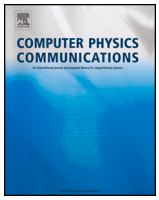
Computer Physics Communications 315 (2025) 109741

Contents lists available at ScienceDirect

Computer Physics Communications

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



Computational Physics

# A local and explicit forcing correction for Lagrangian immersed boundary methods

Giovanni Vagnoli [a], Martino Andrea Scarpolini [a,d], Roberto Verzicco [a,b,c], Francesco Viola [a,*]

[a] Gran Sasso Science Institute (GSSI), Viale Luigi Rendina, 26, L'Aquila, 67100, Italy
[b] Dipartimento di Ingegneria Industriale, University of Rome "Tor Vergata", Via del Politecnico 1, Rome, 00133, Italy
[c] Department of Physics, Mesa+ Institute, and J. M. Burgers Centre for Fluid Dynamics, University of Twente, AE Enschede, Twente, 7500, the Netherlands
[d] INFN Tor Vergata, Via della Ricerca Scientifica 1, Rome, 00133, Italy

ARTICLE INFO

The review of this paper was arranged by Prof. Weigel Martin

*Keywords:*
Immersed boundary method
Direct forcing
Moving least squares

ABSTRACT

Lagrangian immersed boundary methods (IBM) are widely used to study flows with complex and moving boundaries, such as biological flows. However, it has been noted that in some cases the resulting no-slip condition at the wet tissues is inaccurate. In this work, we propose an improved technique to evaluate the IBM forces with the aim of reducing slip and transpiration velocities at the immersed wet surfaces. In the framework of Moving Least Squares transfer function (MLS-IBM), we first formulate the problem of determining the IB forces in an implicit manner. Although this approach enforces the no-slip condition to machine precision, we demonstrate that this method is unaffordable for moderate/high Reynolds number flows. Nevertheless, insight of the implicit formulation reveals that it is possible to derive analytically a local correction to the IB forcing via an approximate factorisation of the system matrix of the implicit IBM. The resulting correction is fully explicit and it can be easily implemented in existing IB codes and extended to other Lagrangian IBMs. Our approach is then tested and compared against standard IBM-MLS (also in its iterative version) in a series of benchmark flows, including fixed and moving rigid bodies, and finally in the biological flow within a rigid aorta. Importantly, the proposed correction is seen to greatly improve the consistency and convergence properties of the IBM-MLS.

## 1. Introduction

The Immersed Boundary Method (IBM) is a valuable tool for investigating fluid flows in presence of moving boundaries due to its ability to deal with complex geometries. Introduced for biological flows [1], it has been widely used in the context of fluid-structure interaction (FSI) [2–6]. In IBM the Eulerian mesh, used to discretise the flow equations, does not conform to the body and the no slip condition is imposed on the fluid using ad hoc volume forces, intended to reproduce the effects of the wet solid boundaries on the fluid. This allows the use of a structured Eulerian mesh that has the possibility of exploiting highly efficient numerical solvers such as finite difference schemes along with the approximate tridiagonal factorisation of the viscous terms and the use of Fast Fourier Transform to solve the elliptic equation for pseudopressure [7,8].

Several approaches for applying and discretising IBM forcing have been proposed over the years (see the review papers [9–11]). A convenient approach is the so-called *explicit direct forcing Lagrangian method*, in which the forcing is calculated at some points located on the body surface (Lagrangian markers) and then transferred to the Eulerian grid [12,13].

A key aspect of Lagrangian IBM is the choice of the transfer function used to build (i) the interpolation operator $\mathcal{I}$ to interpolate the Eulerian velocity field on the Lagrangian markers and (ii) the corresponding spreading operator $\mathcal{S}$ needed to spread the IB forcing from the Lagrangian markers to the Eulerian grid. Specifically, regularised Dirac functions and Moving Least Squares (MLS) kernels have been used as transfer functions in continuous [14] and discrete [13] IBM, respectively.

Although being numerically efficient, explicit direct forcing methods have been observed not to spread correctly the IBM forcing from the Lagrangian markers to the Eulerian grid required to satisfy the boundary condition. This is mainly due to the non-reciprocity of the spreading and interpolation operators, resulting in a slip velocity error that can be large [15]. Several amendments to the Lagrangian IBM formulation have been proposed to circumvent this problem, such as the implicit

* Corresponding author.
*E-mail address:* francesco.viola@gssi.it (F. Viola).

https://doi.org/10.1016/j.cpc.2025.109741
Received 17 December 2024; Received in revised form 27 June 2025; Accepted 1 July 2025
Available online 10 July 2025

treatment of forcing, in which the immersed boundary forces are obtained by the solution of a linear system [16]; or projection approaches [17], whose forcing is regarded as a Lagrange multiplier for the boundary condition constraint. Despite these approaches allow to impose the boundary conditions to machine precision, they require deep changes of the numerical solver structure and greatly increase the computational costs. Eventually, even employing ad hoc preconditioners as in [18], three-dimensional simulations become unaffordable.

Alternatively, explicit methods can be improved through forcing corrections. For example, a simple improvement is the introduction of forcing iterations [19], which increase the precision of the method with minimal modifications to its numerical implementation and a tolerable increase of computational costs. However, it has been noted that, even if decreasing the overall boundary error, the iterative approach fails to reduce it beyond a problem depending threshold. In fact, if the number of iterations is increased, an error saturation is always observed. In [15] and more recently in [20,21], a purely geometrical correction factor depending on the shape of the transfer functions is introduced to correct the IB forces evaluated on the markers. Similarly, the work of [22] introduces a global correction considering also the underlying fluid flow to improve the evaluation of the Lagrangian forces. Those methods are capable of reducing the error at the wet boundary without any forcing iteration. In [23], the pressure equation is modified in order to explicitly impose the pressure boundary condition, which in turn reduces the boundary error when there are large pressure gradients through the walls.

In this paper, we propose an improvement of explicit direct forcing methods that is capable of reducing slip error. The correction introduced here is explicit and takes into account the local spreading error of the forcing from each Lagrangian marker by means of an appropriate approximation of the matrix of the IB forcing system. Furthermore, the correction also dynamically accounts for the instantaneous flow over the immersed surfaces. In this way, it is possible to estimate with improved precision the immersed boundary forcing without solving any linear systems.

The remainder of the paper is organised as follows. In Section 2 we introduce the numerical method for the flow equations and review the explicit direct forcing method; in Section 3 we introduce the implicit formulation of IBM in the framework of MLS transfer functions and the approximate factorisation required to impose the explicit correction; in Section 4 the novel method is tested against some classical but significant fluid flow to validate the results and show the precision of the correction. Finally, we present the conclusions in Section 5.

## 2. Numerical method

### 2.1. Flow solver

Let us consider a three-dimensional incompressible flow governed by the Navier-Stokes equations, which in nondimensional form read:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} + \mathbf{f}, \tag{1a}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{1b}$$

where $\mathbf{u}$ and $p$ are the flow velocity and pressure, respectively, $\mathbf{f}$ is the volume force acting on the fluid, including the IB forcing. $Re = UL/\nu$ is the Reynolds number based on the kinematic viscosity of the fluid $\nu$, a characteristic length $L$ and a characteristic velocity $U$. The no slip condition $\mathbf{u} = \mathbf{U}^b$ is imposed on each solid wall, $\mathbf{U}^b$ being the wall velocity. The governing equations are integrated using a fractional step method [7,24]. Equation (1a) is advanced from time $t^n$ to $t^{n+1}$ using an implicit Crank-Nicholson scheme for the viscous term and an explicit Adam-Bashfort scheme for the non-linear terms, which yields the following predictor equation:

$$\frac{\hat{\mathbf{u}} - \mathbf{u}^n}{\Delta t^n} = -\nabla p^n + \frac{3}{2}\mathbf{H}^n - \frac{1}{2}\mathbf{H}^{n-1} + \frac{1}{2Re}\nabla^2 (\hat{\mathbf{u}} + \mathbf{u}^n), \tag{2}$$

where $\mathbf{u}^n$ and $p^n$ are the velocity and pressure at time $t^n$, $\hat{\mathbf{u}}$ is the provisional velocity, $\Delta t^n = t^{n+1} - t^n$ is the time step and $\mathbf{H}$ contains the non-linear terms. Note that in this first predictor step there is no forcing term on the right-hand-side. Equation (2) is discretised in space using a second-order accurate centred finite difference schemes on a Cartesian staggered grid $\mathcal{G}$, giving a linear system of algebraic equations, which is solved with an approximate factorisation of the system matrix [7]. IB forcing is applied to the provisional velocity $\hat{\mathbf{u}}$, updating the velocity as follows:

$$\mathbf{u}^* = \hat{\mathbf{u}} + \mathbf{f}\,\Delta t^n, \tag{3}$$

where $\mathbf{u}^*$ is a non-solenoidal vector field and $\mathbf{f}$ is the MLS-IB forcing, which will be discussed in detail in the next section. To enforce the divergence-free constraint, we introduce the pseudopressure $\varphi$, such that:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t^n \nabla \varphi. \tag{4}$$

Applying the divergence operator to the above equation and requiring $\nabla \cdot \mathbf{u}^{n+1} = 0$ leads to the following elliptic equation for $\varphi$:

$$\nabla^2 \varphi = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t^n}, \tag{5}$$

which is solved using FFTs in the periodic directions. The pressure is then calculated as follows:

$$p^{n+1} = p^n + \varphi - \frac{\Delta t^n}{2Re}\nabla^2 \varphi. \tag{6}$$

### 2.2. Explicit direct forcing immersed boundary treatment

To impose the no slip boundary condition by the explicit direct forcing method [13], we discretize the surface of the body by $N_t$ triangular elements, whose centroids $\mathbf{X}_l = \left[X_l^{(1)}, X_l^{(2)}, X_l^{(3)}\right]^T$, $l = 1, \ldots, N_t$, are denoted Lagrangian markers. For each marker, a tridimensional support domain $S_l$ is created centred on the marker, as shown in Fig. 1. Each domain contains $N_E$ Eulerian points $\mathbf{x}_k$, depending on the dimension of the support domain. Typically, 3 points are considered in each direction, for a total of 9 points in 2D problems and 27 points in 3D; therefore, $S_l$ is defined as follows:

$$S_l = \left\{ \mathbf{x}_k \in \mathcal{G} : \left| x_k^{(i)} - X_l^{(i)} \right| \leq \frac{3}{2}\Delta x_k^{(i)},\ i = 1,2,3 \right\}, \tag{7}$$

where $\Delta x_k^{(i)}$ is the local Eulerian grid spacing in the $i$-th direction and the coordinates $\mathbf{x}_k = \left[x_k^{(1)}, x_k^{(2)}, x_k^{(3)}\right]^T$ are the location of the $k$-th Eulerian grid point.

The transfer functions required to interpolate velocity from the Eulerian grid to the Lagrangian markers and then spread the forcing from the markers to the Eulerian grid are based on the Moving Least Squares kernels $\phi\left(\mathbf{x}_k, \mathbf{X}_l\right)$ [13,25], which are built to minimise the quadratic interpolation error of the velocity. In the following, uppercase letters refer to quantities defined on the Lagrangian markers, while lowercase letters refer to fields defined on the Eulerian grid. The velocity field $\hat{u}_i$ interpolated at a point $\mathbf{x}$ within the supporting domain of $\mathbf{X}_l$ is given by the following:

$$\hat{U}_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\,\mathbf{a}(\mathbf{x}), \tag{8}$$

where $\mathbf{p}(\mathbf{x}) = [1, x^{(1)}, x^{(2)}, x^{(3)}]^T$ is a linear basis function and $\mathbf{a}(\mathbf{x})$ is the vector of coefficients, which is obtained by minimising the weighted $L_2$-norm $J = \sum_{k \in S_l} W\left(\mathbf{x} - \mathbf{x}_k\right)\left[\mathbf{p}^T\left(\mathbf{x}_k\right)\mathbf{a}(\mathbf{x}) - \hat{u}_i\left(\mathbf{x}_k\right)\right]^2$ with respect to $\mathbf{a}(\mathbf{x})$. The weight function $W$ is defined as a product of one-dimensional functions $W(\mathbf{x}) = \Pi_{i=1}^3 W_i(\mathbf{x})$, being the 1D weight function built as follows:

$$W_i(\mathbf{x}) = \begin{cases} e^{-(r_i/\varepsilon)^2} & r_i \leq 1, \\ 0 & r_i > 1, \end{cases} \tag{9}$$

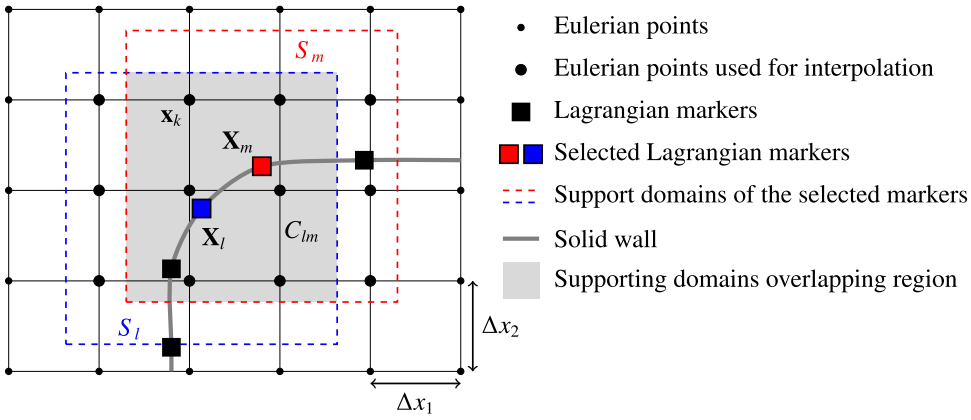



**Fig. 1.** Bidimensional sketch of the setting required to perform the IB treatment. For each marker, its supporting domain is represented by a dashed line. The shaded area is the region of overlapping of the supporting domains for the highlighted markers.

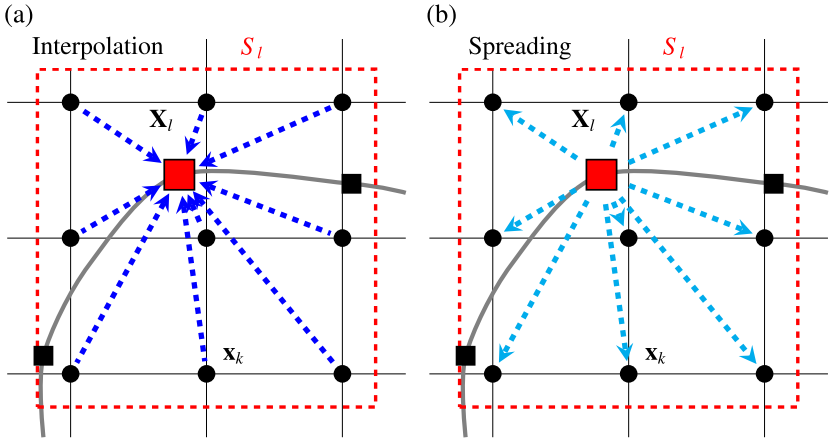



**Fig. 2.** (a) Sketch of the action of the interpolation operator: the information is transferred from the Eulerian grid to the Lagrangian marker; (b) Sketch of the action of the spreading operator: the information is transferred from the Lagrangian marker to the Eulerian grid.

with $r_i = |\mathbf{x}|/H_i$. $H_i = 1.5\Delta x^{(i)}$ is the dimension of the supporting domain in the $i$-th direction and $\epsilon$ determining the width of the weight function is typically set in the range $[0.3 - 0.7]$.

The minimisation of $J$ leads to the linear system of equations $A(\mathbf{x})\,\mathbf{a}(\mathbf{x}) = B(\mathbf{x})\left[\hat{u}_i(\mathbf{x}_1),\ldots,\hat{u}_i\left(\mathbf{x}_{N_E}\right)\right]^T$, where $A(\mathbf{x}) = \sum_{k\in S_l} W(\mathbf{x}-\mathbf{x}_k)\,\mathbf{p}(\mathbf{x}_k)\,\mathbf{p}^T(\mathbf{x}_k) \in \mathbb{R}^{4\times4}$ and $B(\mathbf{x}) = \left[W(\mathbf{x}-\mathbf{x}_1)\,\mathbf{p}(\mathbf{x}_1),\ldots,W\left(\mathbf{x}-\mathbf{x}_{N_E}\right)\mathbf{p}\left(\mathbf{x}_{N_E}\right)\right] \in \mathbb{R}^{4\times N_E}$. We rewrite equation (8) on the marker $\mathbf{X}_l$ as follows:

$$\hat{U}_i(\mathbf{X}_l) = \mathbf{\Phi}^T(\mathbf{X}_l)\left[\hat{u}_i(\mathbf{x}_1),\ldots,\hat{u}_i\left(\mathbf{x}_{N_E}\right)\right]^T, \tag{10}$$

where $\mathbf{\Phi}^T(\mathbf{X}_l) = \left[\phi(\mathbf{x}_1,\mathbf{X}_l),\ldots,\phi\left(\mathbf{x}_{N_E},\mathbf{X}_l\right)\right] = \mathbf{p}^T(\mathbf{X}_l)\,A^{-1}(\mathbf{X}_l)\,B(\mathbf{X}_l) \in \mathbb{R}^{N_E}$ are the MLS kernels.

The latter equation defines the MLS interpolator operator $\mathcal{I}\left[\hat{u}_i\right](\mathbf{X}_l)$, which is rewritten as follows (see Fig. 2a):

$$\hat{U}_i(\mathbf{X}_l) = \mathcal{I}\left[\hat{u}_i\right](\mathbf{X}_l) = \sum_{k\in S_l} \phi(\mathbf{x}_k,\mathbf{X}_l)\,\hat{u}_i(\mathbf{x}_k), \tag{11}$$

where the square and round brackets indicate that the interpolator takes as input the Eulerian velocities $\hat{u}_i$ and that provides as output the interpolated velocity on the position $\mathbf{X}_l$. Then, according to the explicit direct forcing, the immersed boundary forcing evaluated at the Lagrangian marker reads:

$$F_i^E(\mathbf{X}_l) = \frac{U_i^b(\mathbf{X}_l) - \mathcal{I}\left[\hat{u}_i\right](\mathbf{X}_l)}{\Delta t^n}. \tag{12}$$

Using the same basis functions, the spreading of the forcing $\mathcal{S}\left[F_i^E\right]$ from the Lagrangian markers to the $k$-th element of the Eulerian grid is defined as (see Fig. 2b):

$$f_i^E(\mathbf{x}_k) = \mathcal{S}\left[F_i^E\right](\mathbf{x}_k) = \sum_{l=1}^{N_t} c_l\,\phi(\mathbf{x}_k,\mathbf{X}_l)\;F_i^E(\mathbf{X}_l), \tag{13}$$

where the square and round brackets indicate here that the spreading operator takes as input the Lagrangian force, which is smeared on the Eulerian grid points $\mathbf{x}_k$.

The coefficient $c_l$ is such that the total force acting on the fluid is not changed by the transfer [13]:

$$c_l = \frac{\Delta V_l}{\sum_{k\in S_l}\phi(\mathbf{x}_k,\mathbf{X}_l)\,\Delta v_k}, \tag{14}$$

where $\Delta v_k = \Delta x_k^{(1)}\Delta x_k^{(2)}\Delta x_k^{(3)}$ is the volume of the local Eulerian cell and $\Delta V_l = A_l h_l$ is a volume associated to each Lagrangian marker, with $A_l$ the area of the $l$-th triangle and $h_l = 1/3\sum_{k\in S_l}\phi(\mathbf{x}_k,\mathbf{X}_l)\left(\Delta x_k^{(1)} + \Delta x_k^{(2)} + \Delta x_k^{(3)}\right)$.

Finally, the updated Eulerian velocity is computed using equation (3). Since the IB forcing is directly evaluated (through equations (12) and (13)) for each Lagrangian marker independently, the method just described is said *explicit* forcing.

As anticipated in the introduction, the explicit direct forcing method can be enforced in an iterative fashion [19] in order to reduce slippage and transpiration at the wet interface. In this case, the sequence of equations (12), (13) and (3) is solved multiple times (until a certain

threshold is met), where the intermediate velocity $u_i^*$ is updated with the IB-forcing and then used to evaluate a new IB-forcing.

## 3. Improved immersed boundary methods

In the explicit direct forcing method described above [13], the IB forcing is evaluated at the Lagrangian level and then distributed to the surrounding fluid nodes using the spreading operator. This method, however, may produce a large transpiration error at the wet boundary, especially in unsteady flows with a large pressure jump across the surface [26]. The error in the no slip condition has a threefold reason. Firstly, as shown in Fig. 1, there exists an overlap region $C_{lm} = S_l \cap S_m$ in which the forcing is spread to the Eulerian grid from two markers $\mathbf{X}_l$ and $\mathbf{X}_m$ independently of each other. As an example, both markers add a contribution to the Eulerian location $\mathbf{x}_k \in C_{lm}$.

Secondly, the spreading operator is not the inverse of the interpolator: therefore, the IB force $f_i^E$, obtained from the spreading of $F_i^E$, if interpolated back to the Lagrangian markers, does not reproduce exactly the source $F_i^E$. The latter can be easily seen if we focus on a single Lagrangian marker, for which equation (13) rewrites as follows:

$$\mathcal{S}\left[F_i^E\right](\mathbf{x}_k) = c_l\,\phi(\mathbf{x}_k,\mathbf{X}_l)\,F_i^E(\mathbf{X}_l). \tag{15}$$

Interpolating back the forcing to the Lagrangian marker leads to the following equation:

$$\mathcal{M}\left[F_i^E\right](\mathbf{X}_l) = F_i^E\,c_l \sum_{k\in S_l} \phi^2(\mathbf{x}_k,\mathbf{X}_l) \neq F_i^E(\mathbf{X}_l), \tag{16}$$

where $\mathcal{M}\left[F_i^E\right](\mathbf{X}_l) = (\mathcal{I}\circ\mathcal{S})\left[F_i^E\right](\mathbf{X}_l)$ is the composition of the spreading and interpolation operator, that takes a Lagrangian field, spreads it onto the Eulerian grid points and finally interpolates it back to the marker. Since the coefficients $c_l$ are chosen to satisfy the conservation of momentum [13], in general $c_l \sum_{k\in S_l}\phi^2(\mathbf{x}_k,\mathbf{X}_l) \neq 1$, then, even for a single marker, $\mathcal{M}$ it is not the identity operator.

An additional source of error is that, in the fractional step method, the IB forcing is applied to the provisional velocity $u_i^*$ rather than $u_i^{n+1}$. In fact, to obtain the final solenoidal velocity, the projection step (4) may eventually perturb the value of the velocity at the immersed boundary. To reduce this error, we should impose the IB forcing and the incompressibility constraint at the same time, as was done for 2D flows in [17]. Alternatively, as suggested in [27] this transpiration error can be reduced by iterating the IB forcing step (3) and the solenoidal projection (4).

In this work we aim at reducing the first and second sources of error, by proposing a local correction to the interpolation and spreading operators. In particular, following [16] we first implemented the implicit IB method in the MLS framework, which evaluates the Lagrangian force to be transferred to the fluid in order to impose the no slip boundary condition at machine precision. However, through systematic tests, we will show that the implicit method entails the inversion of an ill-conditioned matrix and, as a consequence, this approach becomes computationally unaffordable for three-dimensional problems and relatively large Reynolds number. In order to avoid this bottleneck, we derive a novel explicit method to efficiently reduce the slip error at the wet boundaries with respect to explicit direct forcing.

### 3.1. Implicit IB method

Since the no slip condition is prescribed on the Lagrangian markers, it is convenient to define the IB forcing of equation (3) as the spread of an unknown Lagrangian field $F_i(\mathbf{X}_l)$, that is, $f_i(\mathbf{x}_k) = \mathcal{S}\left[F_i\right](\mathbf{x}_k)$. In this setting, equation (3) recasts as follows:

$$u_i^*(\mathbf{x}_k) = \hat{u}_i(\mathbf{x}_k) + \mathcal{S}\left[F_i\right](\mathbf{x}_k)\,\Delta t^n, \tag{17}$$

which is then interpolated on the Lagrangian marker using the linear operator $\mathcal{I}$:

$$U_i^b(\mathbf{X}_l) = \mathcal{I}\left[\hat{u}_i\right](\mathbf{X}_l) + \mathcal{M}\left[F_i\right](\mathbf{X}_l)\,\Delta t^n, \tag{18}$$

where at the left-hand-side we impose the condition that the interpolated fluid velocity at the Lagrangian maker matches the velocity at the boundary, i.e. $U_i^b(\mathbf{X}_l) = \mathcal{I}\left[u_i^*\right](\mathbf{X}_l)$. The only unknowns of equation (18) are the Lagrangian values $F_i$. From the definition of the interpolation and spreading operators, the last term of the right-hand side can be rewritten as follows:

$$\begin{aligned}\mathcal{M}\left[F_i\right](\mathbf{X}_l) &= \sum_{k\in S_l}\phi(\mathbf{x}_k,\mathbf{X}_l)\sum_{m=1}^{N_t} c_m\,\phi(\mathbf{x}_k,\mathbf{X}_m)\,F_i(\mathbf{X}_m)\\ &= \sum_{m=1}^{N_t} c_m\,F_i(\mathbf{X}_m)\sum_{k\in S_l}\phi(\mathbf{x}_k,\mathbf{X}_l)\,\phi(\mathbf{x}_k,\mathbf{X}_m)\\ &= \sum_{m=1}^{N_t} M_{lm}\,c_m\,F_i(\mathbf{X}_m),\end{aligned} \tag{19}$$

where we have introduced the matrix $M \in \mathbb{R}^{N_t\times N_t}$:

$$M_{lm} = \sum_{k\in C_{lm}} \phi(\mathbf{x}_k,\mathbf{X}_l)\,\phi(\mathbf{x}_k,\mathbf{X}_m). \tag{20}$$

Hence, equation (18) can be rewritten as a linear system of equations:

$$\sum_{m=1}^{N_t} M_{lm}\,c_m\,F_i(\mathbf{X}_m) = F_i^E(\mathbf{X}_l), \quad l = 1,\dots,N_t, \tag{21}$$

where the size of the matrix $M$ scales with the number of Lagrangian markers $N_t$. It is crucial to note that $N_t$ and the total number of Eulerian grid points are related, as we will show below. In particular, the larger the number of Eulerian points, the larger the number of Lagrangian markers: therefore, in flow with large Reynolds number or in presence of complex wall geometries, we cannot resort to direct or matrix-based methods to solve the linear system, but we are forced to use matrix-free methods, such as BiCGSTAB or GMRES.

The corresponding $F_i$ satisfying the system equations (21) can finally be spread onto the Eulerian grid to obtain $f_i$ and thus $u_i^*$ (by means of Eq. (3)) satisfying no slip on the Lagrangian markers to machine precision.

### 3.2. Analysis of the system matrix for implicit IB method

By definition, the matrix of the IB system $M$ is symmetric and positive definite. In fact, we can construct the rectangular matrix $D \in \mathbb{R}^{N_t\times N_E}$ such that $D_{lk} = \phi(\mathbf{x}_k,\mathbf{X}_l)$, which gives $M = DD^T \in \mathbb{R}^{N_t\times N_t}$. Using this notation, we can express the interpolation and spreading operators as follows:

$$\begin{aligned}\begin{bmatrix}\mathcal{I}[u](\mathbf{X}_1)\\ \vdots\\ \mathcal{I}[u](\mathbf{X}_{N_t})\end{bmatrix} &= D\begin{bmatrix}u(\mathbf{x}_1)\\ \vdots\\ u(\mathbf{x}_{N_E})\end{bmatrix},\\ \begin{bmatrix}\mathcal{S}[U](\mathbf{x}_1)\\ \vdots\\ \mathcal{S}[U](\mathbf{x}_{N_E})\end{bmatrix} &= D^T\begin{bmatrix}c_1\,U(\mathbf{X}_1)\\ \vdots\\ c_{N_t}\,U(\mathbf{X}_{N_t})\end{bmatrix}.\end{aligned} \tag{22}$$

Furthermore, the entries of $M$ depend only on the grid spacing of the Lagrangian markers $h$ and of the Eulerian points $\Delta x$ (in particular on their ratio $h/\Delta x$) and therefore are time independent in a flow around a fixed body. Given the shape of the MLS functions, each Lagrangian marker interacts only with the surrounding ones, resulting in a sparse matrix. In particular, $M_{lm} = 0$ if and only if $C_{lm} = \emptyset$.

In order to investigate the numerical properties of the matrix of the system, let us consider a 2D ring of unitary diameter, with $N_t$ Lagrangian markers $\mathbf{X}_l$ equally spaced, which is immersed in a non-staggered Cartesian uniform grid made of 300 points in each direction. Since the grid is

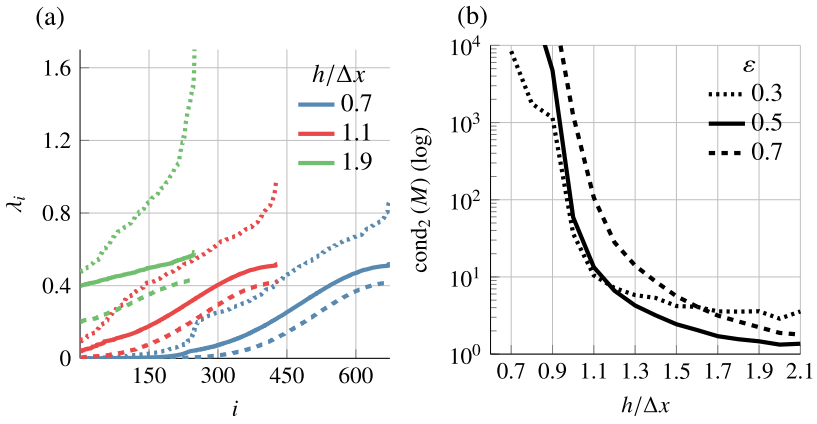



**Fig. 3.** (a) Spectrum and (b) 2-condition number of the system matrix $M$ for different values of $h/\Delta x$ and $\varepsilon$. Each colour refers to a different value of $h/\Delta x$, while the various line types identify the value of $\varepsilon$ in the MLS interpolation (see equation (9)).

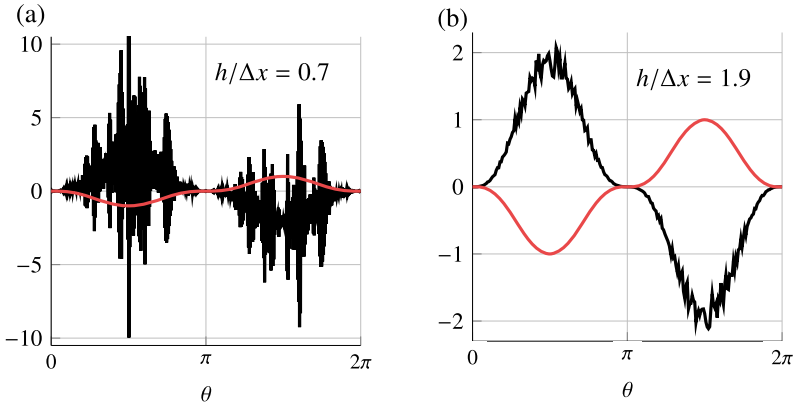



**Fig. 4.** IB forcing $F$ (—) along the ring for (a) $h/\Delta x = 0.7$ and (b) $h/\Delta x = 1.9$ compared to $U$ (—). As $h/\Delta x$ is reduced, $F$ shows large oscillations.

not staggered, the system matrix is the same for both spatial directions.

In Fig. 3a we report the eigenvalues $\lambda_i$ with $i = 1, \ldots, N_t$, of $M$ for different values of $h/\Delta x$, obtained by varying the number of Lagrangian markers. For each value of $h/\Delta x$, the eigenvalues are sorted in ascending order. The maximum value of the eigenvalues $\lambda_{max}$ shows little variation as $h/\Delta x$ decreases, while the minimum value $\lambda_{min}$ quickly decreases to zero. In contrast, the effect of a reduction of $\varepsilon$ in the MLS interpolation (see Eq. (9)) is to increase both $\lambda_{max}$ and $\lambda_{min}$. To better explore the variations of the eigenvalues, Fig. 3b shows the 2-condition number of $M$, defined as $\mathrm{cond}_2\,(M) = \lambda_{max}/\lambda_{min}$. Each curve represents the variations of $\mathrm{cond}_2\,(M)$ with $h/\Delta x$ for a fixed value of $\varepsilon$. The curves, thus increase steeply as $h/\Delta x$ is reduced and, in particular, for $h/\Delta x < 1$ the condition number starts to diverge regardless of $\varepsilon$.

Next, we consider an arbitrary field $u\left(x_1, x_2\right) = -\cos(\pi\, x_1)\, \sin(\pi\, x_2)$, and we evaluate the force $F\left(\mathbf{X}_l\right)$ required to set $u = 0$ on the ring using equation (21). Fig. 4 shows the IB forcing on each Lagrangian marker along the angular coordinate $\theta \in [0, 2\pi)$ of the ring, for $h/\Delta x = 0.7$ and $h/\Delta x = 1.9$, respectively, together with $U$, that is, the exact value of $u$ on the markers. For $h/\Delta x = 0.7$, the solution shows strong oscillations on a length scale much smaller than variation of $U$ along the ring; on the contrary, for $h/\Delta x = 1.9$, the solution is more regular, with some small oscillations where $U$ reaches its extreme points. This behaviour resembles Runge's phenomenon, typically observed in interpolation theory: in particular, for $h/\Delta x$ small, the spreading and interpolation operators act on a stencil that is far from the optimal one, resulting in undesired oscillations.

Unfortunately, the conditioning of the system matrix is not the unique constraint on $h/\Delta x$: In fact, this ratio quantifies the number of Lagrangian markers where the no slip condition is imposed. In an IB framework, the solid walls are represented only by a point cloud and, therefore, it is mandatory to have enough points to impose the boundary condition with sufficient spatial resolution. This last constraint clearly requires a sufficiently small value of $h/\Delta x$, which is typically $\lesssim 1$. This value of $h/\Delta x$, as shown above, results in a ill conditioned matrix $M$, thus increasing the number of iterations required to solve the linear system and, in turn, slowing down the temporal iteration.

To address this issue in a more realistic framework, we consider a 3D flow around an horizontally oscillating sphere of diameter $D$. The computational domain is $6\,D \times 6\,D \times 12\,D$, a uniform grid size $\Delta x = 0.04\,D$ is used, while the time step is kept fixed at $\Delta t = 5 \times 10^{-5}\, D/U$. The centre of the sphere is initially at the centre of the computational box and the law of motion of any point of the body $\mathbf{X}$ is described as follows:

$$\mathbf{X}\,(t) = \mathbf{X}\,(0) + A\,\sin\,(\omega\, t)\,\mathbf{e}_3, \tag{23}$$

being $A$ and $\omega$ the amplitude and pulsation of the oscillation, respectively. The Reynolds number is based on the diameter and maximum velocity of the sphere, that is, $Re = A\omega D/\nu$, and is set equal to 30, with $A$ equal to $D$. $N_t$ Lagrangian markers are distributed over the sphere surface with the Lagrangian to Eulerian grid ratio varied as $h/\Delta x \in [1.04, 1.18, 1.38]$. Notice that in this case a staggered Eulerian grid is employed, thus the matrix $M$ must be constructed for each spatial direction, since three different Eulerian grids are used for each of

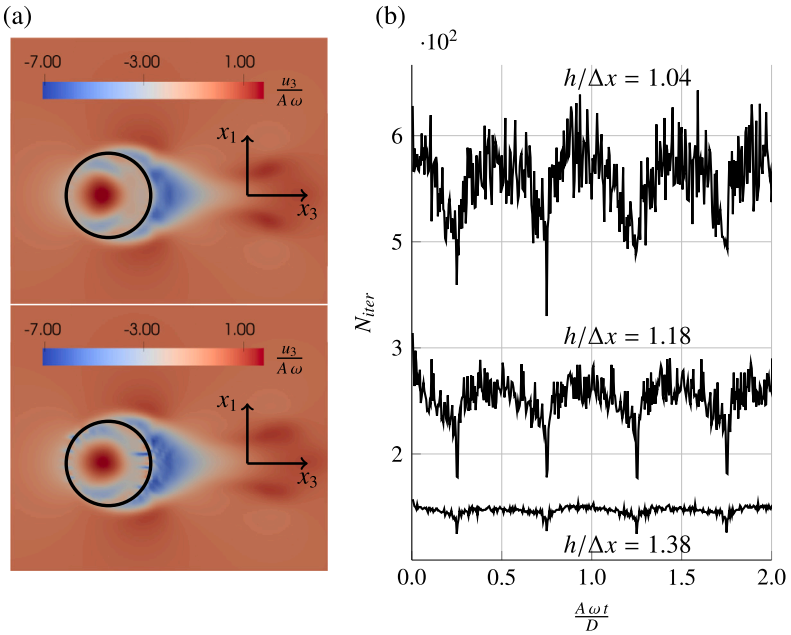



**Fig. 5.** a) Streamwise component $u_3$ of the flow around the horizontally oscillating sphere at maximum left displacement and maximum acceleration of the sphere for (top) $h/\Delta x = 1.04$ and (bottom) $h/\Delta x = 1.38$. b) Total number of iterations required to solve the linear system (21) for different values of $h/\Delta x \in [1.04, 1.18, 1.38]$.

**Table 1**
Comparison of the mean value of WCT over a period required to perform one temporal iteration for the horizontally oscillating sphere of the implicit method with respect to explicit direct forcing, $\text{WCT}^E$.

| | $h/\Delta x = 1.04$ | $h/\Delta x = 1.18$ | $h/\Delta x = 1.38$ |
|---|---|---|---|
| $\frac{\langle \text{WCT} \rangle}{\langle \text{WCT}^E \rangle}$ | 412.9 | 185.4 | 50.7 |

them. Furthermore, the system matrices cannot be explicitly assembled, and the linear system (21) is solved in an iterative way using BiCGSTAB, which is stopped when the residual of the linear system is lower than $1 \times 10^{-8}$ in non dimensional units.

After a preliminary transient, the flow reaches a periodic motion: this condition is particularly critical for the implicit method because the system matrix changes over time.

In Fig. 5a, the horizontal component $u_3$ of the velocity of the flow around the sphere in the $x_1 - x_3$ plane when the amplitude of the oscillation is maximum is represented for $h/\Delta x = 1.04$ (top) and $h/\Delta x = 1.38$ (bottom). Far from the body, the flow is similar and the near wake is indistinguishable between the two cases. The major differences are observed near the surface: for $h/\Delta x = 1.04$ we do not observe any nonphysical flow through the sphere, whereas large $u_3$ oscillations are present for $h/\Delta x = 1.38$ near the surface of the sphere. Indeed, in the first case the density of Lagrangian markers is sufficient to impose the no slip condition on the sphere, while in the second case the forcing is applied in an insufficient number of points, leaving enough space to allow fluid leakage from the inside of the sphere and the outer flow, and vice versa.

In Fig. 5b, the total number of iterations $N_{iter}$ required to evaluate the IB forces in each spatial direction is shown for three different values of $h/\Delta x$. $N_{iter}$ shows a periodic behaviour, but its mean value is strongly influenced by $h/\Delta x$. In particular, $N_{iter}$ increases as $h/\Delta x$ is reduced.

The rise of $N_{iter}$ in turn increases the computational time, as shown in Table 1, where for the implicit method the values of the wall clock time (WCT) required to complete one temporal iteration averaged over a period $\langle \text{WCT} \rangle$ and normalised with respect to the wall clock time of explicit direct forcing $\langle \text{WCT}^E \rangle$ are reported. For the implicit method, the WCT increases as $h/\Delta x$ decreases monotonically, reaching more than 400 times the baseline explicit method if $h/\Delta x = 1.04$.

From the above discussion, it is clear that $h/\Delta x$ plays a fundamental role in evaluating the IB forces. However, we observed that in the range of $h/\Delta x$ for which the no slip boundary condition is applied with sufficient precision, computational cost skyrockets, thus preventing the use of implicit methods in more complex configurations or at larger Reynolds numbers.

### 3.3. An improved explicit IB method: MLS-κ

Due to the issues highlighted above, implicit methods cannot be used for solving large 3D problems and therefore we are forced to resort to explicit but approximate methods. Interestingly, the explicit direct forcing (see equation (12)) can be derived from the implicit method by means of an approximate factorisation of $M$. In fact, if $M$ is approximated as $M^{-1} \simeq \text{diag}\left[c_1, \ldots, c_{N_t}\right]$, i.e. a diagonal matrix whose entries are the coefficients $c_l$ defined by equation (14), then $F_i\left(\mathbf{X}_l\right) = F_i^E\left(\mathbf{X}_l\right)$. This approximation is based on the idea that the spreading operator $S$ should be designed to be the inverse of the interpolator $\mathcal{I}$. As noted above, however, this property does not hold because the supporting domains of the Lagrangian markers overlap (non diagonal terms of $M$), and also because the spreading operator is just an approximation of the inverse of $\mathcal{I}$. Nevertheless, this inexpensive approach is justified by the good results achieved, especially when applied in an iterative manner (see the end of Section 2), which can be proven [28] to be equivalent to a gradient descent method applied to the linear system (21). For this reason, it is not convenient to iterate several times the explicit forcing because the system matrix gets singular when $h/\Delta x$ is sufficiently small (typically $h/\Delta x = 0.7$ in this setting). Indeed, in most applications, the number of iterations is limited to 4-8 as the solution precision does not sufficiently increase further iterating.

In this work, we propose a new approximate factorisation of the system matrix by seeking a solution of the form:

$$F_i\left(\mathbf{X}_l\right) \simeq \kappa_i\left(\mathbf{X}_l\right)\, F_i^E\left(\mathbf{X}_l\right), \tag{24}$$

where the coefficients $\kappa_i\left(\mathbf{X}_l\right)$ are unknown. This decomposition is equivalent to approximating the inverse of $M$ as $M^{-1} \simeq$

**Algorithm 1** The MLS-$\kappa$ algorithm.

```
for i = 1,2,3 do
    for l = 1,...,N_t do
        Determine the support domain S_l of X_l;
        For each x_k ∈ S_l, compute the MLS basis functions φ(x_k, X_l);
        Interpolate the provisional velocity û_i onto the Lagrangian markers;
        Compute the explicit IB forcing F_i^E(X_l);
        Compute the c_l;
        if |M[F_i^E](X_l)| > tol then
            Compute the correction κ_i(X_l);
        else
            Set κ_i(X_l) = 1;
        end if
        Spread the corrected forcing κ_i(X_l) F_i^E(X_l) to the Eulerian grid.
    end for
end for
```

diag $\left[\kappa_i\left(\mathbf{X}_1\right),\ldots,\kappa_i\left(\mathbf{X}_{N_t}\right)\right]$. Introducing equation (24) into the linear system (21), we obtain:

$$\sum_{m=1}^{N_t} M_{lm}\, c_m\, \kappa_i\left(\mathbf{X}_m\right)\, F_i^E\left(\mathbf{X}_m\right) \simeq F_i^E\left(\mathbf{X}_l\right), \quad l=1,\ldots,N_t. \tag{25}$$

Given two markers $\mathbf{X}_l$ and $\mathbf{X}_m$ for which $C_{lm}\neq\emptyset$, we have $\|\mathbf{X}_m-\mathbf{X}_l\|=\mathcal{O}(h)$; therefore, if we assume continuity of the $\kappa_i$, we can write $\kappa_i\left(\mathbf{X}_l\right)\simeq\kappa_i\left(\mathbf{X}_m\right)$, which leads to the following:

$$\kappa_i\left(\mathbf{X}_l\right)\sum_{m=1}^{N_t} M_{lm}\, c_m\, F_i^E\left(\mathbf{X}_m\right) \simeq F_i^E\left(\mathbf{X}_l\right), \quad l=1,\ldots,N_t. \tag{26}$$

Neglecting the higher-order terms, we obtain the $N_t$ uncoupled equations for $\kappa_i\left(\mathbf{X}_l\right)$:

$$\kappa_i\left(\mathbf{X}_l\right)=\frac{F_i^E\left(\mathbf{X}_l\right)}{\sum_{m=1}^{N_t} M_{lm}\, c_m\, F_i^E\left(\mathbf{X}_m\right)}, \quad l=1,\ldots,N_t, \tag{27}$$

from which we can reconstruct the forcing by means of equation (24), yielding the following Eulerian forcing applied to the fluid:

$$f_i\left(\mathbf{x}_k\right)=\mathcal{S}\left[\kappa_i\, F_i^E\right]\left(\mathbf{x}_k\right)=\sum_{l=1}^{N_t} c_l\,\phi\left(\mathbf{x}_k,\mathbf{X}_l\right)\,\kappa_i\left(\mathbf{X}_l\right)\,F_i^E\left(\mathbf{X}_l\right). \tag{28}$$

To understand the physical meaning of the correction, equation (27) can be rewritten:

$$\kappa_i\left(\mathbf{X}_l\right)=\frac{F_i^E\left(\mathbf{X}_l\right)}{\mathcal{M}\left[F_i^E\right]\left(\mathbf{X}_l\right)}, \tag{29}$$

and we can thus interpret $\kappa_i\left(\mathbf{X}_l\right)$ as the ratio, at the Lagrangian level, of the force that we should impose to the fluid to satisfy the no slip condition (numerator) and the force that is actually applied to the fluid owing to the errors introduced by the spreading operation and by the overlapping of the supporting domains (denominator). Therefore, the multipliers $\kappa_i$ provide a local correction to these two errors. Importantly, equation (29) reveals that building the system matrix $M$ is not needed for computing $\kappa_i$, since $\mathcal{M}\left[F_i^E\right]\left(\mathbf{X}_l\right)=(\mathcal{I}\circ\mathcal{S})\left[F_i^E\right]\left(\mathbf{X}_l\right)$ is simply a consecutive spreading and interpolation of $F_i^E$. It should be remarked that the correction is not exact as the ansatz in equation (24) corresponds to an approximate solution of the implicit system. Nevertheless, we will show in the following that this approach provides better results with respect to standard explicit direct forcing with a relatively small increase in the computational cost. In the case of unsteady flows, the values of $\kappa_i$ may show oscillations that can be quenched by decreasing the time step or resorting to a relaxation parameter $\gamma\in(0,1]$, such that: $\kappa_i^n=\gamma\,\kappa_i+(1-\gamma)\,\kappa_i^{n-1}$, being $\kappa_i$ the value obtained from equation (27), and $\kappa_i^n$ the actual value of the correction applied at time $t^n$. Moreover, if $F_i^E\left(\mathbf{X}_l\right)\to 0$ then $\mathcal{M}\left[F_i^E\right]\left(\mathbf{X}_l\right)\to 0$, and $\kappa_i\left(\mathbf{X}_l\right)$ might show oscillatory behaviour. In this case, $\kappa_i$ can be set to unity when the magnitude of the denominator in equation (29) is below a small tolerance, typically $10^{-8}$. The procedure to correct the explicit forcing IB is summarised in Algorithm 1.

It should be remarked that the IBM correction (27) differs from others approaches already present in the literature. For instance, the correction proposed in [15] only depends on the shape of the transfer functions, thus only reducing the error owing to the non-reciprocity of the interpolation and spreading operators, while the problem of the overlapping of the supporting domains is not addressed. The improved Lagrangian weight $W_l$, proposed in [20] in the framework of Lattice Boltzmann Method using regularized $\delta$-functions as interpolation kernel, improves the aforementioned method by eliminating the small curvature hypothesis and taking into account the overlapping of the supporting domains. Nevertheless, the correction depends only on the geometry of the surface and on the distribution of the Lagrangian markers, neglecting any fluid information to calibrate the corresponding IB forces. Differently, the coefficients $\kappa_i$ defined here not only accounts for the relative position of the markers and the errors introduced by the spreading operator, but also for the local force $F_i^E$ applied on each Lagrangian marker. As an example, in the case of an unsteady flow around a fixed body, the method proposed in [20,21] provides a steady IBM correction, whereas $\kappa_i$ adapts dynamically through the term $F_i^E$ in equation (27).

Furthermore, the work of [22] introduces a global correction factor (called $Z_i$) for all the IB forces acting on the wet surface. Instead, as already discussed, the correction proposed in this work is local and it accounts simultaneously for the error of the spreading operator and the one given by the overlap of the supporting domains, on every Lagrangian marker.

Notice that even if this correction has been derived within the framework of MLS-IBM, equation (29) is general and can be applied to every Lagrangian method. For the remainder of the paper, we will refer to the novel method as MLS-$\kappa$, and for it we will fix $\varepsilon=0.5$ in equation (9) as detailed in Section 4.1.

## 4. Numerical results

### 4.1. Flow around a fixed sphere

To check the performance and the convergence order of MLS-$\kappa$, we initially consider a fixed sphere of diameter $D$ immersed in a uniform flow $U$, as depicted in Fig. 6a. We define the $L_1$ norm of the error on the interpolated normal and tangential velocity on the Lagrangian markers as follows:

$$\|u_n\|_1=\sum_{l=1}^{N_t}\left|\mathcal{I}\left[u_n\right]\left(\mathbf{X}_l\right)-U_n^b\left(\mathbf{X}_l\right)\right|A_l, \tag{30}$$

$$\|u_t\|_1=\sum_{l=1}^{N_t}\sqrt{\sum_{\alpha=1}^{2}\left[\mathcal{I}\left[u_{t_\alpha}\right]\left(\mathbf{X}_l\right)-U_{t_\alpha}^b\left(\mathbf{X}_l\right)\right]^2}\,A_l, \tag{31}$$

being $u_n$, $U_n^b$ the normal component of the fluid and body velocity and $u_{t_\alpha}$, $U_{t_\alpha}^b$ their tangential components. $\mathbf{X}_l$ indicates the position of the $l$-th Lagrangian marker, which coincides with the centre of the $l$-th triangle of the IB grid having area $A_l$.

The computational domain is a box of dimension $4D\times4D\times8D$, and the sphere is placed two diameters from the flow inlet. The Reynolds number based on the velocity of the free stream $U$ and the diameter of the sphere $D$ is set equal to 100. The Eulerian grid has $N_3=N$ points in the $x_3$ direction and $N_1=N_2=N/2$ points in the $x_1$ and $x_2$ directions, respectively, resulting in uniform spacing $\Delta x$, and $h/\Delta x=0.7$. Since the Reynolds number is below the critical one for the sphere, after a short transient the flow will reach a steady state. The steady state velocity field obtained using MLS-$\kappa$ is depicted in Fig. 6b.

The spatial order of convergence of MLS-$\kappa$ is investigated gradually increasing the number of Eulerian grid points from $N=200$ to

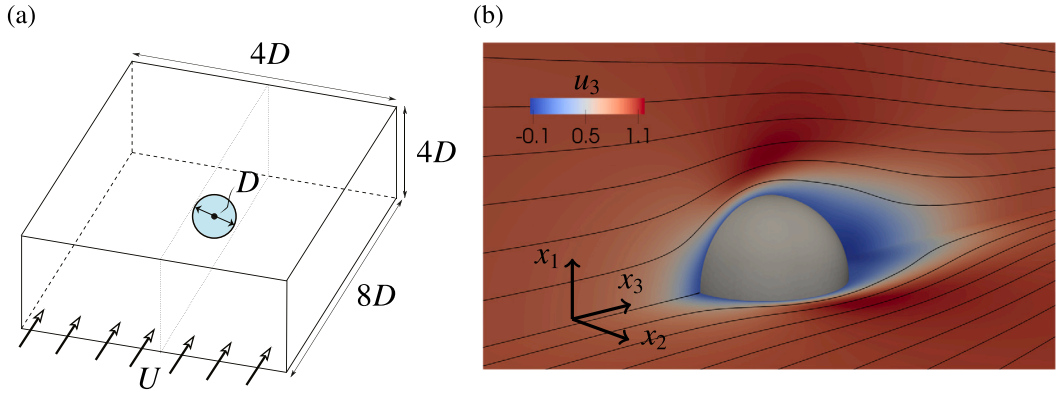



**Fig. 6.** a) Sketch of the computational domain of the flow around the sphere; b) Streamwise component $u_3$ and velocity streamlines of the flow around the sphere at the steady state obtained with MLS-$\kappa$.

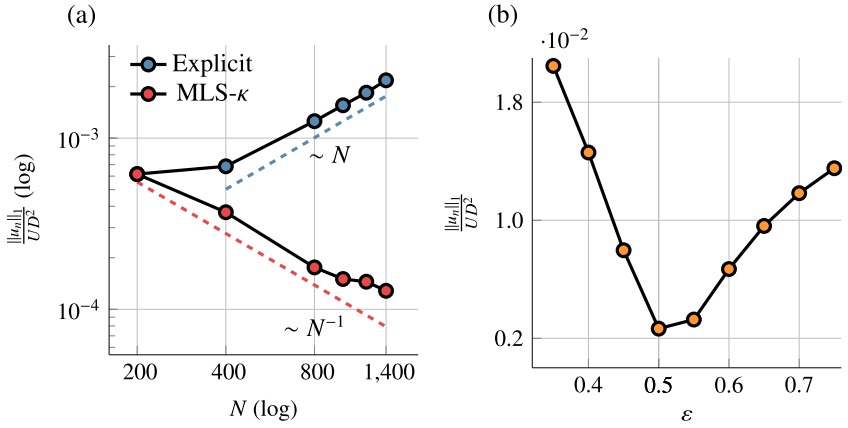



**Fig. 7.** a) Variation of the transpiration error $\|u_n\|_1$ with the number of Eulerian grid nodes $N$ for the explicit method and MLS-$\kappa$. The red dashed line (- - -) has slope -1 while the blue (- - -) has slope 1; b) Variation of $\|u_n\|_1$ with $\varepsilon$ for the flow around the sphere at the steady state obtained with MLS-$\kappa$. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

$N = 1400$, keeping the time step fixed at $5 \times 10^{-5} D/U$. In Fig. 7a we report the variation of $\|u_n\|_1$ with $N$ for the explicit method and MLS-$\kappa$ when the steady state is reached. Interestingly, if $N$ is increased, the boundary velocity error obtained with the explicit method increases, showing that the explicit method diverges linearly, confirming the scaling $\|u_n\|_1 / U\, D^2 \sim N$ proposed by [15]. On the contrary, MLS-$\kappa$ shows a decreasing behaviour with $N$, therefore it is convergent with respect to the spatial grid. In particular, $\|u_n\|_1$ decreases almost linearly with the number of grid points. For the rest of this section we fix $\Delta x = 0.01\ D$ and $\Delta t = 5 \times 10^{-5} D/U$.

Next, we numerically determine the optimal $\varepsilon$ for the weights defined in equation (9), evaluating $\|u_n\|_1$ at the steady state for different values of $\varepsilon$ with MLS-$\kappa$. The results are shown in Fig. 7b: $\|u_n\|_1$ decreases monotonically with $\varepsilon$ up to $\varepsilon \simeq 0.5$, where a minimum is reached, whereas further increments in $\varepsilon$ lead to a growth in $\|u_n\|_1$. Given that, for MLS-$\kappa$ we fix $\varepsilon = 0.5$.

In Fig. 8 the temporal evolution of $\|u_n\|_1$ and $\|u_t\|_1$ from the initial time until the steady regime, obtained with explicit direct forcing, the iterative method, and MLS-$\kappa$ is shown.

The figures show that explicit direct forcing provides the largest error, which is reduced by resorting to the iterative method. Remarkably, the lowest error on both $\|u_n\|_1$ and $\|u_t\|_1$ are achieved by MLS-$\kappa$ (without iterations), thus reducing by one order of magnitude the boundary velocity error with respect to explicit direct forcing, also outperforming the iterative approach. This result is of particular interest because MLS-$\kappa$ is computationally cheaper than the iterative method, as the calculation of $\kappa_i(\mathbf{X}_l)$ does not require to update $u_i^*$, as done in the iterative approach. We also introduced iterations in the MLS-$\kappa$ method with the same strategy as explicit direct forcing, with a new $\kappa_i(\mathbf{X}_l)$ applied to the IB force in each iteration. Unlike what is observed with direct forcing, for MLS-$\kappa$ the introduction of iterations does not significantly improve accuracy. This behaviour is related to the fact that, with a single iteration, MLS-$\kappa$ already yields a sufficiently accurate approximation of the implicit system (21), and the subsequent iterations are thus unable to further reduce the error, owing to the poor conditioning of the system matrix. On the other hand, the explicit direct forcing corresponds to the first step of an iterative CG method, which benefits of further CG iterations, before saturating when the residual error become small, still owing to the poor conditioning of the system. Interestingly, MLS-$\kappa$ is capable of improving not only the normal component of velocity error but also the tangential error with respect to the explicit direct forcing method.

In the same figures, we also report the normal and tangential velocity error evaluated for the non-solenoidal velocity $u_i^*$, which is obtained from equation (3), before the solenoidal projection (4) is enforced. The curves are superimposed on those of the final velocity, showing that in this case the projection error is negligible. However, the projection error may become significant if the pressure gradient across the surface is large, and in such situations one can resort to explicit treatment of the pressure boundary condition as in [23].

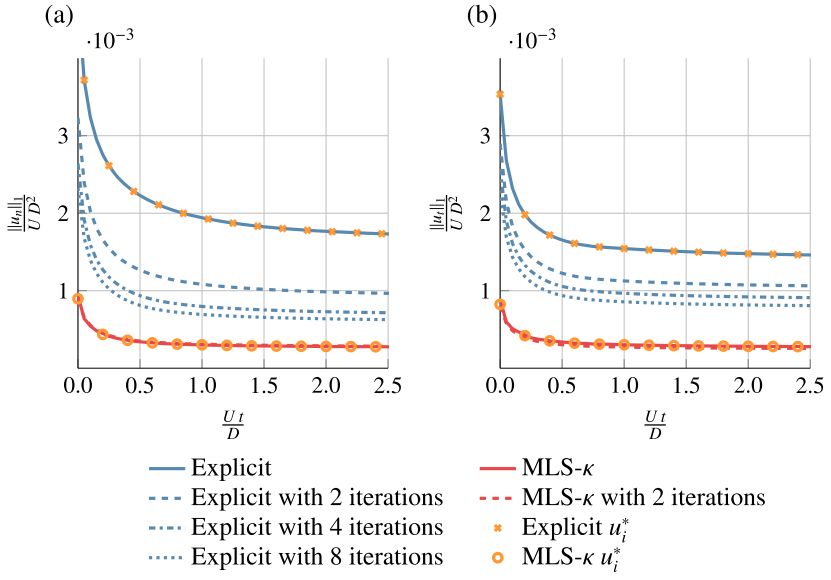



**Fig. 8.** (a) Normal and (b) tangential velocity error for the flow around a sphere at $Re = 100$ for the MLS-$\kappa$ compared against the standard and iterative explicit MLS. The orange symbols correspond to the slip error on the provisional velocity $u_i^*$.

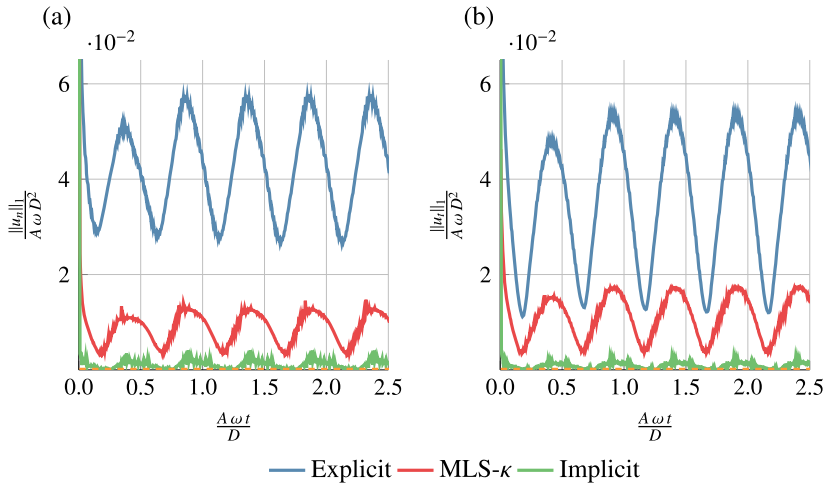



**Fig. 9.** (a) Normal and (b) tangential velocity error for the horizontally oscillating sphere at $Re = 30$. The line - - - indicates the normal and tangential velocity error on the boundary before applying the solenoidal projection (4), obtained using the implicit method with $h/\Delta x = 1.18$.

### 4.2. Horizontally oscillating sphere

We now consider again the case of the horizontally oscillating sphere described in Section 3.2. In Fig. 9, $\|u_n\|_1$ and $\|u_t\|_1$ are shown for the same value of the Reynolds number in the case of explicit direct forcing and MLS-$\kappa$.

After a short transient, the curves show an oscillatory behaviour due to the periodic nature of the flow. Similarly to the case of the fixed sphere, MLS-$\kappa$ reduces by one order of magnitude the error $\|u_n\|_1$ with respect to the explicit method, and greatly reduces the tangential component $\|u_t\|_1$. Therefore, also in this case, MLS-$\kappa$ outperforms the explicit method in both the normal and the tangential directions.

For the sake of completeness, on the same figures we also report the residues obtained with the implicit method. As expected, this approach provides the lowest mean and peak residues, but, as described in Section 3, it cannot be used in problems that require a larger number of Eulerian grid points. However, as mentioned above, the fluid transpiration at the wall is not zero at machine precision due to the projection step equation (4), which is performed after the IB forcing is applied. For this reason, we also reported in the figure the boundary error of $u_i^*$ obtained for the implicit method, which is zero to machine precision. Nevertheless, it is clear from the figure that the projection effect on the boundary velocity is small.

Let us now consider the same flow at $Re = 100$, with the Eulerian grid size reduced to $\Delta x = 0.02\,D$. For this value of the Reynolds number, the velocity errors on the wet surface given by explicit direct forcing, MLS-$\kappa$ and the iterative method are shown in Fig. 10. As before, MLS-$\kappa$ performs better than explicit direct forcing and, as in the fixed sphere case, it also outperforms the iterative approach.

Finally, in Table 2 we compare the variation of the wall clock time $\langle \mathrm{WCT} \rangle$ and the normal boundary error $\langle \|u_n\|_1 \rangle$ of the different methods averaged over one period of oscillation of the sphere. In the first column, we report the Reynolds number used to perform the simulations along with the corresponding grid size (that is uniform in the domain). In the following columns the additional wall clock time $\Delta\langle \mathrm{WCT} \rangle$ of MLS-$\kappa$ and of the iterative method (with 2 and 8 iterations) with respect to the one of the baseline explicit forcing $\langle \mathrm{WCT}^E \rangle$ is reported. It can be observed that MLS-$\kappa$ only slightly increases the computational time by roughly 2%, where the weak increasing trend with the Reynolds number is due to the different grid sizes. The computational cost, instead, rises for the 2 iterations (of about 4%) and for the 8 iterations (of about 26%) methods.

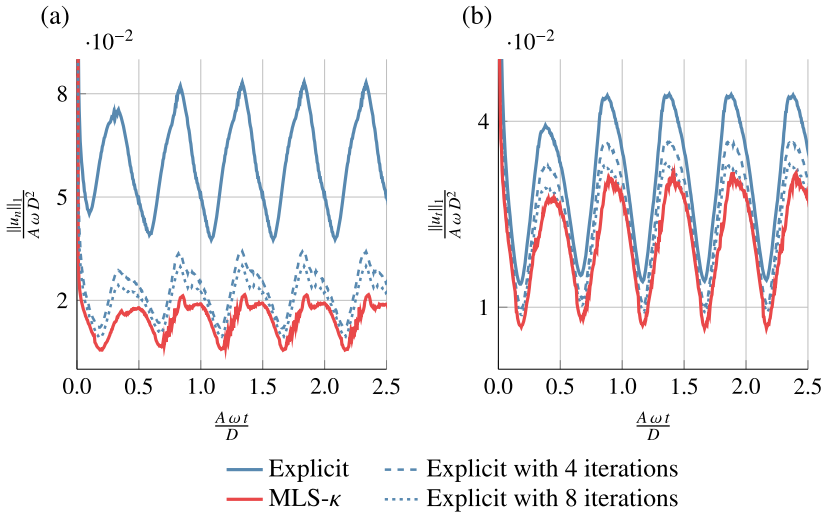



**Fig. 10.** (a) Normal and (b) tangential velocity error for the horizontally oscillating sphere at $Re = 100$.

In particular, even if only 2 iterations are performed, for all the Reynolds number values examined, the iterative method requires a larger WCT with respect to MLS-$\kappa$. Furthermore, the corresponding relative error on the transpiration velocity $\Delta\langle\|u_n\|_1\rangle$ normalised with the one of the explicit direct forcing $\langle\|u_n\|_1^E\rangle$ is reported in the same table. Despite the lower computational burden, MLS-$\kappa$ increases the numerical precision of about 76%, thus outperforming all the other methods also in enforcing the no slip condition on the IBs.

### 4.3. An FSI case: Vortex Induced Vibration

We next examine the fluid-structure interaction case of a steady flow $U$ moving past a homogeneous sphere with diameter $D$ and mass $m$, which is restricted to oscillate in the $x_2$ direction under the effect of a linear spring with stiffness $k$, as shown in Fig. 11.

In non-dimensional terms, the vertical motion of the sphere is driven by the following equation:

$$m^* \ddot{y} + \left(\frac{2\pi}{U^*}\right)^2 m^* y = \frac{6}{\pi} F_y, \tag{32}$$

being $y$ the vertical displacement of the centre of the sphere with respect to some fixed point. $F_y$ is the $x_2$-component of the hydrodynamical forces acting on the sphere, $m^*$ is the body-to-fluid density ratio and $U^* = U/f_n\,D$ is the *reduced velocity*, where $f_n = \sqrt{k/m}/2\pi$ is the natural frequency of the mass-spring system. In the following, we fix $Re = 300$, $U^* = 7$ and $m^* = 2.865$.

The computational domain is a box of dimension $5\,D \times 5\,D \times 10\,D$ and the uniform Eulerian grid has 500 points in the $x_1$ and $x_2$ directions and 1000 in the $x_3$ direction, resulting in a constant grid spacing $\Delta x = 10^{-2}\,D$. The centre of the sphere is placed at $2D$ from the inlet. The time step is kept fixed at $\Delta t = 5\times10^{-4}\,D/U$. The hydrodynamic forces acting on the sphere are evaluated following the method described in [2].

Under these parameter conditions, the flow breaks the cylindrical symmetry of the system, leading to a periodic shedding of vortices. Due to the vortices, an alternating lift force emerges, causing the sphere to start vibrating, ultimately reaching a limit cycle for $Ut/D \gtrsim 150$, where the oscillation maintains a sinusoidal motion with frequency $f$ and amplitude $A$. In this setting, the sphere oscillation frequency is synchronized with the natural frequency of the system, and it is close to the static body vortex shedding frequency. This locked-in regime with large amplitude vortex-induced vibration is called *branch A*, see [29].

The time evolution of the position and velocity of the centre of the sphere obtained with MLS-$\kappa$ is shown in Fig. 12, together with the results extracted from [29]. The figure shows that MLS-$\kappa$ is able to reproduce the disruption of the flow symmetry and the reaching of a periodic

**Table 2**
Wall clock time (WCT) and corresponding velocity transpiration $\|u_n\|_1$ averaged over one period of oscillation of the horizontally oscillating sphere for the MLS-$\kappa$ compared against the standard and the iterative explicit MLS methods. For all $Re$, the quantities are normalized with respect to the explicit MLS results.

| | | MLS-$\kappa$ | | Explicit + 2 iterations | | Explicit + 8 iterations | |
|---|---|---|---|---|---|---|---|
| Re | $\Delta x/D$ | $\frac{\Delta\langle\text{WCT}\rangle}{\langle\text{WCT}^E\rangle}$ | $\frac{\Delta\langle\|u_n\|_1\rangle}{\langle\|u_n\|_1^E\rangle}$ | $\frac{\Delta\langle\text{WCT}\rangle}{\langle\text{WCT}^E\rangle}$ | $\frac{\Delta\langle\|u_n\|_1\rangle}{\langle\|u_n\|_1^E\rangle}$ | $\frac{\Delta\langle\text{WCT}\rangle}{\langle\text{WCT}^E\rangle}$ | $\frac{\Delta\langle\|u_n\|_1\rangle}{\langle\|u_n\|_1^E\rangle}$ |
| 30 | 0.04 | +1% | -79% | +4% | -29% | +25% | -51% |
| 100 | 0.02 | +2% | -75% | +3% | -53% | +28% | -66% |
| 300 | 0.01 | +2% | -78% | +4% | -47% | +24% | -71% |

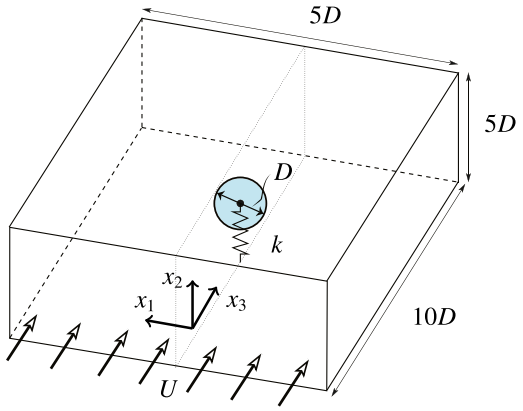



**Fig. 11.** Sketch of the spring-mass system subjected to uniform flow and depiction of computational domain.

orbit. In particular, the reduced frequency $f^* = f/f_n$ and amplitude $A^* = \sqrt{2}\,y_{rms}/D$ evaluated with MLS-$\kappa$ are 0.93 and 0.35 respectively, while the reference paper reports 0.95 and 0.35. Therefore, the limit cycle of the system is accurately described in terms of the amplitude and frequency of the oscillation. On the other hand, the difference in the transient phase can be ascribed to the destabilisation mechanism of the sphere that can reorient the wake in any transversal direction, thus affecting the initial flow regime (see Fig. 13). Although in this flow case the explicit MLS method provides low slip and transpiration error on the surface, employment of MLS-$\kappa$ allows to reduce the normal mean transpiration error over a period by $\sim 33\%$, while the tangential mean transpiration error over a period is reduced by $\sim 5\%$. Therefore, we con-

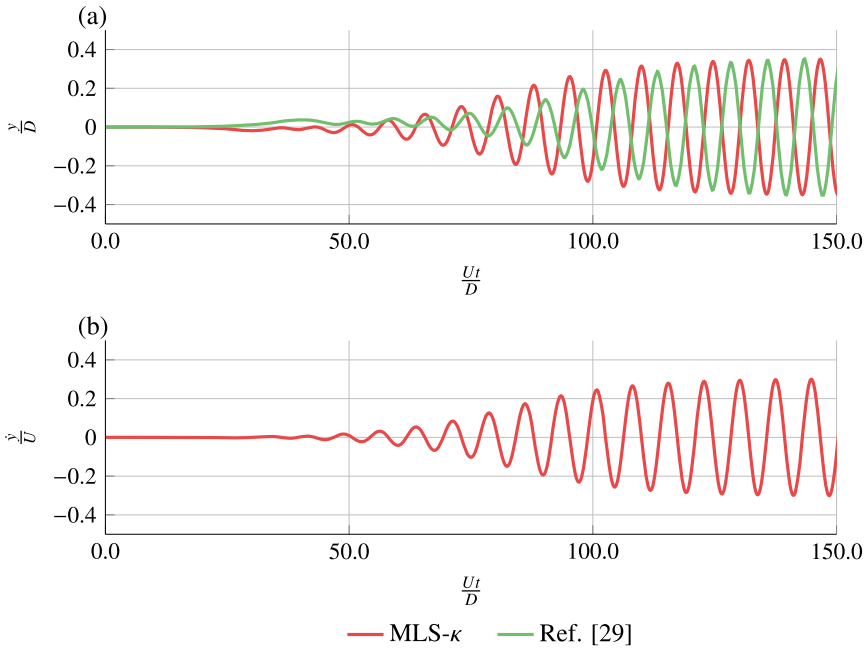



**Fig. 12.** (a) Position $y$ and (b) velocity $\dot{y}$ of the centre of the sphere. The reference position is extracted form [29].

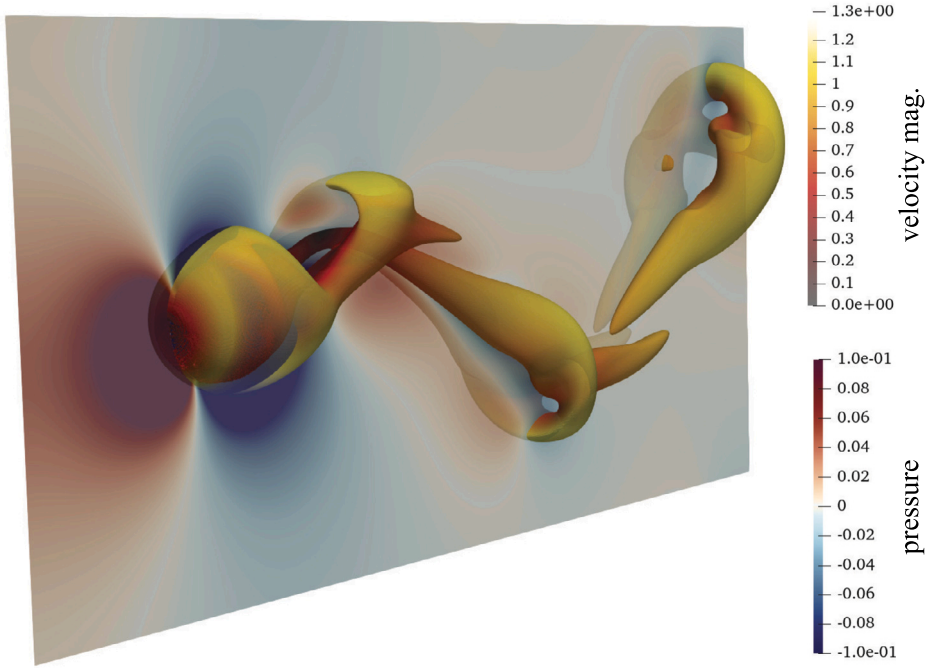



**Fig. 13.** Snapshot of the vortical structures (Q-criterion) behind the vortex-induced oscillating sphere. The isosurface is coloured with the velocity magnitude, whereas the vertical plane displays pressure.

clude that even for this FSI case MLS-$\kappa$ improves the imposition of the no slip boundary condition, and the resulting flow (and structure) dynamics is in good agreement with the results available in the literature.

### 4.4. A bio-inspired case: idealised aorta

The last test case that we analyse is the pulsatile flow within a simplified and rigid aorta, the geometry of which is shown in Fig. 14a.

The inlet diameter of the aorta is $D$, and before the inlet we position a rigid pipe with the same diameter $D$ and length $3\,D$ to allow the flow to correctly develop, indicated by the dashed area in Fig. 14a. The inflow profile to the pipe $\mathbf{u}^{in} = u_3^{in}\,\mathbf{e}_3$ is such that:

$$u_3^{in}(t,r) = U(t)\,\frac{\tanh\left[\alpha\,(1-2r/D)\right]}{\tanh\alpha}, \tag{33}$$

with $\alpha = 20$, $r = \sqrt{x_1^2 + x_2^2}$ and $U(t)$, depicted in Fig. 14b, is a temporal profile built to mimic the systolic and diastolic phases of the cardiac cycle and its maximum value is $U^m$. The Reynolds number based on $D$ and $U^m$ is 3000. The computational domain is a box of dimension $5.5\,D \times 3\,D \times 8.5\,D$ and the uniform Eulerian grid has 1100, 600 and 1700 points in the $x_1$, $x_2$ and $x_3$ directions, respectively, for a total of 1.122 billion of grid points, resulting in a constant grid spacing $\Delta x = 5 \times 10^{-3}\,D$ in all directions. The time step is kept fixed at

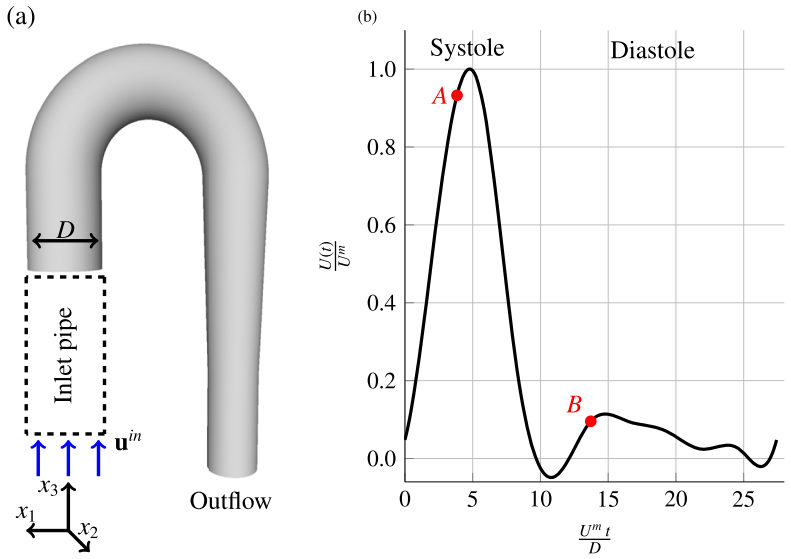



**Fig. 14.** (a) Geometrical configuration of the flow within the simplified aorta. The dashed line is a sketch of the inlet pipe required to let the flow develop. (b) Temporal profile $U(t)$ of the inflow $\mathbf{u}^{in}$. The corresponding velocity field at the time instants depicted by the dots (•) are reported in Fig. 15.

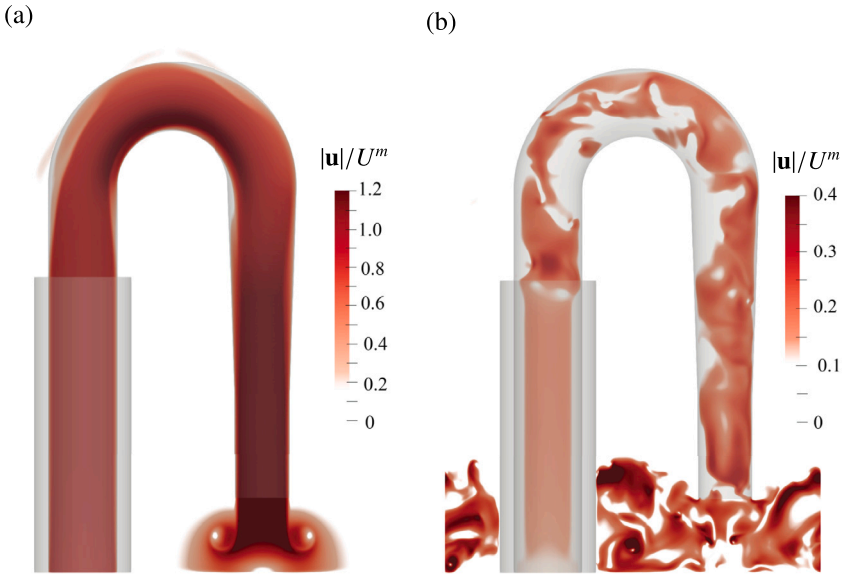



**Fig. 15.** Magnitude of the flow velocity at the time instant indicated by (a) point $A$ and (b) point $B$ in Fig. 15a.

$\Delta t = 5 \times 10^{-5}\, D/U^m$. The body surface is discretised using $2.7 \times 10^6$ Lagrangian markers to keep $h/\Delta x \leq 0.7$.

This flow configuration is particularly critical for the IBM since the flow is forced to perform a 180° turn and a strong pressure jump develops across the boundary. In Fig. 15 the magnitude $|\mathbf{u}| = \sqrt{u_1^2 + u_2^2 + u_3^2}$ is shown for points $A$ and $B$ of Fig. 14b, obtained with MLS-$\kappa$. Near the peak of the systolic phase (point $A$ of Fig. 14b), the flow follows quite uniformly the direction of the aorta, from the inlet to the outlet, with a small region of reduced speed at the apex of the aortic arch. During the diastolic phase (point $B$ of Fig. 14b), the flow manifests recirculation zones and separations throughout the aorta.

In Fig. 16, the velocity errors $\|u_n\|_1$ and $\|u_t\|_1$ are represented over a period. The error behaviour closely resembles that of prior cases. Both the iterative method and MLS-$\kappa$ significantly decrease the transpiration error compared to the explicit method; however, compared to the 8 iterations scenario, MLS-$\kappa$ performs marginally worse for $\|u_n\|_1$ (Fig. 16a). Regarding the error $\|u_t\|_1$ (Fig. 16b), the curves are closely grouped, resulting in comparable slip error values.

In Fig. 17a the flow magnitude $|\mathbf{u}|$ interpolated on the markers for the explicit method and MLS-$\kappa$ is depicted at point $A$. With the explicit method, $|\mathbf{u}|$ varies significantly along the aorta: It reaches its maximum value near the inlet and the aortic arch and then decreases approaching the flow outlet. In contrast, using MLS-$\kappa$, the values of $|\mathbf{u}|$ are not only lower compared to the explicit method but also remain quite constant across the aorta's surface.

Finally, in Fig. 17b is depicted the absolute value of the velocity flux across the surface of the aorta $|\Phi(\mathbf{u})|$, defined as $\Phi(\mathbf{u}) = \sum_{l=1}^{N_t} \mathcal{I}\left[u_n\right](\mathbf{X}_l)\, A_l$, where $A_l$ is the area of the $l$-th triangle of the body mesh. As anticipated, the explicit method yields the highest flux among all methods. Implementing iterations in the scheme mitigates this error; however, for this particular flow configuration, the MLS-$\kappa$ method consistently achieves the lowest flux at all time steps. This finding is particularly interesting because, although the $\|u_n\|_1$ produced by MLS-$\kappa$ is

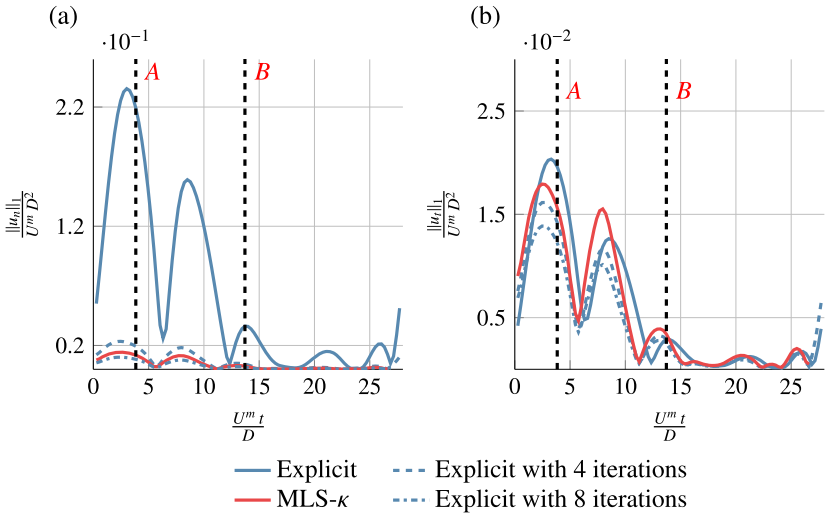



**Fig. 16.** (a) Normal and (b) tangential boundary velocity error for the flow within the idealised aorta at $Re = 3000$. The vertical dashed lines indicate the time instants $A$ and $B$ corresponding to the snapshots reported in Fig. 15.

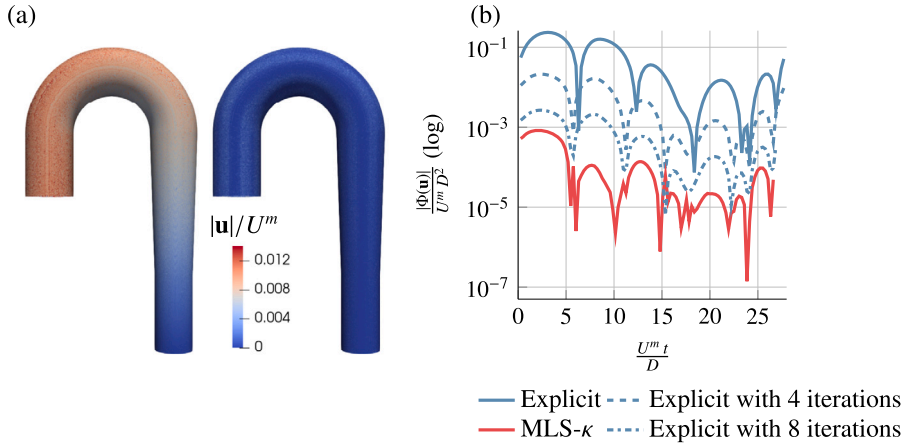



**Fig. 17.** a) Velocity magnitude $|\mathbf{u}|$ on the aorta's surface at point $A$ obtained with the explicit method (left) and MLS-$\kappa$ (right); b) Absolute value of the velocity flux across the surface of the aorta.

slightly higher than the one obtained with 8 iterations, the flux across the surface produced by MLS-$\kappa$ is two orders of magnitude lower than the explicit baseline method and it is also lower with respect the 8 iterations method by approximately one order of magnitude.

## 5. Conclusions

In this work, an explicit and local correction is proposed to the direct explicit forcing IBM-MLS formulation.

The starting point of our analysis is the derivation in the case of IBM-MLS of the implicit forcing method, in which the Lagrangian IB forces required to impose the no slip boundary condition on the wet walls are computed taking into account the contribution of all Lagrangian markers, simultaneously. Following this approach, at each time step, the forces are obtained upon the solution of a linear system of equations for each velocity components (hence, three linear systems in a three-dimensional case). However the system matrix is seen to become ill conditioned if $h/\Delta x < 1$, which is needed to effectively impose the no slip condition. As a consequence, it turns out that the implicit method cannot be used to solve three-dimensional problems requiring fine Eulerian grids, such as high Reynolds number flows or flows around complex geometries.

Then, starting from the implicit IB formulation, we proposed to solve the linear system of equations via an approximate factorisation of the system matrix, that ultimately leads to a correction of the direct forcing explicit method through a coefficient $\kappa_i(\mathbf{X}_l)$ multiplying the IB Lagrangian forcing. The coefficient $\kappa_i(\mathbf{X}_l)$ is determined dynamically at each temporal step for each Lagrangian marker, thus providing a local and instantaneous correction to the non-reciprocity of the interpolation and spreading operator. Therefore, differently from the correction factor $Z_i$ introduced in [22,21], which is defined globally for all the IB forces, $\kappa_i(\mathbf{X}_l)$ acts locally and its value can change from one marker to another. Thus, the proper balance between the Eulerian and the corresponding Lagrangian forcing is guaranteed for each marker. Furthermore, differently from the improved Lagrangian weight $W_l$ introduced [20,21], $\kappa_i(\mathbf{X}_l)$ depends not only on the geometry of the immersed surface, but also on the instantaneous flow dynamics through the explicit IB forcing, thus assuring to correctly balance the force applied on each marker.

The proposed correction is tested and compared against standard IBM-MLS (also in its iterative fashion) in a series of benchmark flows, including fixed and moving rigid bodies. The novel method outperforms in terms of numerical precision the explicit direct forcing MLS approach in all test cases considered, greatly reducing the boundary error. Due to its explicit nature, it does not significantly increase the computational cost with respect to the standard explicit MLS and it is shown to be computationally cheaper than the iterative MLS method.

The correction proposed here has been derived in the framework of IBM-MLS, but it can be easily extended to the continuous forcing method

or to IBM/Lattice-Boltzmann methods. Given the formulation of $\kappa_i\left(\mathbf{X}_l\right)$ (see equation (29)), such correction can be promptly implemented in existing numerical codes, with minimal modification to the IBM routines.

## CRediT authorship contribution statement

**Giovanni Vagnoli:** Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Martino Andrea Scarpolini:** Writing – review & editing, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation. **Roberto Verzicco:** Writing – review & editing, Supervision, Methodology, Investigation. **Francesco Viola:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation program (Grant No. 101039657, CARDIOTRIALS to F.V.). CINECA is gratefully acknowledged for the access to high-performance computing resources under the computational grant IsB23-CaRdiopT.

## Data availability

Data will be made available on request.

## References

[1] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (2) (1972).
[2] M.D. de Tullio, G. Pascazio, A moving-least-squares immersed boundary method for simulating the fluid–structure interaction of elastic bodies with arbitrary thickness, J. Comput. Phys. 325 (2016).
[3] F. Viola, G. Del Corso, R. De Paulis, R. Verzicco, GPU accelerated digital twins of the human heart open new routes for cardiovascular research, Sci. Rep. 13 (1) (2023).
[4] S. Pirozzoli, P. Orlandi, M. Bernardini, The fluid dynamics of rolling wheels at low Reynolds number, J. Fluid Mech. 706 (2012).
[5] P. Windes, X. Fan, M. Bender, D.K. Tafti, R. Müller, A computational investigation of lift generation and power expenditure of Pratt's roundleaf bat (Hipposideros pratti) in forward flight, PLoS ONE 13 (11) (2018).
[6] M. Bozkurttas, R. Mittal, H. Dong, G.V. Lauder, P. Madden, Low-dimensional models and performance scaling of a highly deformable fish pectoral fin, J. Fluid Mech. 631 (2009).
[7] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, J. Comput. Phys. 59 (2) (1985).
[8] X. Zhu, E. Phillips, V. Spandan, J. Donners, G. Ruetsch, J. Romero, R. Ostilla-Mónico, Y. Yang, D. Lohse, R. Verzicco, M. Fatica, R.J.A.M. Stevens, AFiD-GPU: a versatile Navier–Stokes solver for wall-bounded turbulent flows on GPU clusters, Comput. Phys. Commun. 229 (2018).
[9] R. Verzicco, Immersed boundary methods: historical perspective and future outlook, Annu. Rev. Fluid Mech. 55 (1) (2023).
[10] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (1) (2005).
[11] B.E. Griffith, N.A. Patankar, Immersed methods for fluid–structure interaction, Annu. Rev. Fluid Mech. 52 (1) (2020).
[12] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2) (2005).
[13] M. Vanella, E. Balaras, A moving-least-squares reconstruction for embedded-boundary formulations, J. Comput. Phys. 228 (18) (2009).
[14] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002).
[15] S. Gsell, J. Favier, Direct-forcing immersed-boundary method: a simple correction preventing boundary slip error, J. Comput. Phys. 435 (2021).
[16] S.-W. Su, M.-C. Lai, C.-A. Lin, An immersed boundary technique for simulating complex flows with rigid boundary, Comput. Fluids 36 (2) (2007).
[17] K. Taira, T. Colonius, The immersed boundary method: a projection approach, J. Comput. Phys. 225 (2) (2007).
[18] A. Pal Singh Bhalla, B.E. Griffith, N.A. Patankar, A. Donev, A minimally-resolved immersed boundary model for reaction-diffusion problems, J. Chem. Phys. 139 (21) (2013).
[19] T. Kempe, J. Fröhlich, An improved immersed boundary method with direct forcing for the simulation of particle laden flows, J. Comput. Phys. 231 (9) (2012).
[20] I. Cheylan, T. Fringand, J. Jacob, J. Favier, Analysis of the immersed boundary method for turbulent fluid-structure interaction with Lattice Boltzmann method, J. Comput. Phys. 492 (2023).
[21] E. Farah, A. Ouahsine, P.G. Verdin, An improved implicit direct-forcing immersed boundary method (df-ibm) around arbitrarily moving rigid structures, Phys. Fluids 36 (10) (2024).
[22] W. Chen, S. Zou, Q. Cai, Y. Yang, An explicit and non-iterative moving-least-squares immersed-boundary method with low boundary velocity error, J. Comput. Phys. 474 (2023).
[23] I.N. Yildiran, N. Beratlis, F. Capuano, Y.-H. Loke, K. Squires, E. Balaras, Pressure boundary conditions for immersed-boundary methods, J. Comput. Phys. 510 (2024).
[24] E.P. van der Poel, R. Ostilla-Mónico, J. Donners, R. Verzicco, A pencil distributed finite difference code for strongly turbulent wall-bounded flows, Comput. Fluids 116 (2015).
[25] G.-R. Liu, Y.T. Gu, An Introduction to Meshfree Methods and Their Programming, Springer, Dordrecht, 2010.
[26] S. Gsell, U. D'Ortona, J. Favier, Explicit and viscosity-independent immersed-boundary scheme for the lattice Boltzmann method, Phys. Rev. E 100 (3) (2019).
[27] E.A. 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).
[28] X. Zhao, Z. Chen, L. Yang, N. Liu, C. Shu, Efficient boundary condition-enforced immersed boundary method for incompressible flows with moving boundaries, J. Comput. Phys. 441 (2021).
[29] M.M. Rajamuni, M.C. Thompson, K. Hourigan, Transverse flow-induced vibrations of a sphere, J. Fluid Mech. 837 (2018).