valves: A numerical method, *J. Comput. Phys.* 10 (1972) 252–271.
- [9] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *J. Comput. Phys.* 227 (2008) 4825–4852.
- [10] J. Mohd-Yusof, Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries, *CTR Annual Research Briefs* (1997) 317–327.
- [11] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, *J. Comput. Phys.* 171 (2001) 132–150.
- [12] E. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *J. Comput. Phys.* 161 (2000) 35–60.
- [13] R. P. Fedkiw, Coupling an Eulerian fluid calculation to a Lagrangian

solid calculation with the ghost fluid method, *J. Comput. Phys.* 175 (2002) 200–224.

- [14] S. Majumdar, G. Iaccarino, P. Durbin, RANS solver with adaptive structured boundary non conforming grids, *Annual Research Briefs*, Center for Turbulence Research, Stanford University (2001) 353–466.
- [15] Y.-H. Tseng, J. H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, *J. Comput. Phys.* 192 (2003) 593–623.
- [16] R. Ghias, R. Mittal, H. Dong, A sharp interface immersed boundary method for compressible viscous flows, *J. Comput. Phys.* 225 (2007) 528–553.
- [17] D. K. Clarke, H. A. Hassan, M. D. Salas, Euler calculations for multi-element airfoils using Cartesian grids, *AIAA J.* 24 (1986) 353–358.
- [18] R. B. Pember, J. B. Bell, P. Colella, W. Y. Curchfield, M. L. Welcome, An adaptive Cartesian grid method for unsteady compressible flow in irregular regions, *J. Comput. Phys.* 120 (1995) 278–304.
- [19] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys.* 156 (1999) 209–240.
- [20] P. Colella, D. T. Graves, B. J. Keen, D. Modiano, A Cartesian grid embedded boundary method for hyperbolic conservation laws, *J. Comput. Phys.* 211 (2006) 347–366.

- [21] C. Chi, B. J. Lee, H. G. Im, An improved ghost-cell immersed boundary method for compressible flow simulations, *Int. J. Numer. Meth. Fluids* 83 (2017) 132–148.
- [22] T. Gao, Y. Tseng, X. Lu, An improved hybrid Cartesian/immersed boundary method for fluid–solid flows, *Int. J. Numer. Meth. Fluids* 55 (2007) 1189–1211.
- [23] D. Pan, T.-T. Shen, Computation of incompressible flows with immersed bodies by a simple ghost cell method, *Int. J. Numer. Meth. Fluids* 60 (2009) 1378–1401.
- [24] P. A. Berthelsen, O. M. Faltinsen, A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries, *J. Comput. Phys.* 227 (2008) 4354–4397.
- [25] E. Balaras, Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, *Comp. Fluids* 33 (2004) 375–404.
- [26] F. Gibou, R. P. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, *J. Comput. Phys.* 176 (2002) 205–227.
- [27] A. Abdelsamie, G. Fru, T. Oster, F. Dietzscht, G. Janiga, D. Thévenin, Towards direct numerical simulations of low-Mach number turbulent reacting and two-phase flows using immersed boundaries, *Comp. Fluids* 131 (2016) 123–141.

- [28] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes in FORTRAN, 2nd ed., Cambridge University Press, New York, 1992.
- [29] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, *J. Comput. Phys.* 59 (1985) 308–323.
- [30] D. L. Brown, R. Cortez, M. L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, *J. Comput. Phys.* 168 (2001) 464–499.
- [31] J. Lee, D. You, An implicit ghost-cell immersed boundary method for simulations of moving body problems with control of spurious force oscillations, *J. Comput. Phys.* 233 (2013) 295–314.
- [32] Y. Gorsse, A. Iollo, H. Telib, L. Weynans, A simple second order Cartesian scheme for compressible Euler flows, *J. Comput. Phys.* 231 (2012) 7780–7794.
- [33] A. Mark, B. G. van Wachem, Derivation and validation of a novel implicit second-order accurate immersed boundary method, *J. Comput. Phys.* 227 (2008) 6660–6680.
- [34] A. Shinn, M. Goodwin, S. Vanka, Immersed boundary computations of shear- and buoyancy-driven flows in complex enclosures, *Int. J. Heat Mass Transf.* 52 (2009) 4082–4089.
- [35] S. Laizet, E. Lamballais, J. Vassilicos, A numerical strategy to combine high-order schemes, complex geometry and parallel computing for high

resolution DNS of fractal generated turbulence, *Comp. Fluids* 39 (2010) 471–484.

- [36] J. Williamson, Low-storage Runge-Kutta schemes, *J. Comput. Phys.* 35 (1980) 48–56.
- ⁵ [37] K. E. Niemeyer, N. J. Curtis, C.-J. Sung, pyJac: Analytical Jacobian generator for chemical kinetics, *Comput. Phys. Commun.* 215 (2017) 188–203.
- [38] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.* 79 (1988) 12–49.
- ¹⁰ [39] J. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press: Cambridge, 1999.
- [40] J. Sethian, Evolution, implementation, and application of level set and fast marching methods for advancing fronts, *J. Comput. Phys.* 169 (2001) 503–555.
- ¹⁵ [41] S. Osher, J. A. Sethian, *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag: New York, 2003.
- [42] N. Li, S. Laizet, 2DECOMP&FFT - a highly scalable 2D decomposition library and FFT interface, in: Cray User's Group 2010 Conference, 2010.
- ²⁰ [43] R. Mittal, S. Balachandar, Effect of three-dimensionality on the lift and drag of nominally two-dimensional cylinder, *Phys. Fluids* 7 (1995) 1841–1865.

- [44] C. Williamson, Vortex dynamics in the cylinder wake, *Annu. Rev. Fluid Mech.* 8 (1996) 477–539.
- [45] D. Barkley, R. D. Henderson, Three-dimensional Floquet stability analysis of the wake of a circular cylinder, *J. Fluid Mech.* 322 (1996) 215–241.
- 5 [46] M.-C. Lai, C. S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* 160 (2000) 705–719.
- [47] L. Baranyi, Computation of unsteady momentum and heat transfer from a fixed circular cylinder in laminar flow, *J. Comput. Appl. Mech.* 10 4 (2003) 13–25.
- [48] C. H. K. Williamson, Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers, *J. Fluid Mech.* 206 (1989) 579–627.
- 15 [49] H. Dütsch, F. Durst, S. Becker, H. Lienhart, Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan–Carpenter numbers, *J. Fluid Mech.* 360 (1998) 249–271.
- [50] S.-G. Cai, A. Ouahsine, J. Favier, Y. Hoarau, Moving immersed boundary method, *Int. J. Numer. Meth. Fluids* 85 (2017) 288–323.
- 20 [51] Z. J. Wang, Two dimensional mechanism for insect hovering, *Phys. Rev. Lett.* 85 (2000) 2216–2219.
- [52] S. Xu, Z. J. Wang, An immersed interface method for simulating the

- interaction of a fluid with moving boundaries, *J. Comput. Phys.* 216 (2006) 454–493.
- [53] J. Yang, F. Stern, A simple and efficient direct forcing immersed boundary framework for fluid–structure interactions, *J. Comput. Phys.* 231 (2012) 5029–5061.
- [54] R. Clift, J. Grace, M. Weber, *Bubbles Drops and Particles*, Academic Press, New York, 1978.
- [55] H. Sakamoto, H. Haniu, The formation mechanism and shedding frequency of vortices from a sphere in uniform shear flow, *J. Fluid Mech.* 287 (1995) 151–171.
- [56] D. Ormières, M. Provansal, Transition to turbulence in the wake of a sphere, *Phys. Rev. Lett.* 83 (1999) 80–83.
- [57] R. Mittal, A Fourier–Chebyshev spectral collocation method for simulating flow past spheres and spheroids, *Int. J. Numer. Meth. Fluids* 30 (1999) 921–937.
- [58] T. A. Johnson, V. C. Patel, Flow past a sphere up to a Reynolds number of 300, *J. Fluid Mech.* 378 (1999) 19–70.
- [59] S. Taneda, Experimental investigation of the wake behind a sphere at low Reynolds numbers, *J. Phys. Soc. Jpn.* 11 (1956) 1104–1108.
- [60] G. Janiga, P. Berg, S. Sugiyama, K. Kono, D. Steinman, The computational fluid dynamics rupture challenge 2013—Phase I: Prediction of

rupture status in intracranial aneurysms, Am. J. Neuroradiol. 36 (2015) 530–536.

- [61] S. Curtis, M. Bradley, P. Wilde, J. Aw, S. Chakrabarti, M. Hamilton, R. Martin, M. Turner, A. Stuart, Results of screening for intracranial aneurysms in patients with coarctation of the aorta, Am. J. Neuroradiol. 33 (2012) 1182–1186.
- [62] P. Berg, C. Roloff, O. Beuing, S. Voss, S.-I. Sugiyama, N. Aristokleous, A. S. Anayiotos, N. Ashton, A. Revell, N. W. Bressloff, A. G. Brown, B. Jae Chung, J. R. Cebral, G. Copelli, W. Fu, A. Qiao, A. J. Geers, S. Hodis, D. Dragomir-Daescu, E. Nordahl, Y. Bora Suzen, M. Owais Khan, K. Valen-Sendstad, K. Kono, P. G. Menon, P. G. Albal, O. Mierka, R. Münster, H. G. Morales, O. Bonnefous, J. Osman, L. Goubergrits, J. Pallares, S. Cito, A. Passalacqua, S. Piskin, K. Pekkan, S. Ramalho, N. Marques, S. Sanchi, K. R. Schumacher, J. Sturgeon, H. Švihlová, J. Hron, G. Usera, M. Mendina, J. Xiang, H. Meng, D. A. Steinman, G. Janiga, The computational fluid dynamics rupture challenge 2013—Phase II: Variability of hemodynamic simulations in two intracranial aneurysms, J. Biomech. Eng. 137 (2015) 121008/1–13.

Declaration of interests

- ✓ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

The authors declare that there is no conflict of interest for this work.