Applied Mathematical Modelling 99 (2021) 606–627

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

A simple and scalable immersed boundary method for high-fidelity simulations of fixed and moving objects on a Cartesian mesh

Athanasios Emmanouil Giannenas*, Sylvain Laizet

Department of Aeronautics, Imperial College London, London SW7 2AZ, United Kingdom

ARTICLE INFO

Article history:
Received 10 September 2020
Revised 11 June 2021
Accepted 21 June 2021
Available online 16 July 2021

Immersed Boundary Method
High-performance computing
Finite-Difference Scheme
Moving Body Problems
Cartesian Mesh

ABSTRACT

A simple and scalable Immersed Boundary Method based on cubic spline reconstructions is presented for high-fidelity simulations of immersed objects in a turbulent flow on a Cartesian mesh. The novelty of the proposed IBM lies in its simplicity, accuracy, scalability and its ability to simulate both fixed and moving immersed objects. The new IBM is thoroughly validated against a 1D benchmark, with the 2D flow around a cylinder at $Re = 40$ and 300 and the 3D flow around a sphere at $Re = 300$ and $Re = 3700$. Convergence studies and detailed error maps showing the spatial distribution of the velocity L^2-Norm error compared to a spectral reference solution for the cylinders at $Re = 40$ show the robustness of the proposed method. The cost and performance of the method are also presented for multi-billion mesh node simulations with up to 65,536 computational cores. The potential of the method in handling multiple moving objects for practical applications is demonstrated with the control of a square bluff body wake by two rear pitching flaps.

1. Introduction

Performing high-fidelity turbulent simulations with fixed or moving objects immersed in a fluid remains a great challenge in the area of Computational Fluid Dynamics (CFD), particularly when High Performance Computing resources need to be used. Two main strategies have been developed so far: body conforming methods using e.g. finite-volume, finite/spectral element approaches and Immersed Boundary Methods (IBMs). IBMs for moving objects are very competitive and computationally attractive as no re-meshing is required, and they can potentially be easily combined with high-order methods. Fourier or Chebyshev-based spectral methods are the most accurate methods to simulate turbulent flows with the ability to capture a wide range of turbulent scales for limited degrees of freedom. IBMs, often based on Cartesian meshes, only require the addition of an extra forcing term in the governing equations in order to impose the desired boundary conditions on the velocity field. Since the method was first introduced by the pioneering work of Peskin [1–3] with simulations of a beating heart, many IBMs have been developed over the years by the scientific community. A thorough review of various IBMs and their applications can be found in the review of Mittal and Iaccarino [4]. Unfortunately, when an IBM is used for simulations of a moving object, spurious numerical oscillations often appear near the object, the well-known spurious force oscillations (SFOs), with a significant impact on the quality of the solution. These SFOs can be controlled by changing the spatial and

* Corresponding author.
E-mail address: a.giannenas17@imperial.ac.uk (A.E. Giannenas).

https://doi.org/10.1016/j.apm.2021.06.026

temporal discretisation [5,6], by smoothing the fluid-solid interface [5,7–10], by adding a mass source/sink term [11,12], by reconstructing the velocity field in time and/or inside the moving object [13,14], or by using an non-inertial reference frame [15].

The aim of the present study is to present a simple and robust IBM capable of performing high-fidelity simulations using HPC resources, with fixed and moving objects. The concept is to combine a sharp-interface (direct forcing) Alternating Direction Reconstruction Immersed Boundary Method (ADR-IBM) with high-order finite-difference schemes on a Cartesian mesh. The method is implemented, assessed and validated in the high-order finite-difference framework Xcompact3D [16,17]. It is based on a powerful 2D domain decomposition for simulations on HPC facilities and it can scale with up to 10^5 computational cores [18]. The first constraint when developing the ADR-IBM was to ensure its full compatibility with a 2D domain decomposition to allow for large-scale simulations. The second constraint was related to simplicity and cost. Most of recent IBMs are fairly complicated to implement and can potentially be expensive and unsuitable for large-scale 3D problems.

The paper is organised as follows: Section 2 describes the governing equations and numerical methods employed in Xcompact3D. Section 3 presents the traditional No-Reconstruction IBM (NR-IBM) and the proposed Alternating Direction Reconstruction IBM (ADR-IBM). Section 4 focuses on a validation of the proposed method for a simple 1D benchmark. Section 5 presents a thorough validation of the proposed method for the 2D flow around a cylinder at $Re = 40$. Additionally, a convergence study is presented along with detailed error maps. Moreover, further validation of the unsteady 2D flow over a moving cylinder at $Re = 300$ is also presented in Section 6. In sections 7 and 8, 3D simulations are also performed for the flow around a sphere at $Re = 300$ and $Re = 3700$ with a detailed scalability study on Tier-1/0 supercomputer facilities. Finally, the major conclusions from the presented work are summarised in Section 10 along with a description of the future outlook.

2. Numerical methods

2.1. Governing equations

The governing equations are the forced incompressible Navier-Stokes equations:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{1}{2}[\nabla(\boldsymbol{u}\otimes\boldsymbol{u}) + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}] + \frac{1}{\rho}\nabla p - \nu\nabla^2\boldsymbol{u} + \boldsymbol{f} = 0, \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{2}$$

where $\boldsymbol{u}(\boldsymbol{x},t)$ is the velocity field, $p(\boldsymbol{x},t)$ is the pressure field, ρ is the constant density of the fluid, ν is the kinematic viscosity and $\boldsymbol{f}$ is an extra forcing term (related to the immersed boundary method) which acts in the vicinity of the boundary's interface to impose the required boundary conditions on the velocity field. Note finally that the equations are written in skew-symmetric form in order to reduce aliasing errors [19]. The computational domain Ω consists of n solid regions $\Omega_s = \Omega_{s1}, \Omega_{s2}, \ldots, \Omega_n$ that include the immersed boundary surfaces $\partial\Omega_s = \partial\Omega_{s1}, \partial\Omega_{s2}, \ldots, \partial\Omega_n$ (for n immersed objects) and fluid region Ω_f (i.e. $\Omega = \Omega_s \cup \Omega_f$).

2.2. Spatial discretisation

Finite-difference sixth-order accurate compact schemes are employed in `Xcompact3D` for the spatial discretisation of the convective and diffusive terms ($\boldsymbol{F}^k$ in Eq. (6)). Such schemes are very attractive in a Direct Numerical Simulation (DNS) context due to their ability to provide accurate results using a moderate number of degrees of freedom when compared to more conventional low-order schemes. They provide a quasi-spectral accuracy while requiring a compact stencil of five to seven points for the primitive function f, and three for its first f' and second f'' derivatives [20]. The rather large stencils allow the schemes to slightly over-dissipate at the highest wave numbers through the second derivative (in the spectral range where even a high-order finite-difference scheme becomes inaccurate). These schemes are therefore an effective tool to control aliasing errors [21,22]. They are compatible with periodic, free-slip, no-slip and open boundary conditions. Further details can be found in [16].

2.3. Time integration

A three-step fractional step method is employed for the time integration of the momentum equation (Eq. (1)):

$$\frac{\boldsymbol{u}^{*} - \boldsymbol{u}^k}{\Delta t} = a_k\boldsymbol{F}^k + b_k\boldsymbol{F}^{k-1} - c_k\nabla\tilde{p}^k + c_k\tilde{\boldsymbol{f}}^{k+1}, \tag{3}$$

$$\frac{\boldsymbol{u}^{**} - \boldsymbol{u}^{*}}{\Delta t} = c_k\nabla\tilde{p}^k, \tag{4}$$

$$\frac{\boldsymbol{u}^{k+1} - \boldsymbol{u}^{**}}{\Delta t} = -c_k\nabla\tilde{p}^{k+1}, \tag{5}$$

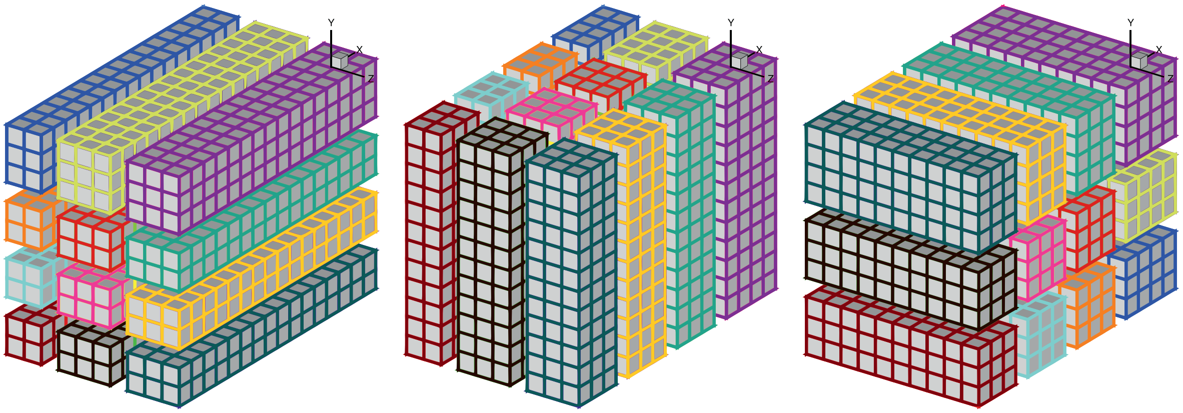

Fig. 1. 2D domain decomposition example with 4 × 3 computational cores and corresponding MPI processes with from left to right, the x-, y- and z-pencils.

where:

$$\boldsymbol{F}^k = -\boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nu \nabla^2 \boldsymbol{u}, \tag{6}$$

and

$$\tilde{p}^{k+1} = \frac{1}{c_k \Delta t} \int_{t_k}^{t_{k+1}} p dt, \quad \tilde{\boldsymbol{f}}^{k+1} = \frac{1}{c_k \Delta t} \int_{t_k}^{t_{k+1}} \boldsymbol{f} dt, \tag{7}$$

where $\boldsymbol{u}^*$ and $\boldsymbol{u}^{**}$ represent the predicted and intermediate velocity fields accordingly. The intermediate velocity obtained through Eq. (4) does not satisfy the divergence-free condition (Eq. (2)). Hence, it is corrected with the pressure at the new time-step via Eq. (5) to obtain the divergence-free velocity $\boldsymbol{u}^{k+1}$. Further, the constants (a_k, b_k and $c_k = a_k + b_k$) correspond to the coefficients of the numerical schemes on n_k sub-time-steps with $k = 1, \dots, n_k$ and a full time-step $\Delta t = t_{n+1} - t_n$. For the present study, a second-order Adams-Bashforth scheme is used with $n_k = 1$, $a_1 = 3/2$ and $b_1 = -1/2$. The time-steps selected for all studies are sufficiently small so that time-integration errors can be neglected (no difference could be observed with a higher-order time-integration scheme). It should be noted that the variables with tilde ($\tilde{p}^{k+1}, \tilde{\boldsymbol{f}}^{k+1}$) indicate a time averaged value on the given sub-time-step $c_k \Delta t$.

In Eq. (5), the pressure field at the new time-step $\tilde{p}^{k+1}$ is required for the correction of the intermediate velocity field $\boldsymbol{u}^{**}$ in order to obtain the divergence-free velocity $\boldsymbol{u}^{k+1}$. The required pressure field $\tilde{p}^{k+1}$ can be obtained by solving a Poisson's equation (see Eq. (8)) which also ensures the incompressibility constraint (see Eq. (2)). It should be noted that the conventional Poisson's equation presented here is suitable for the fluid regions Ω_f but requires further modifications for the solid regions Ω_s where the immersed boundary forcing is acting. Further details are presented in Section 3.

The treatment of the incompressibility condition is a considerable challenge for incompressible flow-solvers based on a fractional-step method as it requires the solution of a Poisson's equation

$$\nabla \cdot \nabla \tilde{p}^{k+1} = \frac{\nabla \cdot \boldsymbol{u}^{**}}{c_k \Delta t} \tag{8}$$

with potentially significant computational overhead. The cost can be particularly penalising when high-order schemes are combined with iterative techniques. Contrastingly, the inversion of a Poisson's equation in Fourier space (when possible) can significantly reduce the cost and simplify the numerical implementation. Using the concept of the modified wave number [20], it is possible to define strictly equivalent operators in Fourier space to sixth-order finite-difference schemes. Hence, in `Xcompact3D`, the spatial discretisation of the pressure field which is defined on a half-staggered mesh to reduce spurious oscillations, is performed in Fourier space through Fast Fourier Transforms (FFTs).

2.4. 2D domain decomposition

In `Xcompact3D`, the 3D computational domain is partitioned in 1D 'pencils' (X-pencil, Y-pencil and Z-pencil) as shown in Fig. 1. The high-order schemes used for the derivative and interpolation operations in each (X-, Y-, Z-) directions are performed in the corresponding pencils and no communication is required between the pencils during these calculations. Global transpose operations to switch data from one pencil decomposition to another are performed through the command `MPI_ALLTOALL(V)` with the MPI library. This strategy allows `Xcompact3D` to scale up to 10^5 computational cores [17,18].

3. Immersed boundary method (IBM)

3.1. No-Reconstruction IBM

The term No-Reconstruction Immersed Boundary Method (NR-IBM) is used to describe the most basic form of the direct forcing IBM which will be used as a reference for the validation of the novel IBM described in Section 3.2. Following the definition of the direct forcing IBM, the forcing term which consists of the acceleration, inertial, viscous and pressure force components, can be defined as

$$c_k \tilde{\boldsymbol{f}}^{k+1} = \varepsilon \left(\frac{\boldsymbol{u}_{IB} - \boldsymbol{u}^k}{\Delta t} - a_k \boldsymbol{F}^k - b_k \boldsymbol{F}^{k-1} + c_k \nabla \tilde{p}^k \right), \tag{9}$$

where $\boldsymbol{u}_{IB}$ corresponds to the imposed immersed boundary velocity and ε to a scalar field equal to zero in the fluid and to 1 in the solid regions:

$$\varepsilon = \begin{cases} 1, & \text{if } x \in \Omega_s \\ 0, & \text{if } x \in \Omega_f. \end{cases} \tag{10}$$

The scalar field ε masks the effect of the forcing in the fluid and is used to ensure that the correct boundary conditions are imposed for the solid regions. ε can be defined on the same mesh as the one used for the simulation or it can be defined on a finer mesh in order to have a more accurate localisation of the immersed boundary interfaces. Note also that an analytical equation, when available, can be used to determine the exact location of the immersed boundary interface (for example for a cylinder or an airfoil). It will be shown later on that the resolution used to define ε can strongly influence the quality of the computed solution. For the no-reconstruction IBM, $\boldsymbol{u}_{IB} = 0$ for fixed solid immersed objects. By substituting Eq. (9) in Eq. (3), it is possible to define a relation between the predicted velocity $\boldsymbol{u}^{**}$ and the desired immersed boundary velocity $\boldsymbol{u}_{IB}$:

$$\frac{\boldsymbol{u}^{**} - \boldsymbol{u}^k}{\Delta t} = \frac{\boldsymbol{u}_{IB} - \boldsymbol{u}^k}{\Delta t} \quad \Leftrightarrow \quad \boldsymbol{u}^{**} = \boldsymbol{u}_{IB}. \tag{11}$$

Since $\boldsymbol{u}_{IB}$ does not necessarily satisfy the divergence-free condition, the Poisson equation needs to be modified as follows:

$$\nabla \cdot \nabla \tilde{p}^{k+1} = \frac{\nabla \cdot [(1-\varepsilon)\boldsymbol{u}^{**} + \varepsilon \boldsymbol{u}_{IB}]}{c_k \Delta t}. \tag{12}$$

The conventional Poisson's equation is recovered in the fluid region Ω_f where $\varepsilon = 0$, while the Laplace equation is recovered with a first-order approximation across the boundary interface $\partial\Omega_s$ (since $\boldsymbol{u}^{**} \to \boldsymbol{u}_{IB}$ as $\varepsilon \to 1$ with first-order accuracy) and recovered exactly inside the solid region Ω_s where $\varepsilon = 1$. Eq. (11) results in $\varepsilon \boldsymbol{u}^{**} = 0$ and $\varepsilon \boldsymbol{u}^{**} = \boldsymbol{u}_m$ for fixed and moving boundaries accordingly (where $\boldsymbol{u}_m$ is the moving velocity of an immersed object). This abrupt imposition of the boundary conditions in the solid region Ω_s can lead to spurious numerical oscillations because of the discontinuity of the velocity field, especially in the context of high-order schemes. Further, the velocity of the fluid approaches the desired immersed boundary velocity with a first-order accuracy ($\boldsymbol{u}^{**} \to \boldsymbol{u}_{IB}$ as $\varepsilon \to 1$). Additionally, the inability of this approach to accurately define the immersed boundary interface can result in a stair-case representation of the object (as the scalar field ε is defined on the same mesh as the one used for the simulation). This problem is illustrated in Fig. 2 (left) for a cylinder. The lost solid area can be seen in grey and the shape of the simulated object can be seen in yellow. If the resolution is sufficiently fine this first-order 'shape' approximation can be acceptable but at marginal resolutions the resulting stair-case geometry could lead to improper mass conservation with an adverse effect on the quality of the solution.

3.2. Alternating Direction Reconstruction IBM (ADR-IBM)

In order to accurately define the immersed boundary interface, a mirroring reconstruction IBM was introduced by Mohd-Yusof [23,24]. It was designed to obtain velocities only for the closest solid nodes of the immersed boundaries, using a second-order multi-linear mirroring interpolation to obtain an artificial velocity in the entire solid domain. Parnaudeau et al. [25,26] extended this mirroring reconstruction approach to high-order finite-difference compact schemes and successfully used it in Xcompact3D for the flow around a fixed cylinder [26], with a reconstruction of the flow inside the cylinder in the wall normal directions. This method proved well suited for simplified geometries like a cylinder but poses various difficulties when more complex objects are considered. For example, for geometries with sharp corners, some solid mesh nodes can be equidistant to more than one boundary which results in multiple possible wall normal directions. Fig. 2 (right) illustrates this problem on a simple 2D geometry with sharp corners where the black dotted lines represent successfully identified wall normal directions while the red lines depict a conflicting case. It is also important to point out that any mirroring reconstruction IBM requires potentially costly interpolations (for instance to find the velocity values in the wall normal directions for a cylinder), and are often not compatible with efficient parallelisation strategies. Most of the reconstruction strategies presented in the literature focus on 2D simulations or 3D simulations with a limited number of degrees of freedom. As far as high-order compact schemes are concerned, Gronskis and Artana [27] proposed a simplified sharp interface immersed

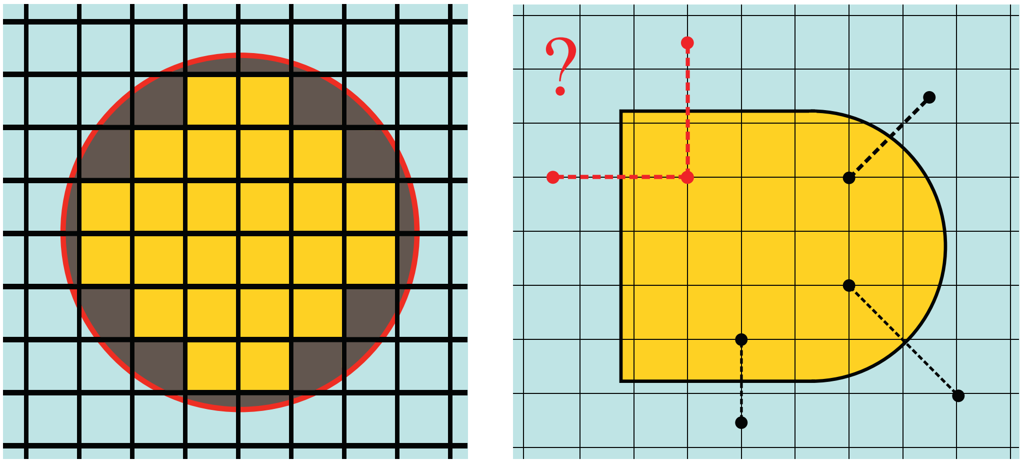

Fig. 2. Left: schematic view of a cylinder on a Cartesian mesh with a coarse resolution. The yellow colour depicts the simulated cylinder while the grey shows the 'lost' solid volume. Right: illustration of problematic points (red) on a simplified geometry with sharp corners when global 2D interpolations are used. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

boundary method in Xcompact3D to study 3D anguilliform swimming. However, the wall normal interpolation had to be simplified by entirely omitting the image points in order to ensure compatibility with the parallel strategy of the solver. Due to this simplification, the order of the solution was reduced near the immersed boundary interface. More recently, Gautier et al. [28] proposed an alternating direction forcing IBM for fixed objects based on 1D and formally second-order Lagrangian polynomial interpolations. The idea is to reconstruct the flow in each spatial direction to be compatible with the pencil domain decomposition strategy of Xcompact3D. However, the Lagrangian interpolations could produce non-physical velocities of extreme magnitudes inside the solid regions leading in some cases to stability constraints and to issues with mass conservation.

Inspired by the IBM introduced by Gautier et al. [28], a new Alternating Direction Reconstruction IBM (ADR-IBM) is proposed in the present study, based on a series of 1D cubic spline reconstructions to obtain the flow inside solid objects in each spatial direction while ensuring a targeted velocity at the wall (zero for fixed objects). The reconstructions are performed before each derivative calculation in the relevant direction. The ADR-IBM owes its name to the alternating reconstructions which are performed in each spatial direction in the corresponding pencils to ensure the continuity of the velocity function. When global 2D or 3D interpolations are used, the continuity of the velocity field is only ensured in one direction (often the wall normal direction) and not in the directions of the derivatives, which could lead to spurious oscillations, not to mention the extra computational overhead for the interpolations. As the ADR-IBM relies on 1D interpolations only, it is fully compatible with the 2D domain decomposition used in Xcompact3D. Furthermore, the method does not require any special treatment of the spatial discretisation schemes such as reducing the order of accuracy close to the immersed boundaries [29]. The simplicity of the method allows for a trivial numerical implementation and is fully compatible with immersed moving objects.

Following the same definition for the forcing term as in Eq. (9) and substituting the immersed boundary velocity $\boldsymbol{u}_{IB}$ for the reconstructed velocity $\boldsymbol{u}_{re}$, $\boldsymbol{u}^{**}$ can be evaluated as

$$\frac{\boldsymbol{u}^{**}-\boldsymbol{u}^{k}}{\Delta t}=\frac{\boldsymbol{u}_{re}-\boldsymbol{u}^{k}}{\Delta t} \quad \Leftrightarrow \quad \boldsymbol{u}^{**}=\boldsymbol{u}_{re}. \tag{13}$$

The modified Poisson equation in Eq. (12) can be used by the ADR-IBM to obtain the pressure field while ensuring the incompressibility constraint. The conventional Poisson equation in Eq. (8) is recovered in the fluid region Ω_f where $\varepsilon = 0$. Finally, the Laplace equation is recovered exactly inside the solid region Ω_s where $\varepsilon = 1$.

3.3. Cubic spline reconstruction

Considering a fully immersed geometry, an even number of boundaries can be identified in each x-, y- and z-direction in a 1D representation. The desired variable (here the velocity field $\boldsymbol{u}_{re}$) in the solid domain (between the identified boundaries) can be interpolated via a cubic spline reconstruction to ensure not only the desired immersed boundary velocity $\boldsymbol{u}_{IB}$ at the corresponding IB location but also the continuity of the reconstructed function across the whole domain. A simplified example can be seen in Fig. 3 with an interpolated function between two identified boundaries. The closest mesh nodes (up to three) on each side of the immersed boundaries and the boundary interface points (IB_P) can be used as inputs for the interpolation. The identification of the boundary interface points can be achieved with an analytical (or an approximated) equation or by using approximate positions on a highly refined mesh. To avoid any numerical instabilities arising from the special case when the first input mesh node is extremely close to the boundary interface, the first mesh node can be

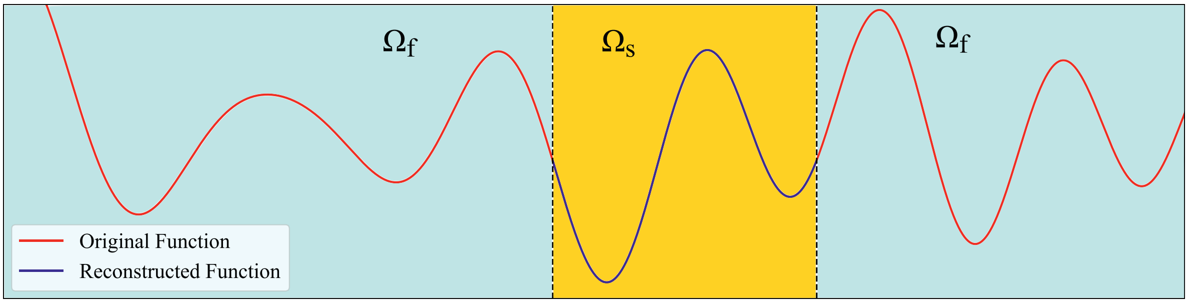

Fig. 3. Simplified example of a 1D reconstructed function in the solid domain Ω_s using a cubic spline approach.

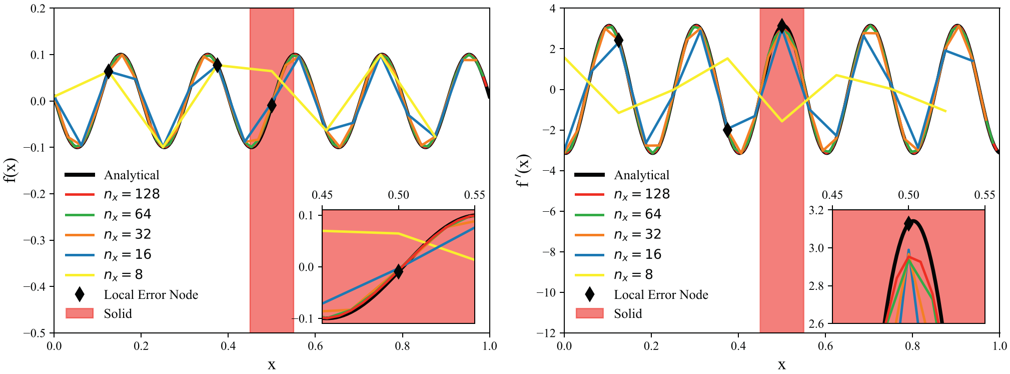

Fig. 4. Exact and reconstructed profiles of the function $f(x)$ and its first derivative $f'(x)$ for $0 \le x \le 1$, with a zoom in the solid region $0.45 \le x_{solid} \le 0.55$. The black diamonds correspond to the mesh nodes used for the calculation of the local errors.

ignored and the second, third and fourth mesh nodes automatically become the input ones. This option will be denoted as 'skip' or 'skipping' hereafter (for further information see Section 5.4). For the special case where one of the immersed boundaries coincides with the boundaries of the computational domain, a number of fictional input points are introduced outside the domain where the appropriate values are imposed. Hence, no special treatment for the boundaries is required during the reconstruction. The mathematical background of the cubic spline interpolation used for the ADR-IBM is presented in Appendix A.

4. 1D benchmark

Here a simple 1D test case is presented for the validation of the ADR-IBM. The periodic function $f(x) = 0.1\cos(10\pi(x + 0.04698674))$ is considered for $x \in [0, 1]$ with periodic boundary conditions. An immersed boundary is introduced in the middle of the domain $0.45 \le x_{solid} \le 0.55$. This function was chosen to ensure that the immersed boundary interfaces do not coincide with any of the mesh nodes. Two fluid points on each side of the immersed boundary interfaces are used as inputs for the spline interpolation (see Section 5.4 for a discussion on the number of input points used for the cubic spline reconstruction). Fig. 4 shows the exact and calculated $f(x)$, $f'(x)$ for five different resolutions ($n_x = 8, 16, 32, 64, 128$) along with the three mesh nodes used for the calculation of local error. The local error is computed at three locations and is defined as $E_L = f'_{cal} - f'_{ref}$, where f'_{cal} corresponds to the calculated derivative while f'_{ref} corresponds to the exact value. The global error can be computed as

$$L^2 = \sqrt{\frac{1}{n_x}\sum_{i=1}^{n_x}[f'_{cal}(x_i) - f'_{ref}(x_i,)]^2}.$$

Fig. 5 shows the convergence of the global L^2 and local errors at three different locations (two in the fluid and one in the solid). As it can be observed, the global convergence is third-order as dictated by the convergence of the cubic spline reconstructions which degrade the solution near the immersed boundary interfaces (when the solid region is removed, the global convergence is sixth-order). At best, a fourth-order convergence can be expected from cubic spline reconstructions [30,31]. For the present ADR-IBM, the global convergence will depend on the dimension of the solid regions as the reconstructions are purely based on inputs from fluid nodes, not inputs from solid nodes. As a result, the local error inside the solid region ($x = 0.5$) in the present example does not converge and is not expected to converge. This can also be observed in Fig. 4 where the zoomed-in regions clearly demonstrate that the reconstructed functions in the solid region do

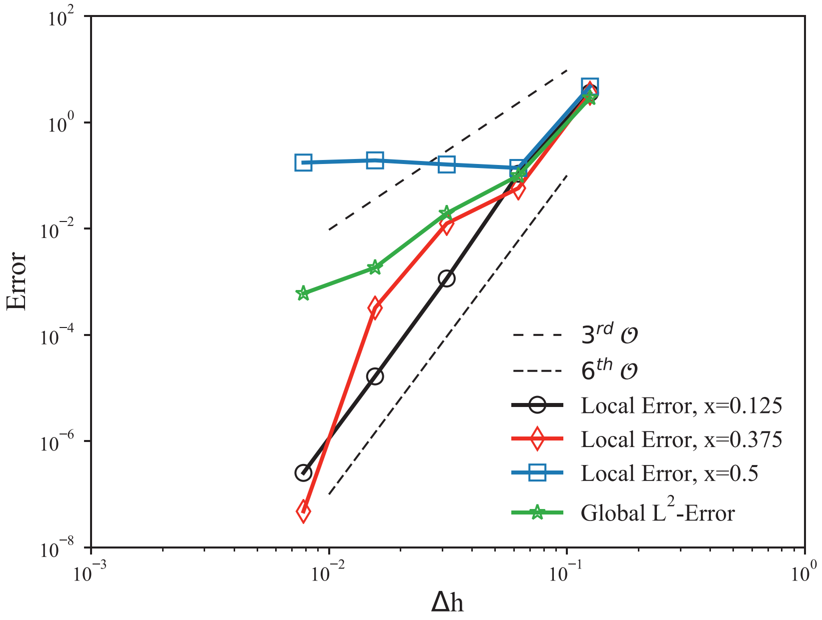

Fig. 5. Convergence of the global L^2 and local errors. Δh represents the mesh spacing.

not approach the exact solution. The sixth-order convergence is recovered further away from the immediate vicinity of the boundary interfaces, as demonstrated by the local convergence of the mesh nodes located at $x = 0.125$ and $x = 0.375$. It confirms that the use of high-order schemes is still relevant with the ADR-IBM, despite a local loss of accuracy close to the solid regions. The sixth-order accuracy and its associated quasi-spectral properties are preserved in the fluid regions, except in the immediate vicinity of the immersed boundary interfaces. In the context of the Navier-Stokes equations, the pressure field, for which an IBM does not impose any value inside the solid regions, will limit the global convergence rate to second-order for the velocity field, see Section 5.3. None of the current IBMs can achieve high-order accuracy close to the walls of an immersed object because the Navier-Stokes equations cannot be satisfied without a specific treatment of the pressure field (the pressure gradients cannot balance the viscous forces). It is the main limiting factor for a global convergence rate of more than second-order for the velocity field [4].

5. Flow around a cylinder at $Re = 40$

The steady flow over a fixed and a moving cylinder is considered here as a validation of the ADR-IBM approach combined with the Navier-Stokes equations, and compared with the No-Reconstruction IBM. The present data will be compared with the reference solution provided by Gautier et al. [32]. This solution was obtained with pseudo-spectral methods in cylindrical coordinates, with a particular treatment for the boundary conditions in order to mimic an infinite flow domain. In the azimuthal direction, the reference solution was expanded in Fourier series while the Chebyshev collocation method was used in the radial direction. The time advancement is second-order accurate, implicit in time for the viscous term, with a Chorin-Temam prediction projection scheme for the divergence-free condition. No slip boundary conditions were imposed on the cylinder and a customised outflow boundary condition was imposed on the far field, following a procedure developed by Hasan et al. [33], for which the radial variation of the velocity field at large distances from the cylinder is assumed to be proportional to $1/D^2$ (see also Gautier et al. [32] for more details). Note finally that using a very refined mesh with `Xcompact3D` to produce a reference solution would not have allowed each type of error contribution to be distinguished, for example those associated with differentiation from those connected to the boundary condition approximations (as it will be seen in the following paragraphs, the errors at the boundaries are accurate up to zero machine, as the boundary conditions are exact).

The flow is characterised by its Reynolds number $Re = u_\infty D/\nu = 40$, where u_∞ is the free-stream velocity, D is the diameter of the cylinder and ν is the kinematic viscosity of the fluid. As the fluid passes around the cylinder, it separates, resulting in a steady recirculation bubble, as seen in Fig. 6. A rectangular domain of dimensions $(L \times H = 20D \times 5D)$ is used for all simulations. The cylinder is placed at the centre of the domain $(x_0, y_0) = (10D, 2.5D)$ as shown in Fig. 6 for all fixed cases. The boundary conditions at the inlet, outlet, top and bottom are extracted from the interpolated reference solution in order to ensure zero error at all boundaries for a proper comparison with the reference data. The interpolated boundary conditions allow the selection of a reduced domain height $(H = 5D)$ (a preliminary study was used to show that the use of a reduced domain height $H = 5D$ results in a difference of less than 0.06% for the drag coefficient when compared with a

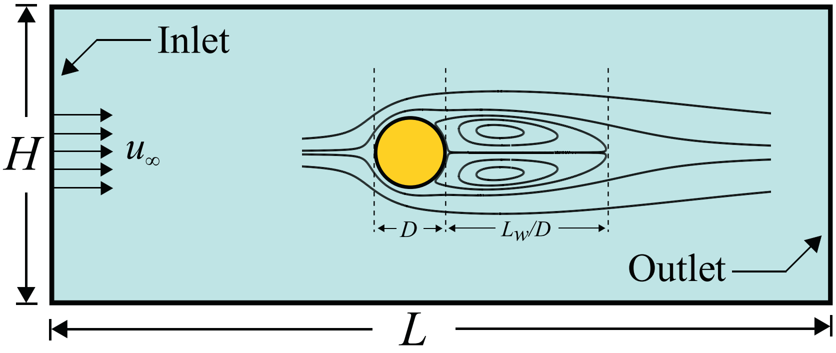

Fig. 6. Sketch of the computational domain for the steady flow around a cylinder at $Re = 40$. The recirculation bubble is characterised by the wake length parameter L_w/D.

domain height of 20D). For all moving cases, the cylinder is placed at $(x_1, y_1) = (14D, 2.5D)$ and the simulations are stopped when the cylinder has travelled four diameters (i.e. when it has reached the centre of the domain at $(x_0, y_0) = (10D, 2.5D)$). The interpolated boundary conditions are updated at each time-step to account for the temporally changing location of the cylinder. The flow field is initialised with a steady-state solution obtained for a fixed cylinder placed at (x_1, y_1). The free-stream velocity is $u_\infty = 0.5$ while the moving velocity of the cylinder is $\boldsymbol{u}_m = (-0.5, 0, 0)$. The x-component of the velocity of the initialisation field is also normalised to match the free stream velocity (i.e. $u_{init} = u_{restart} - 0.5$). This choice for the free-stream and moving velocities (instead of $\boldsymbol{u}_m = (-1, 0, 0)$ and $u_\infty = 0$ for instance) is explained in Section 5.4.

To demonstrate the attractiveness of the ADR-IBM, simulations for a fixed and a moving cylinder are performed for five different spatial resolutions: Case1 with 129×33, Case2 with 257×65, Case3 with 513×129, Case4 with 1025×257 and Case5 with 1537×385 mesh nodes. The cases for the NR-IBM for the fixed and moving cylinders are denoted as NR-F and NR-M while for the ADR-IBM as ADR-F and ADR-M accordingly. A very small time-step $\Delta t = 8 \times 10^{-5} D/u_\infty$ has been selected for all resolutions so that time-integration errors can be neglected. For each case the reference solution is interpolated on the corresponding mesh to obtain a virtually analytical solution for the entire computational domain. The L^2-Norm is computed using

$$L^2 = \sqrt{\frac{1}{n_x n_y} \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_y-1} [q_{cal}(x_i, y_j) - q_{ref}(x_i, y_j)]^2} \tag{14}$$

where q_{cal} corresponds to the calculated velocity while q_{ref} corresponds to the interpolated reference value at each node (x_i, y_j). It should be noted that the L^2-Norms presented in this section for the moving cylinder cases correspond to the error when the cylinder has moved 4D and its centre is located at $(x_0, y_0) = (10D, 2.5D)$ (i.e. the corresponding position of the fixed cylinder). The maximum error is defined as $E_{Max} = q_{cal}(x_i, y_j) - q_{ref}(x_i, y_j)$. For all simulations performed with the ADR-IBM, skipping is always enabled and the number of fluid input point is set to $N_{FP} = 2$, which appears to be the optimal set-up, as explained in Section 5.4.

5.1. Drag and recirculation bubble results for the fixed cylinder

A thorough comparison of the drag and recirculation bubble length is performed for the fixed cylinder. Here, the hydrodynamic drag forces F_D are obtained by utilising a momentum balance in a control volume around the cylinder. The drag coefficient C_D, is then calculated as follows:

$$C_D = \frac{F_D}{\frac{1}{2}\rho u_\infty^2 D}, \tag{15}$$

where $\rho = 1$ (incompressible flow). The present data are compared with the reference data of Gautier et al. [32] for which $C_D = 1.49$ and $L_w/D = 2.24$. The drag coefficients obtained for the NR-F cases at low resolution (1.70/1.52 for Case1/Case2) are fairly high but when the resolution is refined, the drag coefficient eventually converges to the reference value with 1.48/1.49 for Case4/Case5. Regarding the length of the recirculation bubble, the same trend is reported with values of 2.85/2.35 for Case1/Case2. On the finest mesh (Case5), the value obtained is nearly equal to the reference value (2.22 versus 2.24). This indicates that in order to obtain correct predictions for the drag coefficient and recirculation length with the NR-F method, the resolution needs to be very fine. Contrarily, the ADR-F method shows great potential as it produces excellent results for Case3 to Case5 (1.49 for the drag coefficient and 2.24 for the recirculation length), with a much faster convergence towards the reference data when the resolution is increased by comparison to the NR-F approach. For the ADR-F method, the resolution of Case3 is adequate to match the reference data of Gautier et al. [32], suggesting that the ADR method could allow the use of coarser meshes by comparison to the NR method.

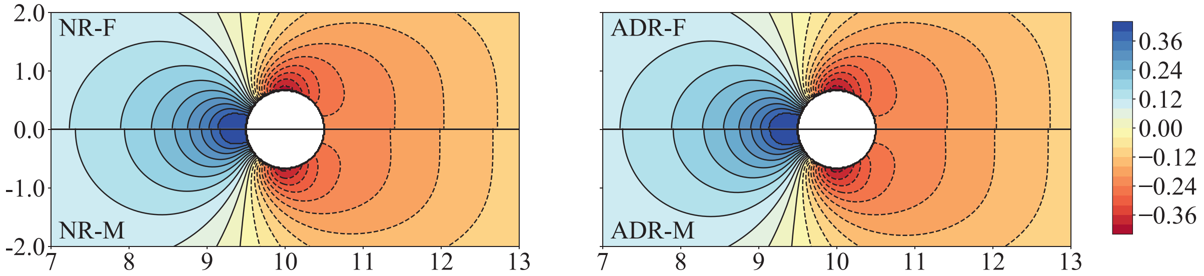

Fig. 7. Pressure fields and contour lines for the flow around a cylinder at $Re = 40$ obtained with the NR-IBM (left) and the ADR-IBM (right).

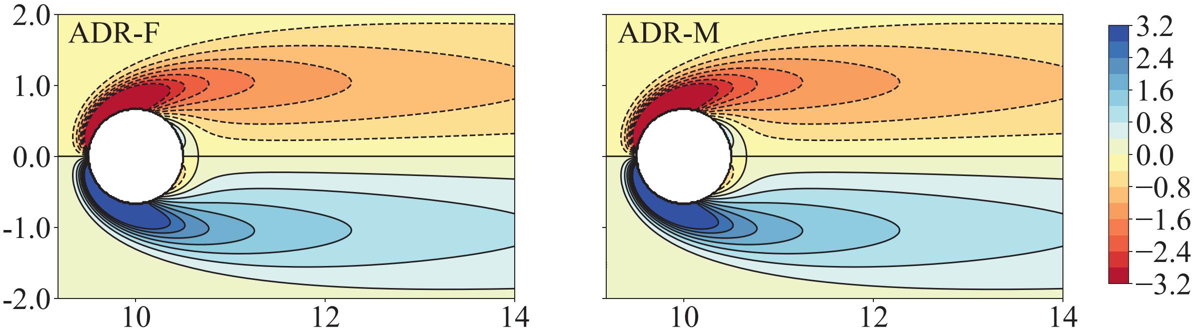

Fig. 8. Vorticity fields and contour lines (with levels set in the range of -3 to 3 with an increment of 0.4) for the flow around a cylinder at $Re = 40$ obtained for the ADR-F (left) and the ADR-M (right).

5.2. Pressure and vorticity contour fields

Fig. 7 shows the pressure fields and pressure contours for the NR-F/NR-M and ADR-F/ADR-M for the highest resolution (Case5). The aim here is to identify discrepancies between the fixed and moving cases for both methods, hence only half of the pressure fields are plotted for each simulation. Overall the pressure fields for the fixed and moving cylinders are almost identical for both methods. The only small difference can be seen upstream of the cylinder with slightly misaligned contour lines. This misalignment can also be seen downstream of the cylinder and it is less pronounced for the ADR approaches which perform slightly better than the NR-IBM for this mesh resolution.

Fig. 8 shows the vorticity contour only for ADR-F/ADR-M for the highest resolution (Case5). The two fields appear identical with no spurious numerical oscillations for the moving case, even if no filtering was used in the simulations. This demonstrates that the ADR-M performs as expected, and provides the same vorticity and pressure fields as the ADR-F, by Galilean invariance.

5.3. Convergence of the solution and 2D error maps

To provide a more in-depth comparison between the ADR-IBM and the NR-IBM, the decay of the velocity and pressure maximum errors and L^2-Norms are plotted for the fixed and moving cases in Fig. 9. As it can be seen in Fig. 9, the rates of convergence for the ADR-IBM and NR-IBM are second-order for the velocity and between first- and second-order for the pressure, in line with published data and other IBMs. Since nothing is done on the pressure field (only the desired values for the velocity fields are imposed), the rate of convergence of an IBM is limited to second-order for the velocity field. However, a discussion about the error levels remains crucial for the evaluation of the quality of an IBM.

As it can be seen in Fig. 9, the ADR-F provides lower global error levels when compared to the NR-F for the range of resolutions considered in this study. As expected, the error levels between the moving and fixed cases are very similar for both the ADR and NR approaches, but with a higher convergence rate for the ADR-IBM. The maximum error levels for the ADR-F and ADR-M are almost identical for all resolutions apart for Case1 where a small discrepancy can be seen in the pressure error. It is the maximum error which dictates the rate of convergence of the global L^2 error. The velocity errors for both approaches are similar for Case1 but diverge as the resolution is increased. The error levels for the ADR approach at the highest resolution are more than 5 times lower than the errors obtained with the NR approach. Impressively, the ADR-M with an intermediate resolution (Case3) provides the same velocity and pressure error levels as the NR-M does at the finest resolution (Case5). It can be concluded that the ADR-IBM requires only one third of the resolution compared to the NR-IBM, in order to obtain the same L^2-Norms. Taking into account the fact that the ADR-IBM is about 1.3 times slower than the NR-IBM (for the reconstruction of the velocity field at each time-step), the ADR-IBM is about 2.3 times cheaper than the NR-IBM, to obtain the same level of accuracy.

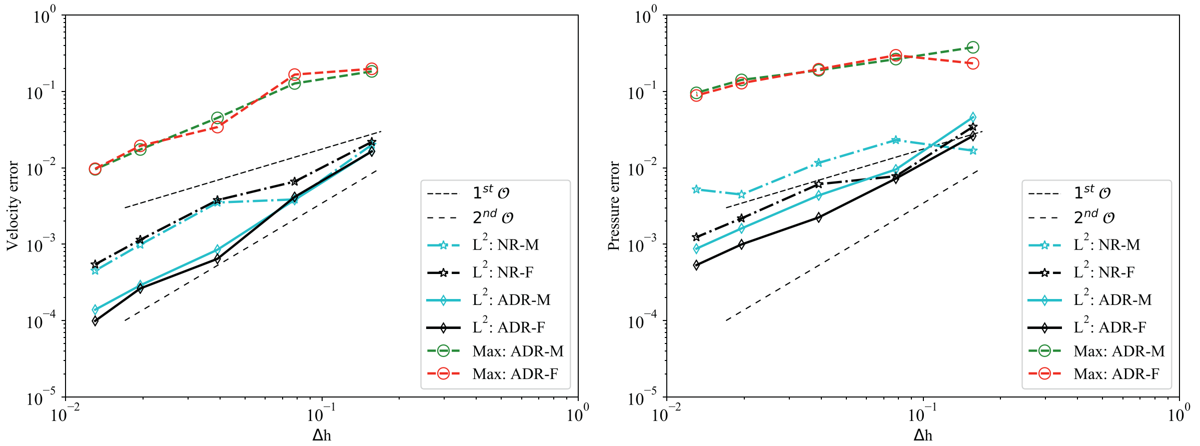

Fig. 9. Decay of the velocity (left) and pressure (right) maximum (Max) and L^2 errors for the flow around a cylinder at $Re = 40$. $\Delta h = \Delta x = \Delta y$ represents the mesh spacing.

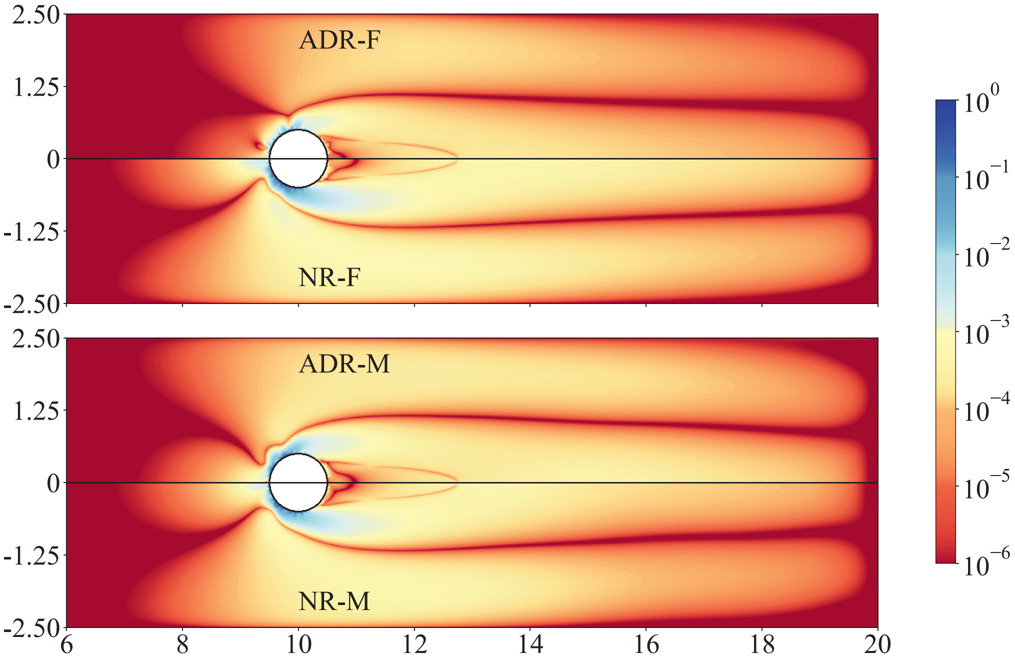

Fig. 10. 2D maps of the normalised velocity L^2 error for the flow around a cylinder at $Re = 40$. The data are for Case5 and are normalised by the maximum error.

The behaviour of the pressure errors is almost identical for all approaches with a convergence rate of around 1 and a slight difference in the error levels. This can be explained by the fact that the ADR-IBM reconstructs solely the velocity and not the pressure field. Since the pressure is treated in the same way for both methods the difference in error remains modest. Nevertheless, the ADR-IBM approach produces slightly lower error levels when compared to the NR-IBM. It should be noted that even though the error levels for the ADR-M are a little bit higher than the ADR-F ones, the convergence rate of the error remains unaffected. This increase could be attributed to pressure jumps which appear when a moving boundary moves into a new mesh node, see Fig. 13 and the related discussion for the flow around a moving cylinder at $Re = 300$. It should be noted that the jumps are not observable on the velocity field and have short lifetimes (about ten time-steps in the present study).

2D error maps for the velocity field can be used not only to identify differences between the various IBMs but also to locate where error levels are of significance. The velocity 2D error maps are presented in Fig. 10 for the fixed ADR/NR (top) and the moving ADR/NR cases (bottom). It can be seen that a zero machine error is observed on the boundaries as the boundary conditions are extracted from the reference solution of Gautier et al. [32]. The difference between the two

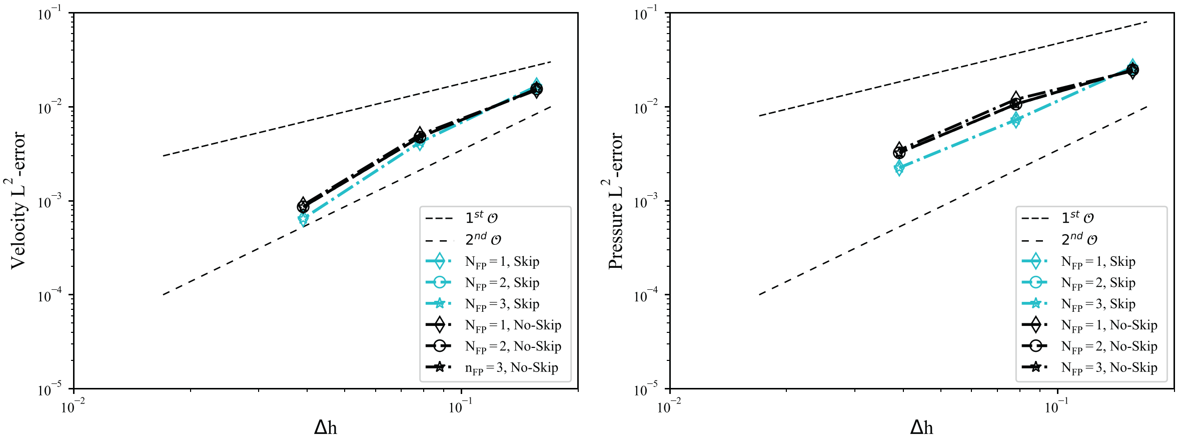

Fig. 11. Decay of the velocity (left) and pressure (right) L^2 error for the flow around a fixed cylinder at $Re = 40$ for the ADR-IBM with different number of interpolation fluid input points (N_{FP}), with and without skipping. $\Delta h = \Delta x = \Delta y$ represents the mesh spacing.

Table 1
Velocity and pressure L^2-Norms and relative error compared to the ADR-F obtained with the ADR-M for a range of moving boundary velocities u_m.

u_m	$U - L^2$-Er.	$P - L^2$-Er.	% U-Er.	% P-Er.
(0,0,0)	6.44×10^{-4}	2.24×10^{-3}	-	-
$(-0.50, 0, 0)$	8.47×10^{-4}	4.36×10^{-3}	23.9	48.6
$(-0.75, 0, 0)$	8.96×10^{-4}	5.95×10^{-3}	28.1	62.3
$(-1.0, 0, 0)$	9.38×10^{-4}	7.53×10^{-3}	31.3	70.2

methods for the fixed cylinder is striking. Firstly, for the ADR-F, the blue area that depicts the highest error is only limited to the immediate vicinity of the immersed boundary interface and does not extend downstream of the cylinder. Conversely, the highest error for the NR-F spreads further downstream of the cylinder. The error levels upstream of the cylinder are much lower for the ADR-F than for the NR-F, with a significantly reduced orange area. Downstream of the cylinder, the error levels are more or less the same, except for locations where the vorticity levels are important (it corresponds to the blue region for the NR-F). The discrepancies between the ADR-M and NR-M are less pronounced compared to the fixed cases but most of the aforementioned differences persist. Overall, the maximum error levels are observed in the vicinity of the cylinder and where the vorticity levels are important.

5.4. Influence of the ADR-IBM parameters

A small study on the effect of the number of input points used for the cubic spline interpolation is performed on the fixed cylinder for Cases1-3. The velocity and pressure L^2-Norm error decay for each mesh resolution with the number of input interpolation points varying between $1 \le N_{FP} \le 3$ and with and without skipping are shown in Fig. 11. At the lowest resolution (Case1), the cases with skipping perform marginally better than the ones with no-skipping. However, for Case2 and Case3, the skipping cases provide lower levels of error for both velocity and pressure. This result was expected since at marginal resolution the probability of problematic points (i.e. points extremely close to the boundary interface) is low and hence does not negatively impact the error. Contrarily, as the resolution is increased, more problematic points appear which negatively (even though marginally) impact the decay of the L^2-Norms. The number of input points for the cubic spline reconstructions does not play a significant role as all cases appear to coincide. However, it should be mentioned that $N_{FP} = 1$ provides the greatest error both with and without skipping while $N_{FP} = 2$ and $N_{FP} = 3$ provide almost identical levels of error. Hence, for all studies performed with the ADR-IBM in this study, skipping is always enabled and the number of input fluid points is set to $N_{FP} = 2$.

The influence of the moving velocity $\boldsymbol{u}_m$ on the velocity and pressure L^2-Norms is investigated in Table 1 for Case3 with the streamwise component of u_m varying from 0 to -1. The corresponding free-stream velocity is adjusted accordingly. It is anticipated that the error should increase when $\boldsymbol{u}_m$ is increased as the spurious oscillations on the pressure field would become more severe. As shown in Table 1, both the velocity and pressure errors increase when the streamwise component of $\boldsymbol{u}_m$ is increased. Compared to the fixed case, the pressure shows the greatest increase in error (up to 70%) which is more than double the one observed for the velocity (up to 31%). It should be noted that the convergence rate of the velocity and pressure errors is not affected when the resolution is increased.

Table 2
Comparison of the averaged drag coefficients and Strouhal numbers for the flow around a cylinder at $Re =$ 300 obtained with sixth- and second-order schemes and ε calculated on various mesh resolutions and with an analytical function.

Study	$\bar{C}_D$	St
Ye et al. [34]	1.38	0.21
Wieselsberger [35]	1.22	-
Mittal and Balachandar [36]	1.37	0.21
Luo et al. [9]	1.43	0.196
Mittal et al. [37]	1.36	0.210
Henderson [38]	1.37	-
Williamson [39]	-	0.203
Williamson [40]	1.28	0.20
Marella et al. [41]	1.28	0.22
Le et al. [42]	-	0.208
Le et al. [43]	-	0.200
Present ADR-Fixed, 6th$\mathcal{O}$, $\varepsilon\|_{n_x,n_y}$	1.50	0.21
Present ADR-Fixed, 6th$\mathcal{O}$, $\varepsilon\|_{5n_x,5n_y}$	1.41	0.21
Present ADR-Fixed, 6th$\mathcal{O}$, $\varepsilon\|_{10n_x,10n_y}$	1.39	0.21
Present ADR-Fixed, 6th$\mathcal{O}$, $\varepsilon\|_{20n_x,20n_y}$	1.41	0.21
Present ADR-Fixed, 6th$\mathcal{O}$, $\varepsilon\|_{Analytical}$	1.40	0.21
Present ADR-Fixed, 2nd$\mathcal{O}$, $\varepsilon\|_{Analytical}$	1.46	0.22
Present ADR-Moving, 6th$\mathcal{O}$, $\varepsilon\|_{Analytical}$	1.38	0.21

6. Flow around a cylinder at $Re = 300$

To demonstrate the robustness of the moving ADR-IBM in accurately capturing drag/lift coefficients and their amplitude, the unsteady flow around a 2D cylinder at $Re = 300$ is considered (the flow around a cylinder at $Re = 40$ is steady with constant drag/lift). Two simulations with the ADR-IBM are performed, one with a fixed and one with a moving cylinder. A square computational domain with length and height $L = H = 25D$ (where D is the cylinders diameter) is considered for both simulations. A resolution of $n_x \times n_y = 385 \times 384$ is used, with a stretched mesh in the vertical direction towards the centre of the domain with the smallest mesh spacing being $\Delta y_{\min} = 0.04$. Initial conditions are the same as in the previous section except for the boundary conditions. A uniform velocity $u_\infty = 1.0$ is imposed at the inlet while a 1D-convection equation is imposed at the outlet. Periodic boundary conditions are imposed in the vertical direction. For the fixed case, the cylinder is placed at the centre of the domain at $(x_0, y_0) = (12.5, 12.5)$. The simulation is stopped after 120 non-dimensional time units. For the moving case, the cylinder is placed at (x_0, y_0) and the flow is initialised with the flow-field obtained from a fully converged simulation with the fixed cylinder at the same location. The inflow free-stream velocity is scaled at $u_\infty = 0.5$ and the moving velocity is set at $\boldsymbol{u}_m = (-0.5, 0, 0)$. The simulation is stopped after the cylinder has moved 8D. A time-step of $\Delta t = 3.2 \times 10^{-4} D/u_\infty$ has been selected for both simulations.

Table 2 summarises the averaged drag coefficients $\bar{C}_D$ and the Strouhal numbers ($St = f_s D/u_\infty$, where f_s denotes the vortex shedding frequency) found in the literature along with the present results obtained with the ADR-IBM with sixth- and second-order schemes. The results are also presented for the ADR-F for various ε scalar fields (see Eq. (10)) defined on the same number of mesh nodes ($n_x \times n_y$) as the ones used for the simulation or on more refined resolutions ($\times 5, \times 10, \times 20$). Additionally, these results are compared to the case where ε has been obtained via the equation of a circle (exact analytical location of the cylinder). As it can be seen, for the sixth-order schemes, the drag coefficients and Strouhal numbers are in excellent agreement with the values reported in literature, except when ε is defined on the same resolution as the one used for the simulation. For this set-up, the size of the cylinder is incorrect, resulting in an erroneous averaged drag coefficient and Strouhal number. It demonstrates that, when a complex geometry (which cannot be described by any analytical equations) is simulated, then ε should be approximated on a finer mesh than the mesh of the simulation. The mean drag coefficient for the ADR-M is very similar to the one obtained for the ADR-F case (relative error of less than 1.5%), with values in line with published data. The advantage of using high-order schemes (even if the formal order of the solution is reduced) is clearly shown with this example, with the second-order schemes producing over-estimated mean drag coefficients, even when the exact ε is used. In order to obtain the correct mean drag coefficient with second-order schemes, the simulation would need to be performed with a higher spatial resolution.

Fig. 12 shows the time evolution of the drag and lift coefficients for the ADR-F and ADR-M. It should be noted that the drag and lift coefficients of the ADR-M, have been filtered with a Savitzky-Golay filter to deal with pressure jumps observed each time a moving boundary moves into a new mesh node (see Fig. 13). The small underestimation of the mean drag reported in Table 2 can be observed as a constant shift, although the amplitude for the drag coefficients for both simulations is the same. The lift coefficient of the ADR-M is found to be in excellent agreement with the ADR-F's, as suggested by the identical Strouhal number obtained by the two simulations.

As it can be seen in Fig. 13, a small pressure jump can be observed upstream of the cylinder when the moving boundary moves into a new mesh node. This pressure jump does not last for a long time, in the present case, as it is no longer visible

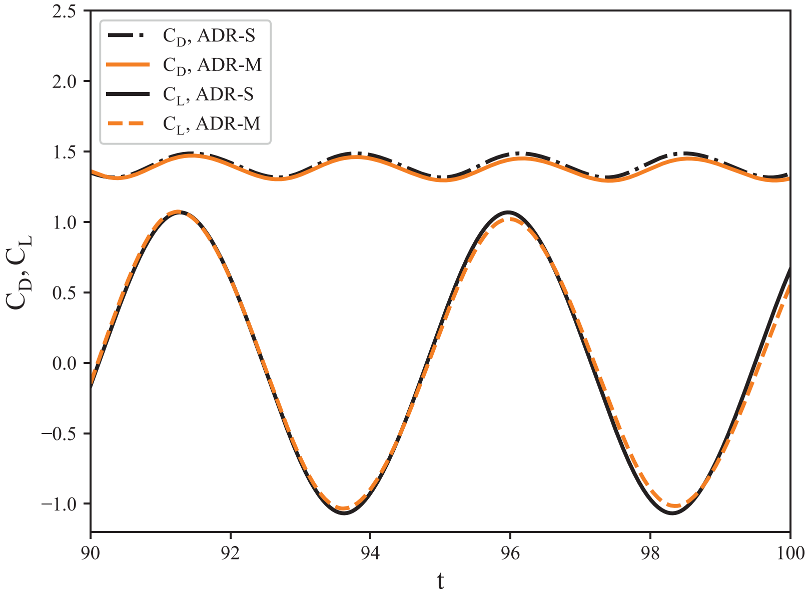

Fig. 12. Evolution of the drag and lift coefficients for the flow around a cylinder at $Re = 300$.

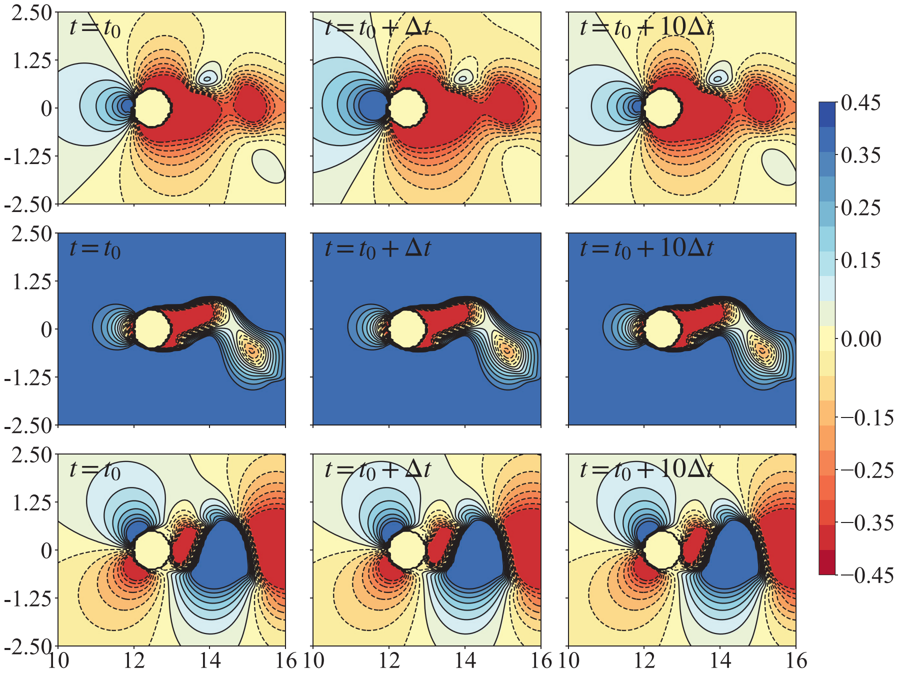

Fig. 13. Instantaneous visualisations of the pressure jump generated for a moving cylinder at $Re = 300$. Top plots: pressure field; middle plots: streamwise velocity component and bottom plots: vertical velocity component.

after 10 time-steps. An important observation here is that this pressure jump does not affect the velocity field as it can be seen for the middle and bottom plots of Fig. 13.

7. Flow around a sphere at $Re = 300$

So far, the ADR-IBM has been tested for 2D flows. To further validate the ADR-IBM in 3D, the flows around a fixed and a moving sphere at $Re = 300$ are considered in this section. The flow at this Reynolds number becomes unsteady with time-periodic and planar-symmetric vortex shedding [44,45]. A rectangular domain of dimensions $L \times H \times W = 40D \times 6D \times 6D$ (where D is the diameter of the sphere) with a resolution of $n_x \times n_y \times n_z = 1025 \times 256 \times 256$ mesh nodes is selected for

Table 3
Comparison of the averaged drag coefficients and Strouhal numbers for the flow around a sphere at $Re = 300$. The obelisk symbol (†) is used to indicate experimental results.

Study	$\bar{C}_D$	St
Le Clair et al. [46]	0.632	-
Roos and Willmarth [47]†	0.629	-
Johnson and Patel [44]	0.656	0.137
Kim et al. [11]	0.657	0.134
Ploumhans et al. [45]	0.683	0.135
Constantinescu and Squires [48]	0.655	0.136
Constantinescu and Squires [49]	0.655	-
Campregher et al. [50]	0.675	-
Mimeau et al. [51]	0.673	0.133
Present ADR-F	0.644	0.136
Present ADR-M	0.633	0.136

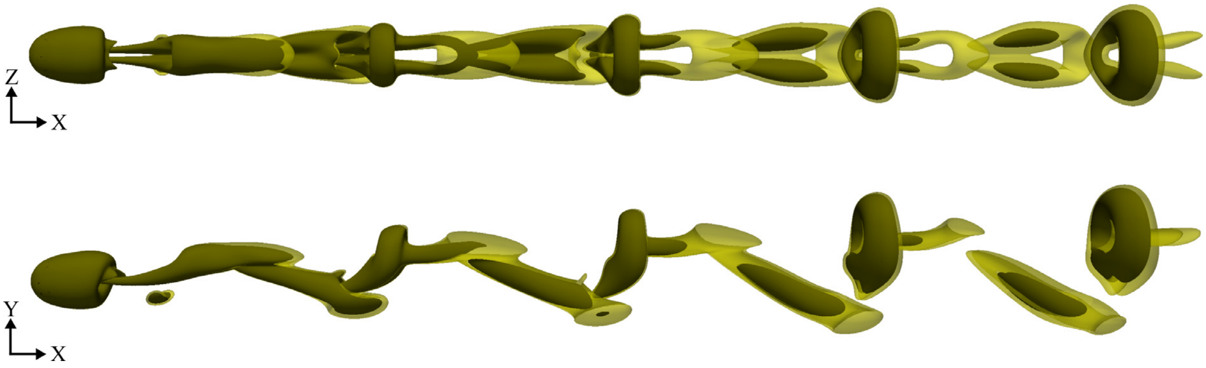

Fig. 14. Visualisation of vortical structures of the wake of a moving sphere at $Re = 300$ in the x-z (top) and x-y (bottom) planes with the Q-criterion (with two levels of isocontours at 0.5 (yellow) and 2.0 (green colours)). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

all simulations. Periodic boundary conditions are applied at the boundaries in the y and z directions. A uniform velocity profile of magnitude $u_\infty = 1.0$ is imposed at the inlet while a 1D-convection equation is imposed at the outlet. The velocity flow field is initialised with random noise which follows a Gaussian distribution with its peak located at the centre of the domain in the y and z directions (with maximum intensity of 0.1% of u_∞). For the fixed case, the sphere is positioned at the centre of the domain at $(x_0, y_0, z_0) = (20D, 3D, 3D)$ and the simulation is performed for 400 non-dimensional time units. The moving sphere is located at the centre of the domain at $t = 0$, and the flow is initialised by the solution provided from a fully converged simulation with a fixed sphere. The inflow free-stream velocity is scaled at $u_\infty = 0.5$ and the moving velocity at $\boldsymbol{u}_m = (-0.5, 0, 0)$. The simulation with the moving sphere is stopped when the sphere has travelled a distance of 10D. A time-step of $\Delta t = 2.0 \times 10^{-3} D/u_\infty$ is selected for both simulations.

Table 3 summarises the averaged drag coefficients and Strouhal numbers obtained with the ADR-F and ADR-M along with experimental and numerical values found in literature. Both the fixed and moving cases are in excellent agreement with published data. As expected, the drag coefficient of the ADR-M is slightly under-predicted compared to the ADR-F, possibly due to the pressure jumps when a moving boundary moves into a new mesh node, but their difference remains small, under 2%. Note that the Strouhal numbers, however, are identical for both ADR-F and ADR-M.

The Q-criterion can be used to visualise the vortical structures in the wake of the moving sphere in the x-z (top view) and x-y (side view) planes as shown in Fig. 14. The two views reveal the three-dimensionality of the flow along with the hairpin vortices when the sphere has traveled $10D$ at $T = 420$, in agreement with published data [45,51].

8. Performance analysis: flow around a sphere at $Re = 3700$

To demonstrate the ability of the proposed method to efficiently perform large 3D turbulent simulations on many computational cores in a DNS setting, the flow around a sphere at $Re = 3700$ is considered with two simulations, one with a fixed and one with a moving sphere. It should be stressed, that in order to obtain statistically significant quantities and perform any meaningful comparison of mean velocity profiles, first- and second-order statistics, etc., the simulation should be run for a significantly longer time. However, the true purpose of this test case is to demonstrate that the ADR-IBM can be used for large simulations on HPC facilities and not to thoroughly analyse the physics of this turbulent flow. Hence, only the mean drag coefficient and the Q-criterion will be used as an indicative comparison to validate the flow physics.

Table 4
Comparison of the averaged drag coefficients for the flow past a sphere at $Re = 3700$.

Study	$\bar{C}_D$
Rodriguez et al. [52] (DNS)	0.394
Yun et al. [53] (LES)	0.355
Schlichting [54] (Exp.)	0.39
Present ADR-F/ADR-M (DNS)	0.398

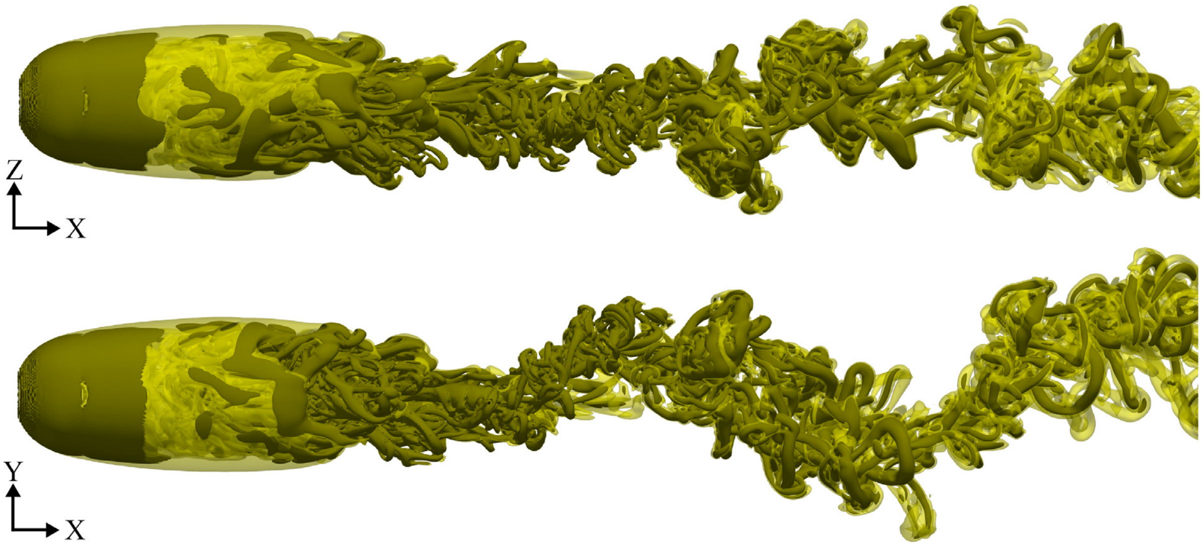

Fig. 15. Visualisation of vortical structures of the wake of a moving sphere at $Re = 3700$ in the x-z (top) and x-y (bottom) planes with the Q-criterion at $t = 64.0$ (with two levels of isocontours at 0.05 (yellow) and 1.5 (green colours)). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

A rectangular domain of dimensions $L \times H \times W = 15D \times 6D \times 6D$ (where D is the diameter of the sphere) with a resolution of $n_x \times n_y \times n_z = 1537 \times 1024 \times 1024$ mesh nodes is selected. Periodic boundary conditions are applied at the boundaries in the y and z directions, while a uniform velocity profile of magnitude $u_\infty = 1.0$ is imposed at the inlet. A 1D-convection equation is imposed at the outlet. The velocity flow field is initialised with random noise which follows a Gaussian distribution with its peak located at the centre of the domain in the y and z directions (maximum intensity of 0.1% of u_∞). For the fixed case, the sphere is located at $(x_0, y_0, z_0) = (5D, 3D, 3D)$ and the simulation is terminated after 63.5 non-dimensional time units. The moving sphere is also placed at $(x_0, y_0, z_0) = (5D, 3D, 3D)$ at $t = 63.5D/u_\infty$, and the flow is initialised with the flow-field obtained from the fixed simulation. The inflow free-stream velocity is scaled at $u_\infty = 0.5$ and the moving velocity is set at $\boldsymbol{u}_m = (-0.5, 0, 0)$. The moving simulation is terminated at $t = 64.0$ after the sphere has travelled a distance of $0.25D$. A time-step of $\Delta t = 1.0 \times 10^{-4} D/u_\infty$ has been selected for both simulations. Both simulations were performed on the Tier-0 supercomputer Irene-Rome on 32,768 computational cores.

Table 4 compares the averaged drag coefficients obtained with the ADR-F/ADR-M with values reported in literature. The value obtained with the proposed immersed boundary method, appears to be in excellent agreement with the DNS and experimental results reported. Additionally, the vortical structures in the wake of the moving sphere (once the sphere has traveled $0.25D$ at $t = 64.0$) in the x-z (top view) and x-y (side view) planes are visualised with the Q-criterion in Fig. 15. A small recirculation region can be identified in the wake which follows a helical pattern as it is convected downstream and results in the big undulations as seen in Fig. 15 in the x-y plane.

The core strengths of the proposed Alternating Direction Reconstruction IBM (ADR-IBM) lie in its simplicity and compatibility with the 2D domain decomposition strategy employed in Xcompact3D. The excellent scalability and performance of the solver without any immersed boundary method has been exhibited across various HPC platforms for up to hundreds of thousands computational cores initially by Laizet and Li [18] and more recently by Bartholomew et al. [17]. To demonstrate the performance of Xcompact3D with the ADR-IBM, two Tier-1/0 supercomputers were used: ARCHER (UK) and Irene-Rome (France). The ARCHER facility is based on the Cray XC30 supercomputer which consists of 4920 nodes, each equipped with two 2.7 GHz, 12-core Intel Xeon E5-2697 v2 (Ivy Bridge) processors which amounts to a total number of 109,056 computational cores. Each node has 64GB of DDR3 memory and a memory bandwidth of 117GB/s (4.9GB/s per core). The Aries Interconnect system has a peak bisection bandwidth of over 19,013 GB/s over the whole system. Irene-Rome, consists of 2292 nodes equipped with two 1.6GHz, 64-core AMD Rome (AVX2) processors for a total 193,376 computational cores (with

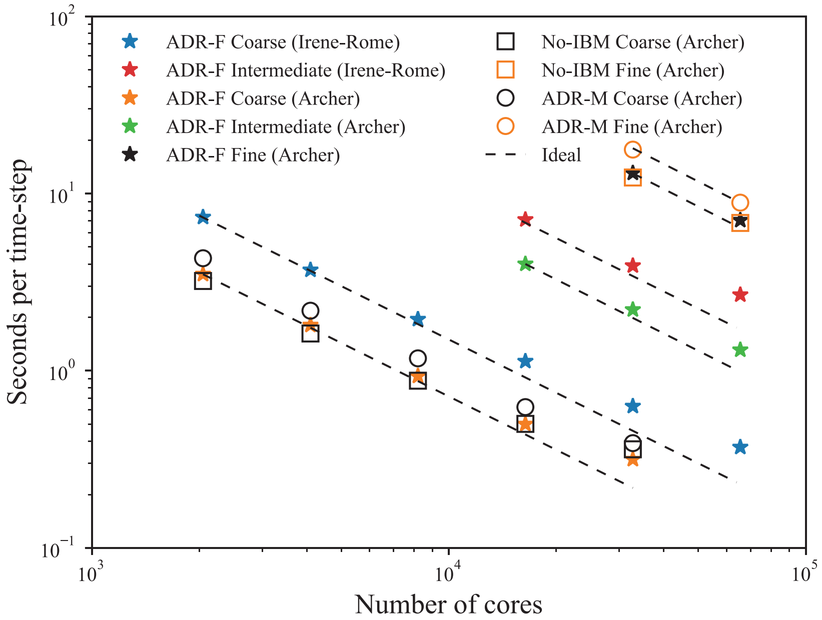

Fig. 16. Strong scaling of `Xcompact3D` with ADR-F, ADR-M and No-IBM for various resolutions on the HPC facilities of Archer and Irene-Rome.

Table 5
Parallel efficiency of `Xcompact3D` for the Coarse, Intermediate and Fine simulations with the ADR-F, ADR-M and No-IBM, for a range of computational cores between 2048 to 65,536. The simulations are performed on ARCHER and Irene-Rome.

Cores	2048	4096	8192	16384	32768	65536
Coarse (No-IBM), Archer	-	0.99	0.91	0.80	0.63	-
Coarse (ADR-F), Archer	-	0.97	0.94	0.88	0.69	-
Coarse (ADR-M), Archer	-	0.99	0.92	0.87	0.69	-
Coarse (ADR-F), Irene-Rome	-	0.99	0.94	0.81	0.73	0.62
Intermediate (ADR-F), Archer	-	-	-	-	0.91	0.76
Intermediate (ADR-F), Irene	-	-	-	-	0.91	0.67
Fine (No-IBM), Archer	-	-	-	-	-	0.91
Fine (ADR-F), Archer	-	-	-	-	-	0.93
Fine (ADR-M), Archer	-	-	-	-	-	1.00

a maximum floating point performance of 41.6Gflops/s per core). Each node has 156GB of memory available (2GB/core) and a memory bandwidth of 320GB/s (2.5GB/s per core). The network performs at 100GB/s shared between 128 cores.

To investigate the performance of the ADR-IBM, various resolutions were considered in the range between 1.6 and 68.7 billion mesh nodes with up to 65,536 computational cores for the flow around a sphere at $Re = 3700$. The focus here is on the strong scaling, which is defined as the variation of the wall clock time with the number of computational cores for a fixed sized problem. Fig. 16 depicts the scaling of `Xcompact3D` with the ADR-IBM for three different mesh resolutions (Coarse: 1537 × 1024 × 1024, Intermediate: 3073 × 2048 × 2048 and Fine: 4097 × 4096 × 4096) on Irene-Rome and ARCHER.

Overall, the No-IBM (which corresponds to a steady flow with no sphere), ADR-F and ADR-M exhibit an excellent strong scaling on both platforms for up to 65,536 computational cores. The only times when the scaling diverges from the ideal one occur when the mesh resolution is too coarse compared to the very large number of cores which results in too few mesh nodes per core and too many global transpositions. This reduces the scaling efficiency as it is observed in the Coarse and Intermediate resolutions for very large number of computational cores. Further, it can be observed that the cost of the ADR-F is only marginally higher than the cost of the solver without the IBM (No-IBM) for all mesh resolutions and computational cores considered. The moving ADR-IBM shows a small cost increase (as the position of the immersed object needs to be evaluated at each time-step) while it maintains the excellent strong scaling of `Xcompact3D`. Note that the simulations on ARCHER are faster than on Irene-Rome because of the higher processor frequency.

Table 5 shows the parallel efficiency of the No-IBM, ADR-F and ADR-M for the data reported in Fig. 16. A reduction in the efficiency can be observed for the Coarse simulations with the increase in the number of computational cores due to the very small number of mesh nodes per core, resulting in a large number of global communications of small size. However, the efficiency recovers and approaches the ideal efficiency for the Intermediate and Fine cases for up to 65,536 computational

Table 6
Percentage cost of communication and the main steps of `Xcompact3D` in a time loop, for the Coarse and Fine simulations with the ADR-F, ADR-M and No-IBM, for a range of computational cores between 2048 to 32,768 performed on ARCHER.

Cores	Com	Conv.-Diff.	Poisson	GradP	Div	ε	Spline
2048 Coarse (No-IBM)	53.93	53.11	20.85	11.35	10.77	-	-
2048 Coarse (ADR-F)	54.01	54.33	20.23	11.07	10.47	1.24	0.13
2048 Coarse (ADR-M)	47.58	41.70	15.45	8.48	8.22	24.14	0.096
4096 Coarse (No-IBM)	56.14	53.81	20.79	11.46	10.81	-	-
4096 Coarse (ADR-F)	56.12	53.62	20.32	11.34	10.76	1.25	0.12
4096 Coarse (ADR-M)	50.96	41.95	15.74	8.44	8.14	23.68	0.092
8192 Coarse (No-IBM)	66.43	53.50	22.80	11.18	9.60	-	-
8192 Coarse (ADR-F)	68.25	53.22	22.10	11.06	10.01	1.15	0.10
8192 Coarse (ADR-M)	60.43	41.20	17.10	8.38	7.69	23.63	0.084
16,384 Coarse (No-IBM)	79.11	51.13	25.42	11.21	9.68	-	-
16,384 Coarse (ADR-F)	79.00	51.60	25.15	10.70	9.39	0.77	0.087
16,384 Coarse (ADR-M)	71.59	43.81	20.42	8.34	8.03	17.34	0.076
32,768 Fine (No-IBM)	65.21	53.05	22.45	11.12	10.86	-	-
32,768 Fine (ADR-F)	66.63	53.81	21.62	10.52	10.60	1.32	0.083
32,768 Fine (ADR-M)	56.20	39.29	15.86	7.80	7.91	27.54	0.061

cores. Additionally, the parallel efficiencies obtained with the ADR-F and ADF-M (which are very similar) are equal or better compared to the ones obtained with the No-IBM for all resolutions and number of computational cores considered. Hence, it can be concluded that, because the ADR-IBM introduces a small computational overhead, it improves the parallel efficiency of `Xcompact3D` thanks to a better ratio of calculations/communications.

To further assess the performance and cost of the proposed ADR-IBM, all aforementioned simulations performed on ARCHER generated a CrayPat report for the profiling of the solver [55]. Table 6 provides a detailed breakdown of the percentage of time spent on communications and on the core parts of `Xcompact3D` (which includes the calculations of the convective and diffusive terms, the Poisson equation, the gradient of the pressure and the divergence) with a particular focus on the immersed boundary method-related costs (including the generation of the ε scalar field, and the cubic spline reconstructions). For the simulations presented in this table, the scalar field ε has been computed on a highly refined mesh (10 times more refined than the mesh of the simulation). As explained already, if ε was calculated on the mesh used for the simulation, the location of the immersed boundary interface of the sphere would not be known accurately, hence the suggestion to obtain ε on a refined mesh for a more accurate enforcement of the boundary conditions via the spline reconstructions. The communication category (Com) includes mainly the `MPI_ALLTOALL(V)` calls while MPI initialisation calls such as `MPI_CART_CREATE` and `MPI_CART_SUB` have been excluded. In more detail, a significant amount of the time is spent on the MPI global communications irrespective of whether any of the IBMs are used or not. As explained in Laizet and Li [18], up to 70 global transpositions are performed per time step. As a result, the communications can represent up to 79% of the total cost to advance the simulations from one time step to another, without, however, impacting the parallel efficiency of the code. The other reason to explain why the communications part is so large is related to the efficiency of the calculation part which is mainly based on highly-vectorised 3D loops. Hence, a small amount of time is spent on calculation and more time is spent on communication, without affecting the scalability of the solver [18].

For all cases considered, the most expensive part of the solver involves the computation of the convective and diffusive terms as it requires many global communications. For the No-IBM and ADR-F, the next most expensive parts are the Poisson's solver followed by the pressure gradient and divergence calculations. For ADR-F, a minuscule percentage ($\leq 1.32\%$) is spent on the demarcation of the immersed boundary interface which is approximated on a very refined mesh (in order to obtain an accurate ε). Furthermore, the cost of all the cubic spline reconstructions remains below 0.13%. Contrarily, the cost of the interface demarcation for the ADR-M is considerably higher as ε has to be redefined at each time-step. Its cost can reach $\approx 27.50\%$ which can even exceed the cost of the Poisson solver for certain cases, but without affecting the scalability. As expected, for the moving ADR-IBM, the cubic spline reconstructions account for a negligible cost of less than 0.1%. Finally, it should be noted that the sum of the percentages for the main steps to advance the simulations from one time-step to another (Conv-Diff, Poisson, DradP, Div, ε and Spline) is not exactly equal to 100 but is very close to 100. The missing percentages are related to the time advancement calculation, to the imposition of the boundary conditions and to the pressure correction step. For clarity, these small steps are not reported in the table as they only account for few percentages. In summary, the highest cost related to the proposed IBMs involves the approximation of the immersed boundary interface which for simplified geometries can be avoided entirely if analytical equations are used. Furthermore, the cubic spline interpolations introduce a negligible computational overhead owing to the simplicity of the 1D reconstructions. Hence, the proposed ADR-IBM introduces a very small computational overhead and is shown to be highly compatible with the 2D domain decomposition strategy employed by `Xcompact3D` ensuring the excellent scalability of the solver with a large number of cores.

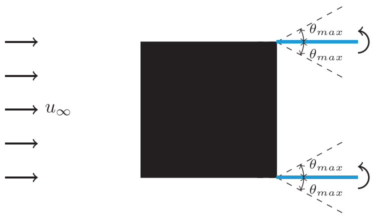

Fig. 17. Schematic demonstrating the in-phase snaking forcing strategy with rear pitching flaps on a square bluff body.

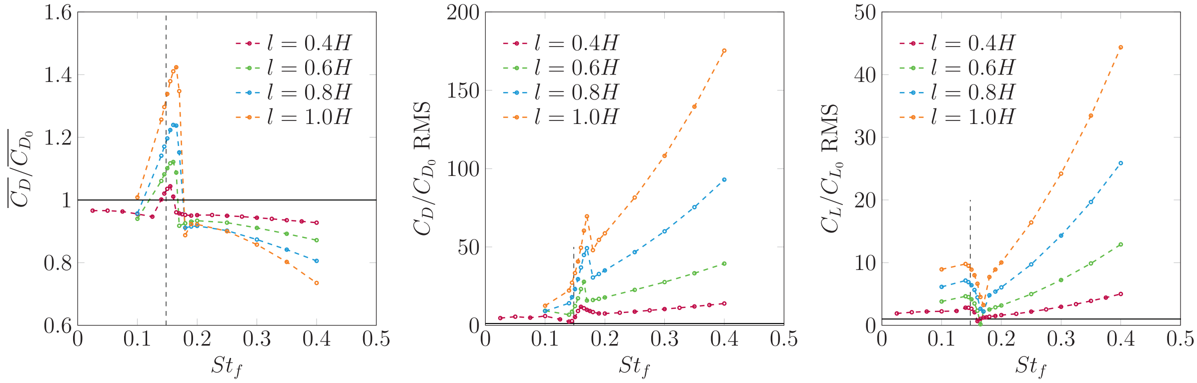

Fig. 18. Response of the normalised mean and RMS of the hydrodynamic coefficients for a wide range of forcing Strouhal numbers. The vertical dashed lines correspond to the unforced natural frequency of the flow while the solid horizontal ones to unity ratios.

9. Forced laminar bluff body wake

The flow over bluff-bodies remains a relevant and flourishing topic due to the plethora of engineering applications [56]. The modification and control of such oscillatory flows through harmonic excitation remains challenging due to the complex and nonlinear wake-forcing interactions. Recently researchers have experimentally studied the effect of harmonic forcing on turbulent bluff body wakes [57–60]. Many interesting nonlinear phenomena such as harmonic and subharmonic resonances which result in significant drag increase, were attributed to the interaction between the forcing and the laminar vortex shedding mode that persists even at high Reynolds numbers. Hence, modelling and understanding the fundamental mechanisms that occur when a laminar bluff body wake is harmonically forced is relevant to many engineering applications [56].

In this section, the case of a harmonically forced laminar bluff body wake at a low Reynolds number $Re = 100$ (based on the height H and free-stream velocity u_∞) is studied with distance and velocity scaled by H and u_∞, respectively. The wake is generated by a square bluff body. The forcing is applied by a pair of thin, flat and pitching plates attached on the rear of the body, as seen in Fig. 17. The flap motion can be described by the flap angle $\theta = \theta_{\max}\sin(2\pi f t)$ where $\theta_{\max}$ is the maximum amplitude of oscillation (in degrees), f the driving frequency and t the time. The flaps move in-phase and with the same direction generating a snaking motion (see Fig. 17). A flap thickness of 5% of the body height is selected and the following flap lengths $l/H = 0.4, 0.6, 0.8, 1.0$ are investigated. The maximum flap angle is set at $\theta_{\max} = 20^\circ$ and the forced Strouhal number $St_f = \dfrac{fH}{u_\infty}$ is a dimensionless frequency. A domain size of $L_x \times L_y = 40H \times 20H$ discretised with $n_x \times n_y = 1025 \times 768$ mesh nodes is used in this section after a preliminary mesh convergence study. A non-uniform mesh is employed in the vertical direction which is refined towards the center of the domain, with the smallest mesh spacing being $\Delta y|_{\min} = 0.017H$. A very small time-step $\Delta t = 5.0 \times 10^{-4} H/u_\infty$ is used to avoid time-integration errors. The free-stream velocity u_∞ is applied at the inlet to the computational domain (making the dimensionless inlet velocity equal to 1), while a 1D convection equation is used at the outlet defined as:

$$\boldsymbol{u}|_{n_x} = \boldsymbol{u}|_{n_x} - \frac{u_\infty \Delta t}{\Delta x}(\boldsymbol{u}|_{n_x} - \boldsymbol{u}|_{n_x-1}), \quad (16)$$

where $\boldsymbol{u}|_{n_x}$ is the outlet velocity and $\boldsymbol{u}|_{n_x-1}$ the velocity one mesh node before the outlet. Periodic boundary conditions are used in the vertical direction. The flow is initialised with random noise which follows a Gaussian distribution centered towards the middle of the domain in the vertical direction with a maximum intensity of 10% of u_∞. The center of the square bluff body is located at $(x_0, y_0) = (10, 10)$.

The frequency response of the forced wake to a wide range of forcing Strouhal numbers is presented in this section. Fig. 18 shows the evolution of the normalised mean drag and root mean square of the drag and lift coefficients for 0.025 ≤

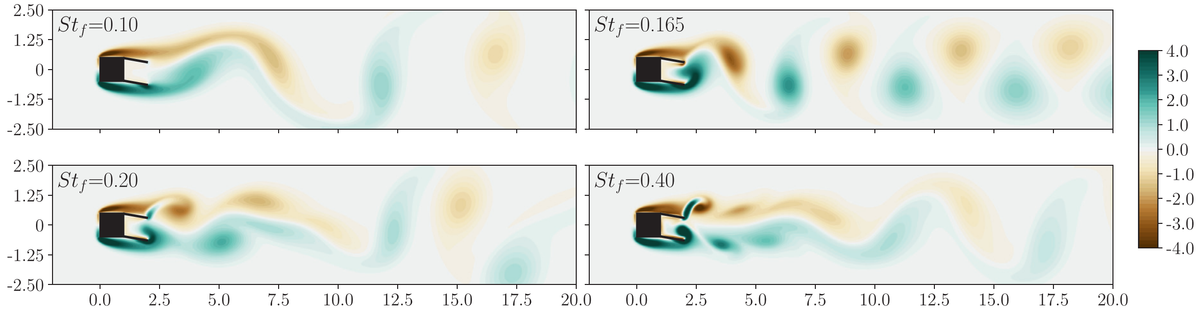

Fig. 19. Instantaneous visualisations of the vorticity field for the forced flow over a square cylinder at $Re = 100$ for various forcing Strouhal numbers with the snaking motion and a flap length of $l_f = 1.0$ at $t = 5T/4$.

$St_f \leq 0.40$. The normalisation of the coefficients is performed with the values of the unforced natural flow over the bluff body without the flaps at the same Reynolds number. The coefficients obtained for the natural flow are as follows: mean drag ($C_{D_0} = 1.489$), drag RMS $C_{D_0} = 1.489$, lift RMS $C_{L_0} = 4.9 \times 10^{-3}$ and Strouhal number $St_0 = 0.148$. These values are in excellent agreement with results reported by Sohankar et al. [61], Sharma and Eswaran [62] and Sahu et al. [63]. As it can be seen, the mean and RMS of the drag coefficient show a clear and distinct peak for all flap lengths near the natural (unforced) Strouhal number. Contrarily, a trough appears on the lift RMS in the same region. The peaks produce a significant mean drag increase of more than 40%. This increase is the result of a resonant interaction between the forcing and the primary vortex shedding instability of the flow. The resonance amplifies the vortex shedding mode as observed in Fig. 19 for $St_f = 0.165$, with a significant increase in the vorticity magnitude. As the forcing frequency is extended past the nonlinear resonant region, a drag reduction can be observed which is the result of a wake reorganisation (see Fig. 19 for $St_f = 0.20$) and a partial wake symmetrisation (see Fig. 19 for $St_f = 0.40$). Overall, larger flap lengths result in the greatest drag reduction. The response of the root mean squares is quite similar for all lengths which show a rapid increase after the nonlinear resonant region.

Finally, it has been demonstrated that the ADR-IBM can handle multiple moving geometries and can capture complex nonlinear resonant phenomena that occur through the interaction of the forcing (applied by the two pitching flaps) and the primary vortex shedding mode.

10. Conclusion and outlook

A novel, simple and scalable Alternating Direction Reconstruction Immersed Boundary Method capable of performing simulations with fixed and moving objects in a viscous incompressible fluid on supercomputers has been presented. The proposed method relies on 1D cubic spline interpolations to reconstruct an artificial flow field inside the immersed object while imposing the appropriate boundary condition on the boundaries of the object. The implementation, assessment and validation of the IBM have been performed using the high-order finite-difference framework Xcompact3D.

Through a unique and thorough validation against a spectral reference solution [32] for fixed and moving cylinders at $Re = 40$, it has been demonstrated that the proposed ADR-IBM is superior to the traditional No-Reconstruction IBM. The proposed method consistently produces significantly lower velocity and pressure error levels for a wide range of spatial resolutions. Further, detailed error maps depicting the spatial distribution of the velocity L^2-Norms illustrate that the error of the proposed method is only limited to the vicinity of the boundaries of the immersed object. Conversely, for the No-Reconstruction IBM, the aforementioned errors affect a greater region due to the inability of the NR-IBM to accurately define the immersed boundary interface.

Additionally, further validations of the flow over a cylinder and a sphere at $Re = 300$ have demonstrated the robustness of the proposed method with accurate predictions of the drag/lift coefficients, even at marginal resolutions. Moreover, the capability of the proposed method to perform turbulent simulations with moving geometries on supercomputers is demonstrated via the test cases of flow around fixed and moving spheres at $Re = 3700$, with multi-billion mesh node simulations.

The excellent scalability and parallel efficiency of Xcompact3D with the ADR-IBM has been evidenced for up to 65,536 computational cores on Tier-1/0 supercomputers for the flow over fixed and moving spheres at $Re = 3700$ with more than 68 billion mesh nodes. The impressive performance of the ADR-IBM sets it apart as very few IBMs have been shown to have similar robustness and scalability when performing DNS with moving objects on massively parallel platforms. It should also be noted that the proposed ADR-IBM will be made open-source via the Github of Xcompact3D so it can be implemented to other flow solvers.

The potential of the proposed method in handling multiple and thin moving boundaries for practical applications has also been demonstrated with the test case of the forced bluff body wake with two rear pitching flaps. The method has been able to properly capture complex nonlinear resonant phenomena which occur when the wake is forced near the frequency of the natural (unforced) flow (i.e. the flow over the body without the flaps) and result in an amplification of the primary vortex shedding mode. It has been shown that rear pitching flaps have the potential to be used as an active drag reduction technique for bluff body wakes.

The next step is to combine the ADR-IBM with various explicit and implicit Large-Eddy-Simulation (LES) models in order to perform turbulent simulations with multiple, complex and moving geometries at high Reynolds numbers. Moreover, the ADR-IBM will be extended to allow the imposition of Neumann boundary conditions at the boundaries of the immersed objects, in order to perform simulations with heat-transfer features.

Acknowledgments

The authors thank EPSRC for the computational time made available on the UK supercomputing facility ARCHER via the UK Turbulence Consortium (EP/R029326/1). Furthermore, computational time was provided by PRACE via the Irene-Rome supercomputer facilities (grant number: 2019215138). Finally, the authors would like to acknowledge the Department of Aeronautics, Imperial College London for supporting this work funded by a PhD studentship.

Appendix A. Cubic spline interpolation

Here, the mathematical background of the cubic spline interpolation is presented. To start with, a cubic polynomial has the form:

$$p(x) = a + bx + cx^2 + dx^3, \tag{17}$$

where a, b, c and d are the polynomial coefficients and x is the dependent variable. Let us assume a generic function $g(x)$ defined on a closed set $[\alpha, \beta]$ where:

$$\alpha = x_1 < x_2 < \cdots < x_{n-1} < x_n = \beta, \tag{18}$$

and a fitting piece-wise cubic polynomial $S_i(x)$ defined on $[x_i, x_{i+1}]$ with $i = 1, 2, \ldots, n-2, n-1$ that has the form of Eq. (19).

$$S(x) = \begin{cases} a_1 + b_1(x-x_1) + c_1(x-x_1)^2 + d_1(x-x_1)^3, & x_1 \le x \le x_2 \\ a_2 + b_2(x-x_2) + c_2(x-x_2)^2 + d_2(x-x_2)^3, & x_2 \le x \le x_3 \\ \vdots & \vdots \\ a_i + b_i(x-x_i) + c_i(x-x_i)^2 + d_i(x-x_i)^3, & x_i \le x \le x_{i+1} \\ \vdots & \vdots \\ a_{n-1} + b_{n-1}(x-x_{n-1}) + c_{n-1}(x-x_{n-1})^2 + d_{n-1}(x-x_{n-1})^3, & x_{n-1} \le x \le x_n \end{cases} \tag{19}$$

In order to fit the generic function $g(x)$ with the polynomial in Eq. (19), the following properties have to be satisfied:

$$S(x_i) = g(x_i) \text{ for } i = 1, 2, \ldots, n \tag{20}$$

$$S_i(x_{i+1}) = S_{i+1}(x_{i+1}) \text{ for } i = 1, 2, \ldots, n-2 \tag{21}$$

$$S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}) \text{ for } i = 1, 2, \ldots, n-2 \tag{22}$$

$$S_i''(x_{i+1}) = S_{i+1}''(x_{i+1}) \text{ for } i = 1, 2, \ldots, n-2 \tag{23}$$

$$S'(x_0) = g'(x_0) \text{ and } S'(x_n) = g'(x_n). \tag{24}$$

Eqs. (21)–(24) ensure the continuity of the interpolated function and its first two derivatives while Eq. (24) represents the clamped boundary conditions of the cubic spline. To perform the interpolation procedure, the coefficients of the polynomials (a, b, c and d) need to be obtained. This can be achieved by solving Eq. (25) where A, c and B take the forms:

$$Ac = B \tag{25}$$

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & \ldots & 0 \\ h_1 & 2(h_1+h_2) & h_2 & 0 & \ldots & 0 \\ 0 & h_2 & 2(h_2+h_3) & h_3 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) & hn-1 \\ 0 & 0 & 0 & 0 & \ldots & 1 \end{bmatrix} \tag{26}$$

$$c = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \frac{3}{h_3}(a_4 - a_3) - \frac{3}{h_2}(a_3 - a_2) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix}. \tag{27}$$

With the definitions of $h_i = x_{i+1} - x_i$ and $a_i = g(x_i)$, the system can be solved for the spline coefficients by any algorithm designed for tri-diagonal systems. In this study, a Thomas algorithm is used, as it requires only $\mathcal{O}(n)$ operations [64].

References

[1] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (2) (1972) 252–271.

[2] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (1977) 220–252.

[3] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.

[4] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.

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

[6] M. Belliard, M. Chandesris, J. Dumas, Y. Gorsse, D. Jamet, C. Josserand, B. Mathieu, An analysis and an affordable regularization technique for the spurious force oscillations in the context of direct-forcing immersed boundary methods, Comput. Math. Appl. 71 (5) (2016) 1089–1113.

[7] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2) (2005) 448–476.

[8] X. Yang, X. Zhang, Z. Li, G.-W. He, A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations, J. Comput. Phys. 228 (20) (2009) 7821–7836.

[9] H. Luo, H. Dai, P.J.F. de Sousa, B. Yin, On the numerical oscillation of the direct-forcing immersed-boundary method for moving boundaries, Comput. Fluids 56 (2012) 61–76.

[10] C.R. Ong, H. Miura, Immersed boundary method with irrotational discrete delta vector for droplet simulations of large density ratio, J. Comput. Phys. 391 (2019) 280–302.

[11] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (1) (2001) 132–150.

[12] M. Kumar, S. Roy, A sharp interface immersed boundary method for moving geometries with mass conservation and smooth pressure variation, Comput. Fluids 137 (2016) 15–35.

[13] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, J. Comput. Phys. 215 (1) (2006) 12–40.

[14] D.M. Martins, D.M. Albuquerque, J.C. Pereira, Continuity constrained least-squares interpolation for SFO suppression in immersed boundary methods, J. Comput. Phys. 336 (2017) 608–626.

[15] D. Kim, H. Choi, Immersed boundary method for flow around an arbitrarily moving body, J. Comput. Phys. 212 (2) (2006) 662–680.

[16] S. Laizet, E. Lamballais, High-order compact schemes for incompressible flows: a simple and efficient method with quasi-spectral accuracy, J. Comput. Phys. 228 (16) (2009) 5989–6015.

[17] P. Bartholomew, G. Deskos, R.A. Frantz, F.N. Schuch, E. Lamballais, S. Laizet, Xcompact3d: an open-source framework for solving turbulence problems on a cartesian mesh, SoftwareX 12 (2020) 100550.

[18] S. Laizet, N. Li, Incompact3d: a powerful tool to tackle turbulence problems with up to O (10^5) computational cores, Int. J. Numer. Methods Fluids 67 (11) (2011) 1735–1757.

[19] A. Kravchenko, P. Moin, On the effect of numerical errors in large eddy simulations of turbulent flows, J. Comput. Phys. 131 (2) (1997) 310–322.

[20] S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1) (1992) 16–42.

[21] E. Lamballais, V. Fortuné, S. Laizet, Straightforward high-order numerical dissipation via the viscous term for direct and large eddy simulation, J. Comput. Phys. 230 (9) (2011) 3270–3275.

[22] T. Dairay, E. Lamballais, S. Laizet, J.C. Vassilicos, Numerical dissipation vs. subgrid-scale modelling for large eddy simulation, J. Comput. Phys. 337 (2017) 252–274.

[23] J. Mohd-Yusof, Combined immersed boundary/b-spline methods for simulation of flow in complex geometries, Ann. Res. Br. 317 (1997).

[24] 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 (1) (2000) 35–60.

[25] P. Parnaudeau, E. Lamballais, D. Heitz, J.H. Silvestrini, Combination of the immersed boundary method with compact schemes for DNS of flows in complex geometry, in: Direct and Large-Eddy Simulation V, Springer, 2004, pp. 581–590.

[26] P. Parnaudeau, J. Carlier, D. Heitz, E. Lamballais, Experimental and numerical studies of the flow over a circular cylinder at reynolds number 3900, Phys. Fluids 20 (8) (2008) 085101.

[27] A. Gronskis, G. Artana, A simple and efficient direct forcing immersed boundary method combined with a high order compact scheme for simulating flows with moving rigid boundaries, Comput. Fluids 124 (2016) 86–104.

[28] R. Gautier, S. Laizet, E. Lamballais, A DNS study of jet control with microjets using an immersed boundary method, Int. J. Comput. Fluid Dyn. 28 (6–10) (2014) 393–410.

[29] J.H. Seo, R. Mittal, A high-order immersed boundary method for acoustic wave scattering and low-mach number flow-induced sound in complex geometries, J. Comput. Phys. 230 (4) (2011) 1000–1019.

[30] C.A. Hall, W.W. Meyer, Optimal error bounds for cubic spline interpolation, J. Approx. Theory 16 (2) (1976) 105–122.

[31] E.N. Houstis, E. Vavalis, J.R. Rice, Convergence of o(h^4) cubic spline collocation methods for elliptic partial differential equations, SIAM J. Numer. Anal. 25 (1) (1988) 54–74.

[32] R. Gautier, D. Biau, E. Lamballais, A reference solution of the flow over a circular cylinder at Re= 40, Comput. Fluids 75 (2013) 103–111.

[33] N. Hasan, S.F. Anwer, S. Sanghi, On the outflow boundary condition for external incompressible flows: a new approach, J. Comput. Phys. 206 (2) (2005) 661–683.

[34] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (2) (1999) 209–240.

[35] C. Wieselsberger, New data on the laws of fluid resistance, Technical Report, National Advisory Committee For Aeronautics, 1922.
[36] R. Mittal, S. Balachandar, On the inclusion of three-dimensional effects in simulations of two-dimensional bluff-body wake flows, in: Proceedings of the ASME Fluids Engineering Division Summer Meeting, 1997, pp. 1–9.
[37] 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 (10) (2008) 4825–4852.
[38] R.D. Henderson, Details of the drag curve near the onset of vortex shedding, Phys. Fluids 7 (9) (1995) 2102–2104.
[39] C. Williamson, The natural and forced formation of spot-like vortex dislocations in the transition of a wake, J. Fluid Mech. 243 (1992) 393–441.
[40] C.H. Williamson, Vortex dynamics in the cylinder wake, Annu. Rev. Fluid Mech. 28 (1) (1996) 477–539.
[41] 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 (1) (2005) 1–31.
[42] D. Le, B. Khoo, K. Lim, An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains, Comput. Methods Appl. Mech. Eng. 197 (25–28) (2008) 2119–2130.
[43] D.-V. Le, B.C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, J. Comput. Phys. 220 (1) (2006) 109–138.
[44] T. Johnson, V. Patel, Flow past a sphere up to a Reynolds number of 300, J. Fluid Mech. 378 (1999) 19–70.
[45] P. Ploumhans, G. Winckelmans, J.K. Salmon, A. Leonard, M. Warren, Vortex methods for direct numerical simulation of three-dimensional bluff body flows: application to the sphere at re= 300, 500, and 1000, J. Comput. Phys. 178 (2) (2002) 427–463.
[46] B. Le Clair, A. Hamielec, H. Pruppacher, A numerical study of the drag on a sphere at low and intermediate reynolds numbers, J. Atmos. Sci. 27 (2) (1970) 308–315.
[47] F.W. Roos, W.W. Willmarth, Some experimental results on sphere and disk drag, AIAA J. 9 (2) (1971) 285–291.
[48] G.S. Constantinescu, K.D. Squires, LES and DES investigations of turbulent flow over a sphere at Re= 10,000, Flow, Turbul. Combust. 70 (1–4) (2003) 267–298.
[49] G. Constantinescu, K. Squires, Numerical investigations of flow over a sphere in the subcritical and supercritical regimes, Phys. Fluids 16 (5) (2004) 1449–1466.
[50] R. Campregher, J. Militzer, S.S. Mansur, A.d. Silveira Neto, Computations of the flow past a still sphere at moderate Reynolds numbers using an immersed boundary method, J. Brazil. Soc. Mech. Sci. Eng. 31 (4) (2009) 344–352.
[51] C. Mimeau, G.-H. Cottet, I. Mortazavi, Direct numerical simulations of three-dimensional flows past obstacles with a vortex penalization method, Comput. Fluids 136 (2016) 331–347.
[52] I. Rodriguez, R. Borrell, O. Lehmkuhl, A. Oliva, C.D. Perez-Segarra, Direct numerical simulation of the flow over a sphere at Re= 3700, in: Proceedings of the Sixth International Symposium On Turbulence Heat and Mass Transfer, J. Fluid Mechanics, 2009, p. 263287.
[53] G. Yun, D. Kim, H. Choi, Vortical structures behind a sphere at subcritical Reynolds numbers, Phys. Fluids 18 (1) (2006) 015102.
[54] H. Schlichting, Boundary-layer theory, 1979.
[55] S. Kaufmann, B. Homer, Craypat-cray x1 performance analysis tool, Cray User Group May (2003).
[56] H. Choi, W.-P. Jeon, J. Kim, Control of flow over a bluff body, Annu. Rev. Fluid Mech. 40 (2008) 113–139.
[57] M. Pastoor, L. Henning, B.R. Noack, R. King, G. Tadmor, Feedback shear layer control for bluff body drag reduction, J. Fluid Mech. 608 (2008) 161.
[58] D. Barros, J. Borée, B.R. Noack, A. Spohn, Resonances in the forced turbulent wake past a 3d blunt body, Phys. Fluids 28 (6) (2016) 065104.
[59] G. Rigas, A.S. Morgans, J.F. Morrison, Weakly nonlinear modelling of a forced turbulent axisymmetric wake, J. Fluid Mech. 814 (2017) 570–591.
[60] B. Herrmann, P. Oswald, R. Semaan, S.L. Brunton, Modeling synchronization in forced turbulent oscillator flows, Commun. Phys. 3 (1) (2020) 1–9.
[61] A. Sohankar, C. Norberg, L. Davidson, Low-reynolds-number flow around a square cylinder at incidence: study of blockage, onset of vortex shedding and outlet boundary condition, Int. J. Numer. Methods Fluids 26 (1) (1998) 39–56.
[62] A. Sharma, V. Eswaran, Heat and fluid flow across a square cylinder in the two-dimensional laminar flow regime, Numer. Heat Transf. Part A Appl. 45 (3) (2004) 247–269.
[63] A.K. Sahu, R. Chhabra, V. Eswaran, Two-dimensional unsteady laminar flow of a power law fluid across a square cylinder, J. Nonnewton Fluid Mech. 160 (2–3) (2009) 157–167.
[64] L.H. Thomas, Elliptic problems in linear difference equations over a network, Watson Sci. Comput. Lab. Rept. 1 (1949). Columbia University, New York