



An efficient method for particle-resolved simulations of neutrally buoyant spheres

Manuel García-Villalba^{a,*}, Blanca Fuentes^b, Jan Dušek^c, Manuel Moriche^d, Markus Uhlmann^d

^a Institute of Fluid Mechanics and Heat Transfer, TU Wien, 1060 Vienna, Austria

^b Universidad Carlos III de Madrid, Leganés 28911, Spain

^c Université de Strasbourg, Fluid Mechanics Department, Institut ICube, France

^d Institute for Hydromechanics, Karlsruhe Institute of Technology, Karlsruhe 76131, Germany

ARTICLE INFO

Keywords:

Particle-resolved simulation
Immersed boundary method
Particle-laden flows
Neutrally buoyant particles

ABSTRACT

We present a simple modification of the direct-forcing immersed boundary method (IBM) proposed by Uhlmann (2005) in order to enable it to be applied to particulate flows with solid-to-fluid density ratios around unity. The main difference with respect to the original formulation lies in the particle velocity update which is performed directly after the preliminary velocity field has been computed in the absence of any IBM volume forcing term. In addition, we apply the forcing term to the entire space occupied by the immersed solid object (instead of to the vicinity of its interface only). The present approach requires the evaluation of integrals of the velocity field over the volume occupied by the solid particle, which are evaluated efficiently as sums over the respective quantities available at particle-attached force points. The resulting method can be used seamlessly for density ratios down to $\rho_p/\rho_f > 0.5$. The new formulation has been validated using three configurations: (i) lateral migration of a neutrally buoyant circular particle in two-dimensional Couette flow; (ii) the release from rest of a neutrally buoyant sphere in a free stream; (iii) the release of a particle in a free stream after an initial phase in which it is translationally fixed with an imposed angular velocity. In all three test cases the present IBM formulation yields a very good agreement with the available reference data. Thus, the proposed approach is a cost-efficient and accurate modification of the original method which allows for the simulation of fluid systems involving density-matched solid particles.

1. Introduction

Over the last 20 years the numerical simulation of particle-laden flows has reached a certain level of maturity. There is now a large number of particle-resolved techniques available in the literature [1–3]. Among them, the immersed boundary method (IBM) [4,5] is a popular approach mainly due to the fact that the underlying computational grid does not need to be adapted over time, leading to relatively simple algorithms. The IBM was first developed by Peskin [6,7] in the context of heart flow simulations and since then multiple variations have been published and are in use today for a variety of problems [8–12]. One method that has found widespread use is the direct-forcing IBM developed by Uhlmann [13]. As recently pointed out by Zhou and Balachandar [14], “[it] is very popular because it is easy to implement and flexible to simulate various problems”. Over the years the method of Uhlmann has been modified and improved by several authors [15–18].

Despite its success, the method of Uhlmann has a number of limitations. One of them arises from the weak coupling between the

Navier–Stokes equations for the fluid phase and the Newton–Euler equations for the particle motion, leading to a lower limit for the density ratio between solid and fluid phase of about 1.2. For lighter particles the computations are unstable which is unsurprising due to the singularity of the formulation for the neutrally buoyant case. In order to overcome this limitation there have been several proposals in the literature. Kim and Choi [19] presented a method in which the particle equations are integrated with an implicit method and then showed how the evaluation of the hydrodynamic force at the new (unknown) time level can be avoided by using Lagrangian quantities. This results in a non-iterative and non-singular formulation for the neutrally buoyant case. However, to the best of our knowledge, this method has not been applied yet to the motion of light particles. A different approach was proposed by Schwarz et al. [20] by adding a stabilizing term to both sides of the Newton–Euler equation. This term is denoted *virtual mass force* and allows for small density ratios to be computed without stability issues. A modification of Schwarz et al.’s method has been

* Corresponding author.

E-mail address: manuel.garcia-villalba@tuwien.ac.at (M. García-Villalba).

reported by Tavanashad and Subramaniam [21] for the case of larger volume fractions.

A further improvement of Schwarz et al.'s method was reported by Tschisgale et al. [22]. These authors restrict the rigid-body assumption to a finite-thickness interface layer and modify the coupling between fluid equations and particle equations, leading to a non-iterative semi-implicit coupling. The resulting method is non-singular and remains valid even in the limit of vanishing particle mass. The authors however report a weak oscillation of the particle velocity when the particle moves free of external forces, a feature that they attribute to the numerical integration of the virtual mass force. In the present work, using some of the ideas of Tschisgale et al. [22], we propose and validate a new modification of the original method of Uhlmann [13] for the computation of neutrally buoyant spheres. As will be shown below, the new method is free of spurious oscillations.

In order to validate the present technique, a new flow configuration is proposed, in which the response of a neutrally buoyant particle involves non-trivial angular motion. In this context, we have generated new high-fidelity reference data with the aid of a spectral-element method on a particle-attached conforming grid [23].

The manuscript is organized as follows. First, the formulation of the problem, the governing equations and a discussion of the fluid-particle coupling are provided in Section 2. This is followed by the numerical details and a complete description of the proposed method in Section 3. We have performed three test cases for the validation of the methodology. The first is the migration of a neutrally buoyant disk in 2D Couette flow (Section 4.1). The second is a neutrally buoyant sphere released from rest in a free stream (Section 4.2). The third and final case is an initially rotating sphere released in a free-stream (Section 4.3). The manuscript ends with some final remarks and conclusions.

2. Formulation of the problem

The physical problem under consideration consists of the interaction of rigid particles of constant density, ρ_p , with a surrounding viscous fluid of constant kinematic viscosity, ν , and constant density, ρ_f , in the incompressible flow limit. For simplicity, we limit the discussion to a single particle, but the extension to several particles is trivial and has been discussed elsewhere [13]. In this work we only consider a circular particle in 2D or a sphere in 3D. The extension of the method to particles of arbitrary shape does not present fundamental problems but only the added complexity of tracking the orientation of the particles. This has already been illustrated for related methods in the literature [24–26].

2.1. Fluid phase equations

The governing equations for the fluid phase are the Navier–Stokes of the incompressible flow for a Newtonian fluid

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

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

where $\mathbf{u} = (u, v, w)$ is the fluid velocity, p is the pressure divided by the fluid density, and \mathbf{f} is a specific volume force term. As in the original method, the equations are enforced throughout the entire domain, Ω , comprising the fluid domain, Ω_f and the space occupied by the particle, S . These equations need to be supplemented with initial and suitable boundary conditions at the outer boundary. The volume force \mathbf{f} is formulated in order to enforce the no-slip boundary condition on the surface of the particle, as discussed in the following sections. In the original method, the forcing was restricted to a layer around the solid–fluid interface. However, recent work has shown that good results are obtained extending the forcing to the interior of the particle [26]. Here, we use the latter approach, as it was found beneficial to treat neutrally buoyant particles by other authors [27].

2.2. Solid phase equations

The motion of the particle is governed by the Newton–Euler equations for the motion of a rigid body under the action of hydrodynamic and gravity forces

$$\rho_p V_c \frac{d\mathbf{u}_p}{dt} = \rho_f \oint_{\partial S} \boldsymbol{\tau} \cdot \mathbf{n} d\sigma + (\rho_p - \rho_f) V_c \mathbf{g}, \quad (3)$$

$$I_p \frac{d\boldsymbol{\omega}_p}{dt} = \rho_f \oint_{\partial S} \mathbf{r} \times (\boldsymbol{\tau} \cdot \mathbf{n}) d\sigma, \quad (4)$$

where V_c is the volume of the particle, $\mathbf{u}_p = (u_p, v_p, w_p)$ the velocity of the center of mass of the particle (located at \mathbf{x}_p), \mathbf{g} the gravitational acceleration, $\boldsymbol{\omega}_p = (\omega_{px}, \omega_{py}, \omega_{pz})$ the angular velocity of the particle, $\mathbf{r} = \mathbf{x} - \mathbf{x}_p$ the position vector of any point in the body with respect to the center of mass, \mathbf{n} the outward-pointing normal unit vector at the particle surface and $\boldsymbol{\tau} = -p\mathbf{I} + \nu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ is the hydrodynamic stress tensor with \mathbf{I} being the identity tensor. In 3D I_p is the moment of inertia with respect to any axis that passes through the center of mass, while in 2D I_p is the moment of inertia with respect to the axis passing through the center of mass and perpendicular to the motion. For convenience, we also introduce the specific moment of inertia $\tilde{I}_p = I_p/\rho_p$. With respect to the hydrodynamic force and torque, following [13] we can write

$$\oint_{\partial S} \boldsymbol{\tau} \cdot \mathbf{n} d\sigma = - \int_S \mathbf{f} d\mathbf{x} + \frac{d}{dt} \int_S \mathbf{u} d\mathbf{x}, \quad (5)$$

$$\oint_{\partial S} \mathbf{r} \times (\boldsymbol{\tau} \cdot \mathbf{n}) d\sigma = - \int_S (\mathbf{r} \times \mathbf{f}) d\mathbf{x} + \frac{d}{dt} \int_S (\mathbf{r} \times \mathbf{u}) d\mathbf{x}. \quad (6)$$

In case of rigid body motion, it can be shown that the following equations hold

$$\frac{d}{dt} \int_S \mathbf{u} d\mathbf{x} = V_c \frac{d\mathbf{u}_p}{dt}, \quad (7)$$

$$\frac{d}{dt} \int_S (\mathbf{r} \times \mathbf{u}) d\mathbf{x} = \frac{I_p}{\rho_p} \frac{d\boldsymbol{\omega}_p}{dt}. \quad (8)$$

Then, we can rewrite Eqs. (3)–(4) as

$$(\rho_p - \rho_f) V_c \frac{d\mathbf{u}_p}{dt} = -\rho_f \int_S \mathbf{f} d\mathbf{x} + (\rho_p - \rho_f) V_c \mathbf{g}, \quad (9)$$

$$\left(1 - \frac{\rho_f}{\rho_p}\right) I_p \frac{d\boldsymbol{\omega}_p}{dt} = -\rho_f \int_S (\mathbf{r} \times \mathbf{f}) d\mathbf{x}. \quad (10)$$

2.3. Fluid–solid coupling

The coupling between the particle motion and the fluid motion is realized via the volume force term \mathbf{f} . We can solve for \mathbf{f} in Eq. (2), and grouping together convective, pressure and viscous terms into the variable \mathbf{rhs} , we may write

$$\mathbf{f} = \frac{\partial \mathbf{u}}{\partial t} - \mathbf{rhs}. \quad (11)$$

Following [22], we integrate Eq. (11) over a time interval $[t_{n-1}, t_n]$

$$\int_{t_{n-1}}^{t_n} \mathbf{f} dt = \int_{t_{n-1}}^{t_n} \left(\frac{\partial \mathbf{u}}{\partial t} - \mathbf{rhs} \right) dt = \mathbf{u}^n - \mathbf{u}^{n-1} - \int_{t_{n-1}}^{t_n} \mathbf{rhs} dt, \quad (12)$$

where \mathbf{u}^n is the fluid velocity at the time instant t_n . Now we link the fluid and particle motions, by imposing that the fluid velocity of any interior point at the time instant t_n coincides with the rigid body velocity

$$\mathbf{u}^n = \mathbf{u}_p + \boldsymbol{\omega}_p \times \mathbf{r} \quad \forall \mathbf{x} \in S. \quad (13)$$

We also introduce a preliminary velocity field obtained without the contribution of the forcing term

$$\tilde{\mathbf{u}} = \mathbf{u}^{n-1} + \int_{t_{n-1}}^{t_n} \mathbf{rhs} dt. \quad (14)$$

Then, we can rewrite Eq. (12) as

$$\int_{t_{n-1}}^{t_n} \mathbf{f} dt = \mathbf{u}^n - \tilde{\mathbf{u}} \quad \forall \mathbf{x} \in S. \quad (15)$$

Note that, a similar analysis has been performed by [22]. A key difference, however, is that these authors only include the forcing term in a layer surrounding the particle boundary in order to impose the no-slip boundary condition. In the present work, we also include the forcing term in the particle interior so that the fluid velocity in the interior of the particle is forced to follow the rigid-body motion.

We can now also integrate the Eqs. (9)–(10) over the time interval $[t_{n-1}, t_n]$

$$(\rho_p - \rho_f)V_c (\mathbf{u}_p^n - \mathbf{u}_p^{n-1}) = -\rho_f \int_S \int_{t_{n-1}}^{t_n} \mathbf{f} dt d\mathbf{x} + (\rho_p - \rho_f)V_c \mathbf{g} \Delta t, \quad (16)$$

$$\left(1 - \frac{\rho_f}{\rho_p}\right) I_p (\boldsymbol{\omega}_p^n - \boldsymbol{\omega}_p^{n-1}) = -\rho_f \int_S \int_{t_{n-1}}^{t_n} (\mathbf{r} \times \mathbf{f}) dt d\mathbf{x}, \quad (17)$$

where $\Delta t = t_n - t_{n-1}$. On the right hand side of Eqs. (16) and (17), the order of integration has been exchanged, as discussed in detail in Appendix A of Tschisgale et al. [22]. Using Eqs. (13) and (15) we can write

$$\int_S \int_{t_{n-1}}^{t_n} \mathbf{f} dt d\mathbf{x} = \int_S (\mathbf{u}_p^n + \boldsymbol{\omega}_p^n \times \mathbf{r} - \tilde{\mathbf{u}}) d\mathbf{x} = V_c \mathbf{u}_p^n - \int_S \tilde{\mathbf{u}} d\mathbf{x}, \quad (18)$$

$$\int_S \int_{t_{n-1}}^{t_n} (\mathbf{r} \times \mathbf{f}) dt d\mathbf{x} = \int_S \left[\mathbf{r} \times (\mathbf{u}_p^n + \boldsymbol{\omega}_p^n \times \mathbf{r} - \tilde{\mathbf{u}}) \right] d\mathbf{x} = \frac{I_p}{\rho_p} \boldsymbol{\omega}_p^n - \int_S \mathbf{r} \times \tilde{\mathbf{u}} d\mathbf{x}. \quad (19)$$

Introducing Eqs. (18)–(19) into Eqs. (16)–(17) and re-arranging terms we obtain

$$\mathbf{u}_p^n = \left(1 - \frac{\rho_f}{\rho_p}\right) \mathbf{u}_p^{n-1} + \frac{1}{V_c} \frac{\rho_f}{\rho_p} \int_S \tilde{\mathbf{u}} d\mathbf{x} + \Delta t \left(1 - \frac{\rho_f}{\rho_p}\right) \mathbf{g}, \quad (20)$$

$$\boldsymbol{\omega}_p^n = \left(1 - \frac{\rho_f}{\rho_p}\right) \boldsymbol{\omega}_p^{n-1} + \frac{1}{I_p} \frac{\rho_f}{\rho_p} \int_S \mathbf{r} \times \tilde{\mathbf{u}} d\mathbf{x}. \quad (21)$$

The formulation (20)–(21) does not present a singularity at $\rho_p/\rho_f = 1$, but it does in the limit $\rho_p/\rho_f \rightarrow 0$. However, the method is numerically unstable for $\rho_p/\rho_f < 0.5$. This is because the coefficient that multiplies \mathbf{u}_p^{n-1} in Eq. (20) becomes larger than 1 in absolute value for $\rho_p/\rho_f < 0.5$, yielding an unstable algorithm (see for example pages 13–14 in [28]). For neutrally buoyant particles, $\rho_p/\rho_f = 1$, expressions (20)–(21) simplify to

$$\mathbf{u}_p^n = \frac{1}{V_c} \int_S \tilde{\mathbf{u}} d\mathbf{x}, \quad (22)$$

$$\boldsymbol{\omega}_p^n = \frac{1}{I_p} \frac{\rho_f}{\rho_p} \int_S \mathbf{r} \times \tilde{\mathbf{u}} d\mathbf{x}, \quad (23)$$

which has already been proposed by [29]. Note that in the neutrally buoyant limit, \mathbf{u}_p^n does not depend explicitly on \mathbf{u}_p^{n-1} , so that the motion of the particle and the motion of the virtual fluid occupying the position of the particle have to be compatible from the initial time. In other words, the initial velocity field assigned in the interior of a particle needs to be compatible with the initial (linear and angular) particle velocity, according to rigid body motion, Eq. (13). This is not the case when $\rho_f \neq \rho_p$.

3. Numerical methodology

The numerical method to solve the governing equations is then very similar to the original method proposed by [13]. In particular we employ the same flow solver, using second-order finite differences on a staggered grid and a semi-implicit 3-stage Runge–Kutta scheme where the linear, viscous terms are treated implicitly and the convective, non-linear terms are treated explicitly.

We also employ separate discretizations for the Eulerian and Lagrangian quantities. We denote Eulerian quantities with lowercase

letters and Lagrangian quantities with uppercase letters. For the Eulerian discretization, we use a fixed, uniform, Cartesian grid g_h of grid spacing h . The location of the grid points is denoted \mathbf{x}_{ijk}^β , where the superscript $\beta = 1, 2, 3$ refers to the staggered grid associated with the velocity component u_β . For the Lagrangian grid, we evenly distribute N_L points throughout the volume occupied by the particle S . The Lagrangian locations are denoted $\mathbf{X}_\ell \in S$ with $1 \leq \ell \leq N_L$. To each point we associate a discrete volume ΔV_ℓ , such that the sum of these volumes equals the total volume of the particle. The transfer of quantities between Lagrangian and Eulerian locations is based on the same regularized delta function δ_h introduced by [7] and defined by [30], as in the original method [13], where additional details can be found.

It is now possible to describe the new proposed algorithm. This is done here in 3D, the reduction to 2D being straightforward. For the k th Runge–Kutta stage, first compute the preliminary fluid velocity $\tilde{\mathbf{u}}$ without accounting for the effect of the immersed boundary, viz.

$$\tilde{\mathbf{u}} = \mathbf{u}^{k-1} + \Delta t (2\alpha_k \nu \nabla^2 \mathbf{u}^{k-1} - 2\alpha_k \nabla p^{k-1} - \gamma_k [(\mathbf{u} \cdot \nabla) \mathbf{u}]^{k-1} - \xi_k [(\mathbf{u} \cdot \nabla) \mathbf{u}]^{k-2}), \quad (24)$$

where the coefficients $\alpha_k, \gamma_k, \xi_k$ ($1 \leq k \leq 3$) are taken from [31]. Next, we transfer the preliminary velocity from the Eulerian to the Lagrangian grid

$$\tilde{U}_\beta(\mathbf{X}_\ell) = \sum_{i,j,k} \tilde{u}_\beta(\mathbf{x}_{ijk}^\beta) \delta_h(\mathbf{x}_{ijk}^\beta - \mathbf{X}_\ell) h^3 \quad \forall \ell; 1 \leq \beta \leq 3. \quad (25)$$

The next step in the original algorithm is the computation of the force volume term using the desired velocity at the Lagrangian points computed from the values of the previous stage $k - 1$. In the present algorithm we instead compute next the new velocity of the center of mass of the particle, \mathbf{u}_p^k , and the new particle angular velocity, $\boldsymbol{\omega}_p^k$, using Eqs. (20)–(21), or for the neutrally buoyant case the simpler Eqs. (22)–(23). In any case, the integrals of the preliminary velocity need to be computed. One of the benefits of using Lagrangian points also in the interior of the particles is that these two integrals can be easily approximated using discrete sums

$$\int_S \tilde{\mathbf{u}} d\mathbf{x} \approx \sum_{\ell=1}^{N_L} \tilde{U}(\mathbf{X}_\ell) \Delta V_\ell, \quad (26)$$

$$\int_S \mathbf{r} \times \tilde{\mathbf{u}} dV \approx \sum_{\ell=1}^{N_L} \mathbf{R}(\mathbf{X}_\ell) \times \tilde{U}(\mathbf{X}_\ell) \Delta V_\ell, \quad (27)$$

where $\mathbf{R}(\mathbf{X}_\ell) = \mathbf{X}_\ell - \mathbf{x}_p$ is the ℓ -th Lagrangian point's position with respect to the particle's center of mass. If the distribution of Lagrangian points is sufficiently homogeneous and uniform the discrete sums result in a second-order approximation to the integrals, which we have verified numerically. For a discussion about how to distribute points uniformly and associate them the corresponding discrete volumes see appendix B2 of [26].

Introducing the above expressions into Eqs. (20)–(21), we obtain

$$\mathbf{u}_p^k = \left(1 - \frac{\rho_f}{\rho_p}\right) \mathbf{u}_p^{k-1} + \frac{1}{V_c} \frac{\rho_f}{\rho_p} \sum_{\ell=1}^{N_L} \tilde{U}(\mathbf{X}_\ell) \Delta V_\ell + 2\alpha_k \Delta t \left(1 - \frac{\rho_f}{\rho_p}\right) \mathbf{g}, \quad (28)$$

$$\boldsymbol{\omega}_p^k = \left(1 - \frac{\rho_f}{\rho_p}\right) \boldsymbol{\omega}_p^{k-1} + \frac{1}{I_p} \frac{\rho_f}{\rho_p} \sum_{\ell=1}^{N_L} [\mathbf{R}(\mathbf{X}_\ell) \times \tilde{U}(\mathbf{X}_\ell)] \Delta V_\ell. \quad (29)$$

In Eq. (28) the coefficient $2\alpha_k$ appears because the derivation of Eq. (20) was done for a generic interval with time step Δt , while the time step between the two Runge–Kutta stages $k - 1$ and k is $2\alpha_k \Delta t$. We compute now the new desired velocity by using the velocity field of the rigid body

$$\mathbf{U}^{(d)}(\mathbf{X}_\ell) = \mathbf{u}_p^k + \boldsymbol{\omega}_p^k \times \mathbf{R}(\mathbf{X}_\ell) \quad \forall \ell. \quad (30)$$

Once the new desired velocity has been computed, the remaining steps of the original algorithm for the fluid phase are unmodified. Therefore,

we have the following sequence of operations:

$$\mathbf{F}(\mathbf{X}_\ell) = \frac{\mathbf{U}^{(d)}(\mathbf{X}_\ell) - \tilde{\mathbf{U}}(\mathbf{X}_\ell)}{\Delta t} \quad \forall \ell, \quad (31)$$

$$f_\beta^k(\mathbf{x}_{ijk}^\beta) = \sum_{\ell=1}^{N_\ell} F_\beta(\mathbf{X}_\ell) \delta_h(\mathbf{x}_{ijk}^\beta - \mathbf{X}_\ell) \Delta V_\ell \quad \forall i, j, k; 1 \leq \beta \leq 3, \quad (32)$$

$$\nabla^2 \mathbf{u}^* - \frac{\mathbf{u}^*}{\nu \alpha_k \Delta t} = -\frac{1}{\nu \alpha_k} \left(\frac{\tilde{\mathbf{u}}}{\Delta t} + \mathbf{f}^k \right) + \nabla^2 \mathbf{u}^{k-1}, \quad (33)$$

$$\nabla^2 \phi^k = \frac{\nabla \cdot \mathbf{u}^*}{2\alpha_k \Delta t}, \quad (34)$$

$$\mathbf{u}^k = \mathbf{u}^* - 2\alpha_k \Delta t \nabla \phi^k, \quad (35)$$

$$p^k = p^{k-1} + \phi^k - \alpha_k \Delta t \nu \nabla^2 \phi^k, \quad (36)$$

where ϕ is the pseudo-pressure. Finally, we compute the new position of the center of mass of the particle

$$\mathbf{x}_p^k = \mathbf{x}_p^{k-1} + \alpha_k \Delta t \left(\mathbf{u}_p^{k-1} + \mathbf{u}_p^k \right), \quad (37)$$

and with this calculation the Runge–Kutta stage is completed. Note that Eq. (37) is independent of some of the previous steps and therefore can be computed at any point after Eq. (32). As a conclusion, the overall algorithm comprises equations ((24)–(25), (28)–(37)) per Runge–Kutta stage.

While the scope of the present work is restricted to an isolated particle, let us note that the treatment of multi-particle configurations does not present any fundamental difficulty. More specifically, one needs to address the particle–particle (and particle–wall) interactions through some suitable contact model, such as e.g. a soft-sphere discrete element method [32–34]. The force and torque arising through solid–solid contact are added to the right-hand-side of the Newton–Euler Eqs. (3)–(4), and then these terms propagate along the derivation such that they enter the final algorithm in Eqs. (28)–(29). A test of multiple interacting particles in the framework of the present formulation, however, is left for future work.

4. Validation

4.1. Neutrally buoyant particle in 2D Couette flow

As a first validation test case we have selected the lateral migration of a neutrally buoyant circular particle in 2D Couette flow [35–37]. When the neutrally buoyant particle is released from the centerline, the particle remains at that position. However, if the particle is released off the centerline, it has been found that the particle has an additional equilibrium position that depends on the particle Reynolds number [36, 37].

In the present study we have chosen the setup and data of Pan et al. [36] as reference. These authors employed a fictitious domain formulation with distributed Lagrange multipliers for the simulation of neutrally buoyant circular and elliptic particles in 2D. Their computational methodology was based on finite element methods and an operator splitting technique. The following notation and definitions are employed. The distance between the walls is H and their velocity difference is ΔU . A neutrally buoyant particle, $\rho_p/\rho_f = 1$, of radius $a = D/2 = H/8$ is released at a distance to the wall $y_0 = 0.4H$, with initial linear and angular velocities compatible with unperturbed Couette flow at y_0 (see the discussion at the end of Section 2.3). We define the particle Reynolds number as

$$Re_p = \frac{\Delta U a^2}{H \nu},$$

where ν is the kinematic viscosity. The length of the computational domain is $L_x = 2H$. We impose no-slip conditions at the walls and periodicity along the streamwise direction. We employ a uniform grid with grid spacing Δx . We have performed several simulations varying the values of Re_p in the range [1,10] for three grid spacings $D/\Delta x = 25$,

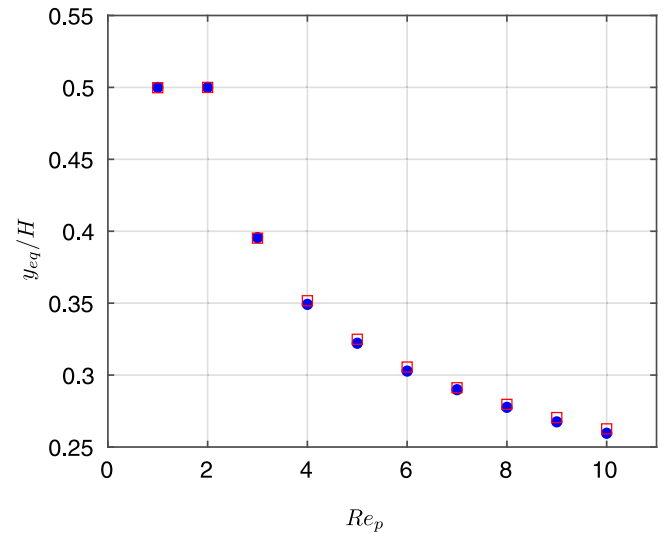


Fig. 1. Equilibrium position, y_{eq}/H , as a function of Re_p . Red squares, present results for $D/\Delta x = 75$. Blue circles, data digitized from Ref. [36].

50 and 75. The corresponding time steps for each grid resolution are $\Delta t/(H/\Delta U) = 0.005$, 0.0025 and 0.00167, respectively.

For all the cases considered, after the particle is released it migrates laterally until reaching an equilibrium position denoted by y_{eq} with constant angular velocity ω_p . For $Re_p = 1$ and 2, the particle settles at the centerline. For $Re_p \geq 3$, the equilibrium position is found off the centerline. The values of y_{eq} for all cases are reported in Table 1 and a comparison with the results of Pan et al. [36] is provided in Fig. 1, showing a very good agreement. The values of ω_p are reported in Table 2; this quantity was not available in the reference. The time evolution of the vertical position of the particle center of mass, y_p , and the particle angular velocity, ω_p , are shown in Fig. 2 for the two cases reported by Pan et al. [36], $Re_p = 1$ and 5. The agreement is also very good.

When comparing the data for the various grid resolutions reported in Tables 1 and 2, it is apparent that the steady-state quantities do not change much with the grid resolution. In particular, the data for $D/\Delta x = 50$ and 75 are very close for all Reynolds numbers considered. The lower resolution case, $D/\Delta x = 25$, presents more deviations with respect to the higher resolution cases. These deviations are not large, with the exception of the case $Re_p = 10$ for which, in the low resolution case, the particle migrates to the centerline.

When considering the transient behavior, some differences are observed among the various grid resolutions. When comparing the cases with $D/\Delta x = 25$ and 50, small but visible differences in the time evolution of y_p appear for $Re_p \geq 3$. When comparing the cases with $D/\Delta x = 50$ and 75, differences in the time evolution of y_p appear for $Re_p > 7$. The trend is that the lower the resolution, the longer the time required for the particle to settle at the equilibrium position. These differences are not shown for the sake of brevity.

4.2. Neutrally buoyant sphere released from rest in a free stream

As a second test case, we have studied the release from rest of a neutrally buoyant sphere ($\rho_p/\rho_f = 1$) of diameter D in a uniform free stream of speed U . Reference data for $Re_p = UD/\nu = 20$ and 100 was obtained with a spectral element code as described in the Appendix. The simulations with the IBM proposed here are performed in a computational domain of size $L_x = 30D$, $L_y = L_z = 5.34D$. The lateral dimensions coincide with those of the study of [23]. At the inlet boundary we impose a Dirichlet boundary condition, $(u, v, w) = (U, 0, 0)$, at the outflow boundary a convective boundary condition and at the lateral boundaries free-slip boundary conditions. We have performed

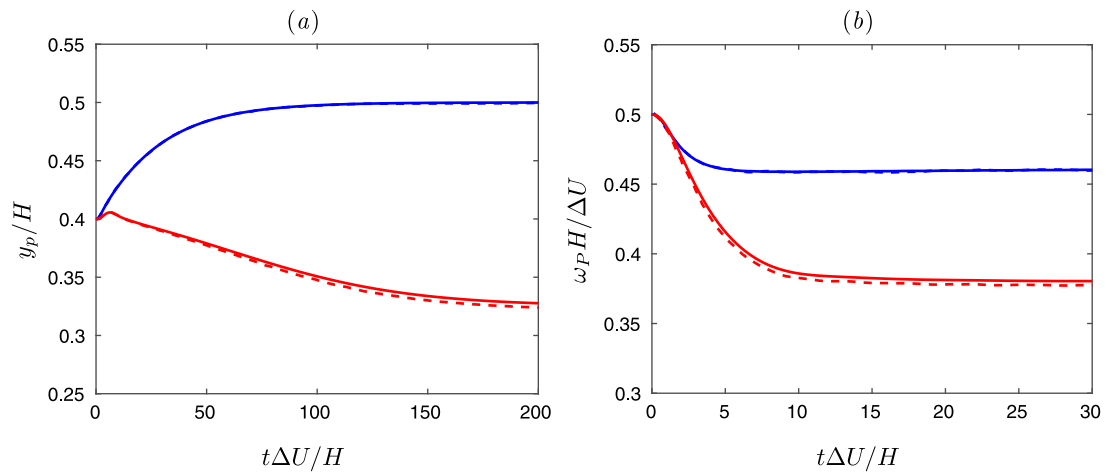


Fig. 2. Time evolution of (a) vertical position of particle center, y_p/H , and (b) particle angular velocity, $\omega_p H/\Delta U$. $Re_p = 1$, Blue. $Re_p = 5$, Red. Solid lines, present simulations with $D/\Delta x = 50$. Dashed lines, data digitized from Pan et al. [36].

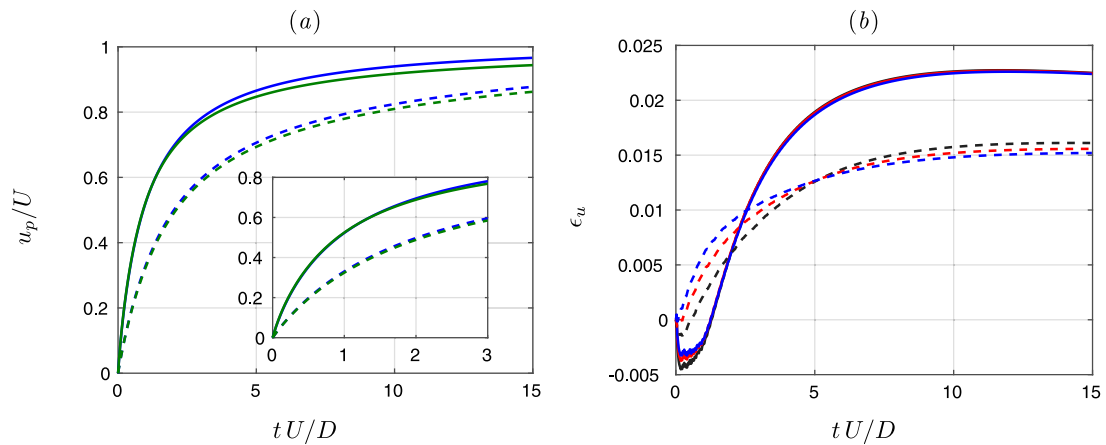


Fig. 3. Time evolution of (a) particle streamwise velocity component, u_p/U , and (b) error with respect to reference data, ϵ_u . Green lines, reference data. $Re_p = 20$, solid lines. $Re_p = 100$, dashed lines. $D/\Delta x = 18$, Black. $D/\Delta x = 24$, Red. $D/\Delta x = 36$, Blue. Inset in (a) highlights the initial phase.

Table 1

Vertical equilibrium position y_{eq}/H as a function of Re_p and $D/\Delta x$.

Re_p	1	2	3	4	5	6	7	8	9	10
$D/\Delta x = 25$	0.5	0.4988	0.3962	0.3538	0.3272	0.3084	0.2946	0.2836	0.2746	0.4896
$D/\Delta x = 50$	0.5	0.5	0.3951	0.3516	0.3250	0.3058	0.2914	0.2801	0.2708	0.2632
$D/\Delta x = 75$	0.5	0.5	0.3952	0.3517	0.3249	0.3056	0.2912	0.2797	0.2705	0.2626

Table 2

Particle angular velocity $\omega_p H/\Delta U$ at equilibrium as a function of Re_p and $D/\Delta x$.

Re_p	1	2	3	4	5	6	7	8	9	10
$D/\Delta x = 25$	0.4612	0.4287	0.4094	0.3957	0.3845	0.3749	0.3666	0.3589	0.3522	0.3411
$D/\Delta x = 50$	0.4610	0.4281	0.4090	0.3955	0.3846	0.3754	0.3675	0.3605	0.3541	0.3483
$D/\Delta x = 75$	0.4608	0.4281	0.4088	0.3955	0.3845	0.3753	0.3675	0.3605	0.3542	0.3485

simulations at $Re_p = 20$ and 100, with grid resolutions $D/\Delta x = 18, 24$ and 36. The time step has been adapted so that the CFL number is approximately 0.1.

The simulations are performed in two phases. In the first phase, the flow is computed with the particle fixed at a distance to the inlet plane $x_0 = 5D$. Once the flow has reached a steady state (i.e the drag force becomes time independent), we release the particle which is then accelerated until it ultimately reaches an equilibrium state with $u_p = U$. Table 3 provides the drag coefficient of the particle before it is released, for both values of Re_p . The variation with the grid resolution is small in all cases. For $Re_p = 20$ ($Re_p = 100$), C_D changes by 0.4% (0.2%)

between $D/\Delta x = 18$ and $D/\Delta x = 36$. The differences observed when comparing the IBM data and the spectral element data are somewhat larger. For $Re_p = 20$ ($Re_p = 100$), C_D differs by 1.8% (2.2%) between the IBM case with $D/\Delta x = 36$ and the spectral element data. Note that these variations are still reasonably small, and that similar differences have already been observed in previous studies where results from boundary-conforming spectral/spectral-element simulations in cylindrical domains were compared to those from immersed-boundary-type non-conforming methods on Cartesian grids [23,26,38]. The remaining differences can be attributed to the larger computational domain of the reference simulation (featuring a larger distance to the inflow

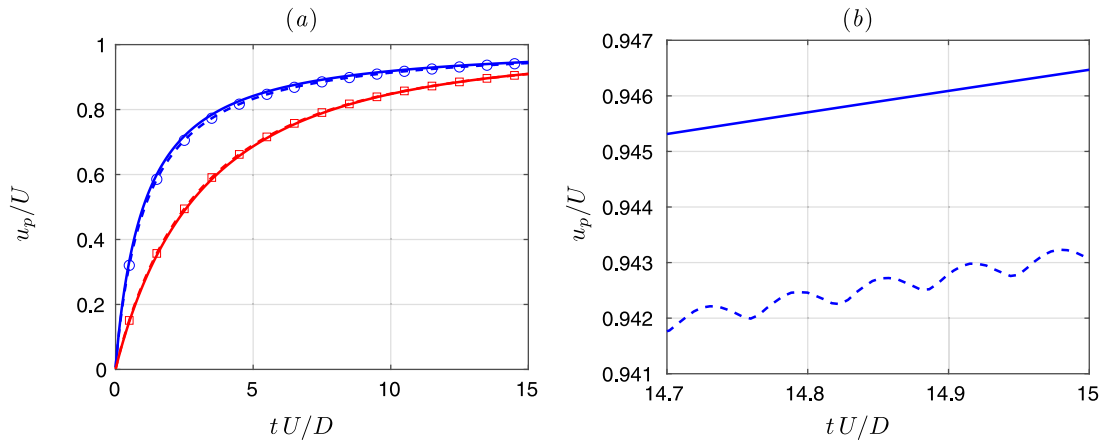


Fig. 4. (a) Time evolution of particle streamwise velocity component, u_p/U . Dashed lines with symbols, reference data from Tschisgale et al. [22]. Solid lines, present results. Blue, $\rho_p/\rho_f = 1.05$. Red, $\rho_p/\rho_f = 5$. (b) Zoom of (a).

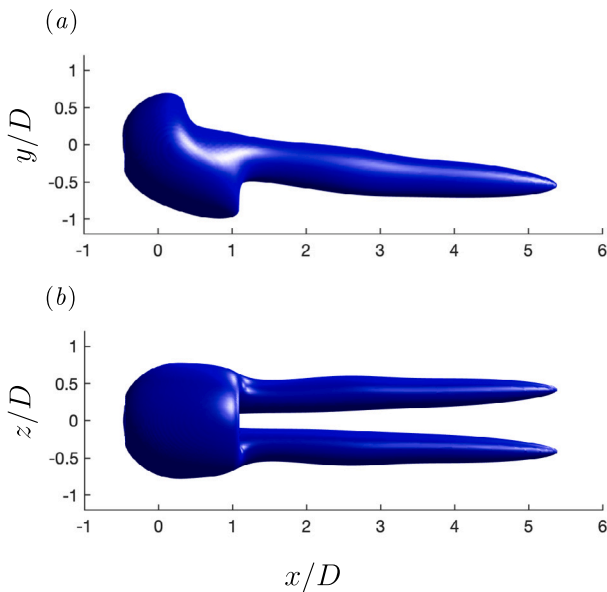


Fig. 5. Flow visualization of wake structure before the rotating particle is released. Two side views of an iso-surface of the second invariant of the velocity gradient tensor, $Q = 0.05U^2/D^2$. Data correspond to the IBM simulation with $D/\Delta x = 36$.

boundary), to the different shape of the domain, and to the different lateral boundary conditions.

Fig. 3a shows the time evolution of the particle streamwise velocity component, u_p , for both Reynolds numbers. The lower the Reynolds number, the faster the velocity of the particle approaches the free-stream velocity. This effect is well captured by the simulations with the IBM method. The agreement with the reference data is good although some differences are visible in the figure. In order to quantify these differences, Fig. 3b, shows the error of the streamwise component of the particle's velocity with respect to the reference data, ϵ_{u_x} , as a function of time. Note that the normalized error for any quantity ϕ here and in the following is defined as

$$\epsilon_{\phi}(t) = (\phi(t) - \phi^{ref}(t)) / \phi_{norm}, \quad (38)$$

where ϕ^{ref} is the reference value and ϕ_{norm} the scale used for normalization. In measuring the error of the streamwise particle velocity ($\phi = u_p$) we use $\phi_{norm} = U$.

The data in Fig. 3b shows that the IBM simulations are generally in better agreement with the reference simulations for $Re_p = 100$ than for

Table 3

Drag coefficient, C_D , of a fixed sphere in a uniform flow as a function of Reynolds number, Re , and grid resolution, $D/\Delta x$. Reference data corresponds to the computations with the spectral element code.

	Reference	$D/\Delta x = 18$	$D/\Delta x = 24$	$D/\Delta x = 36$
$Re = 20$	2.7753	2.8152	2.8195	2.8261
$Re = 100$	1.0962	1.1179	1.1185	1.1201

$Re_p = 20$. For long times, $tU/D > 5$ the error presents a plateau with a value of about 2.3% (1.5%) for $Re_p = 20$ ($Re_p = 100$). The effect of the grid resolution is barely visible for $Re_p = 20$ with the three solid lines collapsing in Fig. 3b, except for very short times, $tU/D < 1$. In the case of $Re_p = 100$, some differences are visible, with the temporal evolution of the error showing a cross-over at about $tU/D \approx 5$. The finest grid with $D/\Delta x = 36$ presents the lowest error of the three in the plateau region, although the differences are small. This indicates that, for this configuration, a grid resolution of $D/\Delta x = 18$ provides reasonable accuracy for simulations at these Reynolds numbers.

A very similar configuration has been studied by other authors [17, 20, 22]. Here we have reproduced their setup to provide a direct assessment of the present methodology. The size of the computational domain is $L_x = 30D$, $L_y = L_z = 15D$. The center of the particle is initially located at $10D$ from the inflow plane and centered with respect to the lateral planes. The number of grid points is $512 \times 256 \times 256$ which corresponds to a grid resolution of $D/\Delta x \approx 17$. The Reynolds number is $Re_p = 20$ and the time step has been adapted so that the CFL number is approximately 0.1. With respect to the simulations discussed above, an additional difference is that Dirichlet boundary conditions are applied at the lateral boundaries, where the velocity of the fluid is set equal to the free stream velocity, U . Two simulations have been performed for the density ratios $\rho_p/\rho_f = 1.05$ and 5. Fig. 4a shows the time evolution of the particle streamwise velocity component, u_p , for both density ratios. The present results are in good agreement with the data from Tschisgale et al. [22]. Fig. 4b shows a zoom of panel (a) towards the end of the simulated interval. This figure illustrates that the present method is free of spurious oscillations.

4.3. Sphere with initial rotation released in a free stream

The test case considered in the previous section is rather simple with unidirectional motion and not involving the rotation of the particle. Therefore, for the final test case, we have tried to avoid those two limitations by defining a test case not reported before. First, while the particle is kept at a fixed location with a uniform inflow velocity, a constant angular particle velocity is imposed so that both lift and drag

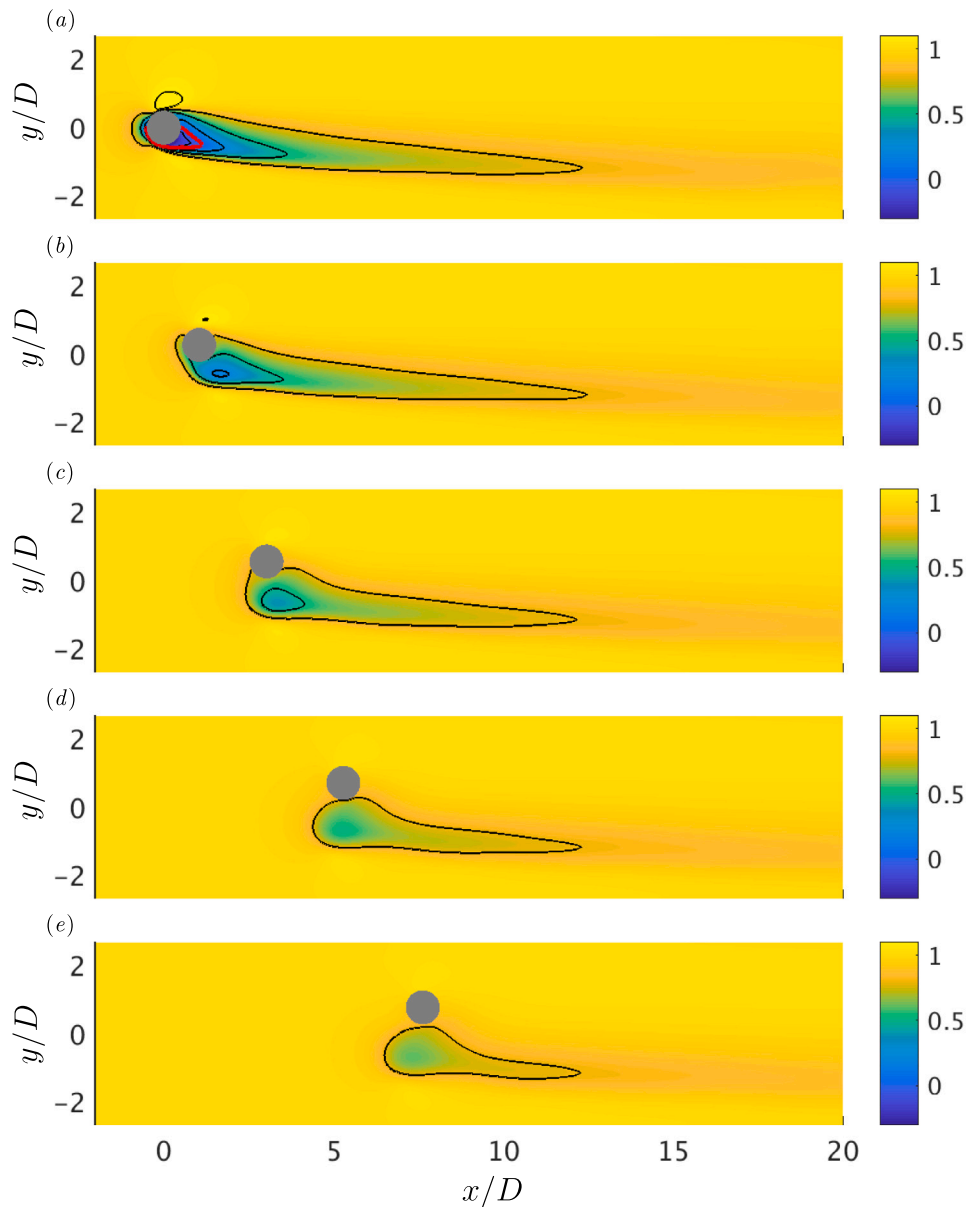


Fig. 6. Contours of streamwise velocity of the fluid phase, u_x , in the plane $z = 0$ for the neutrally buoyant particle which is released after initially rotating at constant angular velocity. Data correspond to the IBM simulation with $D/\Delta x = 36$. From top to bottom, the time instants are $tU/D = 0, 2.5, 5, 7.5$ and 10 after particle release. The red line in panel (a) corresponds to $u/U = 0$. The black lines correspond to $u/U = [-0.1, 0.2, 0.5, 0.8, 1.1]$.

develop. Once the lift and drag forces have converged, we release the particle (at an instant $t = t_1$ that we arbitrarily set to zero, i.e. $t_1 = 0$). In the subsequent evolution, the particle is accelerated in the streamwise direction while, due to the initial lift, a lateral motion is generated. As a consequence, the particle experiences both rotation and lateral motion, which are both damped as time evolves.

We have performed simulations with $Re_p = 100$ and with an initial angular velocity $\omega_{pz}(t \leq t_1) = -\Omega$ with $\Omega D/U = 1$. Reference data was again obtained with the spectral element code described in the [Appendix](#). The computational domain and boundary conditions at inflow and outflow boundaries used in the IBM simulations are the same as in the previous section. However, due to the lateral motion it was not possible to use free-slip conditions at the lateral boundaries since the particle may approach one of the boundaries after it has been released. As a consequence, periodic conditions have been employed at the lateral boundaries. The grid resolution has been varied as in the previous section, with values $D/\Delta x = 18, 24$ and 36 , and the time

step has been adjusted so that $CFL \approx 0.1$. In this section, we consider neutrally buoyant particles ($\rho_p/\rho_f = 1$) and also particles with other density ratios to illustrate the capabilities of the proposed method. Note that gravity is set to zero in all cases, so that the effect of the density ratio only affects the problem via the inertia term.

First, let us characterize the initial state before the particle is released. [Table 4](#) provides the drag and lift coefficients. The effect of grid resolution is once again small: C_D (C_L) varies by approximately 0.1% (0.5%) when comparing the simulations with $D/\Delta x = 18$ and 36 . When comparing with the reference data, we observe that the lift coefficient is closer to the reference data (0.5% difference) than the drag coefficient (2.7% difference). Note that the difference in drag coefficient is of the same order of magnitude as the one observed for the case without rotation of the particle in [Section 4.2](#). This might indicate that the drag coefficient may be more influenced by the differences in shape of the computational domain and lateral boundary conditions than the lift coefficient, the latter being caused by the rotation of the

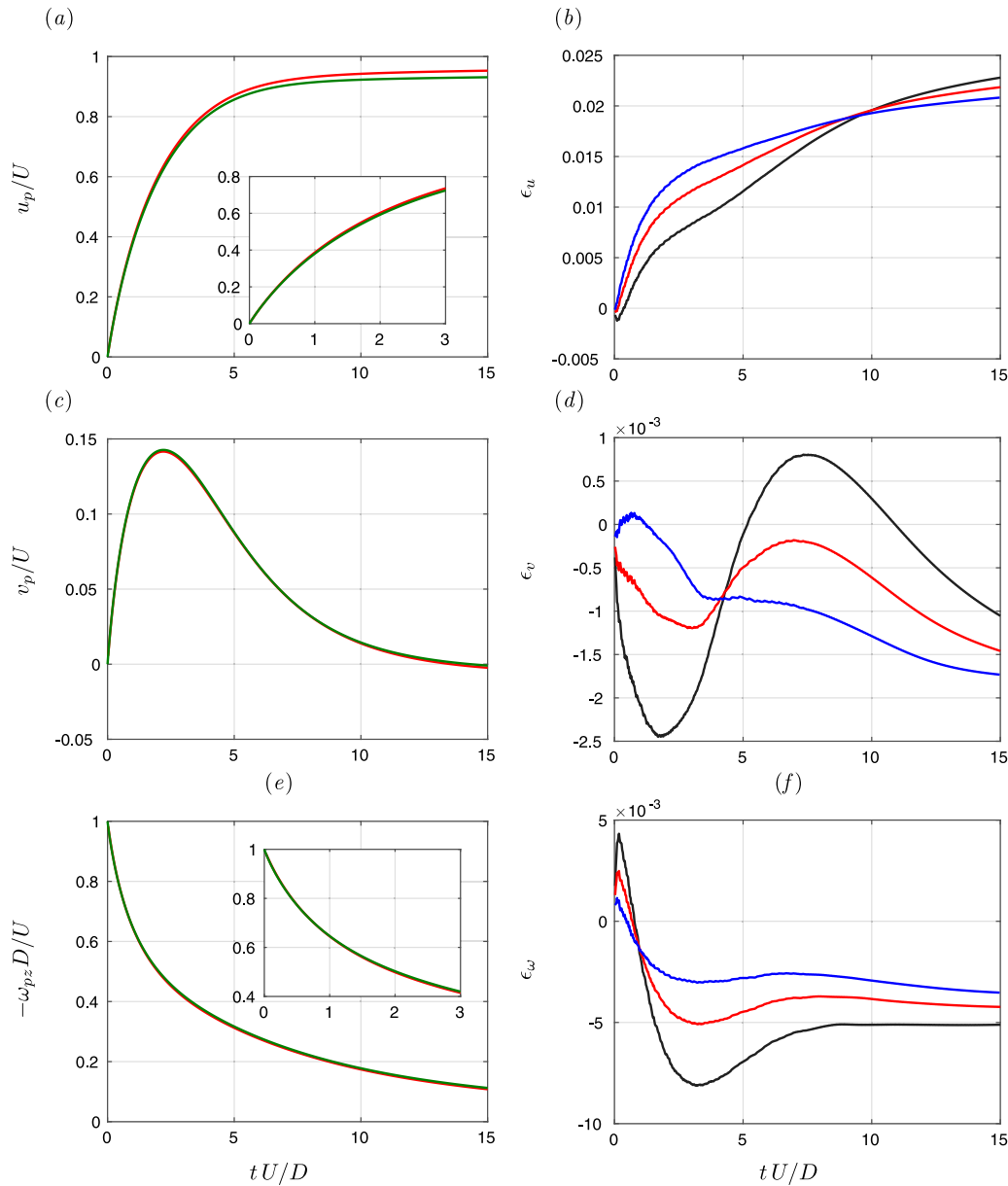


Fig. 7. Time evolution of (a) particle streamwise velocity, u_p/U , (c) particle lateral velocity, v_p/U , (e) particle angular velocity, $\omega_{pz} D/U$, and error with respect to reference data of (b) streamwise velocity, ϵ_u , (d) lateral velocity, ϵ_v , (f) angular velocity, ϵ_ω . Green lines, reference data. $D/\Delta x = 18$, Black. $D/\Delta x = 24$, Red. $D/\Delta x = 36$, Blue. Insets in (a) and (c) highlight the initial phase. In (a), (c), (e) only the results with $D/\Delta x = 24$ are compared to the reference data.

Table 4

Drag and lift coefficients of a fixed, rotating sphere in a uniform flow as a function of grid resolution, $D/\Delta x$. Reference data corresponds to the computations with the spectral element code.

	Reference	$D/\Delta x = 18$	$D/\Delta x = 24$	$D/\Delta x = 36$
C_D	1.2217	1.2539	1.2540	1.2550
C_L	0.4972	0.4973	0.4990	0.5002

particle and therefore having a more local origin. The wake structure of the rotating particle (while it is still held fixed) is visualized from two angles in Fig. 5 with an iso-surface of the second invariant of the velocity gradient tensor, Q . Note that the rotation axis is the z -axis so that the lift force points in the y -direction. Consequently, when the flow is visualized from the z -axis, Fig. 5a, an asymmetry with respect to the xz -plane is observed. When the flow is visualized from the y -axis, Fig. 5b, the double threaded structure of the wake is revealed.

Let us first consider the case of a neutrally buoyant particle. Fig. 6 shows snapshots of the streamwise velocity of the fluid phase in the plane $z = 0$ at several instants. We restrict the analysis to the $z = 0$ plane since the particle does not move laterally along the z -axis as time evolves, being $w_p/U = 0$ for all times. Before the particle is released (panel a) the fluid velocity in this plane is in the range $u/U \in [-0.3, 1.2]$ with the smallest values concentrated directly behind the sphere. As time evolves subsequent to the particle's release, the fluid velocity approaches the free stream velocity U . As an example, for $tU/D = 10$ (panel e) the fluid velocity is in the range $u/U \in [0.6, 1.05]$. Once the particle is released, it starts moving downstream, while also migrating laterally towards positive values of the y coordinate. Thus, the particle becomes detached from its initial wake. As time evolves the particle separates further from the wake and gradually passes the region with low fluid velocities, as clearly seen in panels (c) to (e).

Fig. 7(a,c,e) shows the time evolution of the particle's streamwise and lateral velocity components and the particle's angular velocity. The corresponding error evolutions according to the definition (38) are

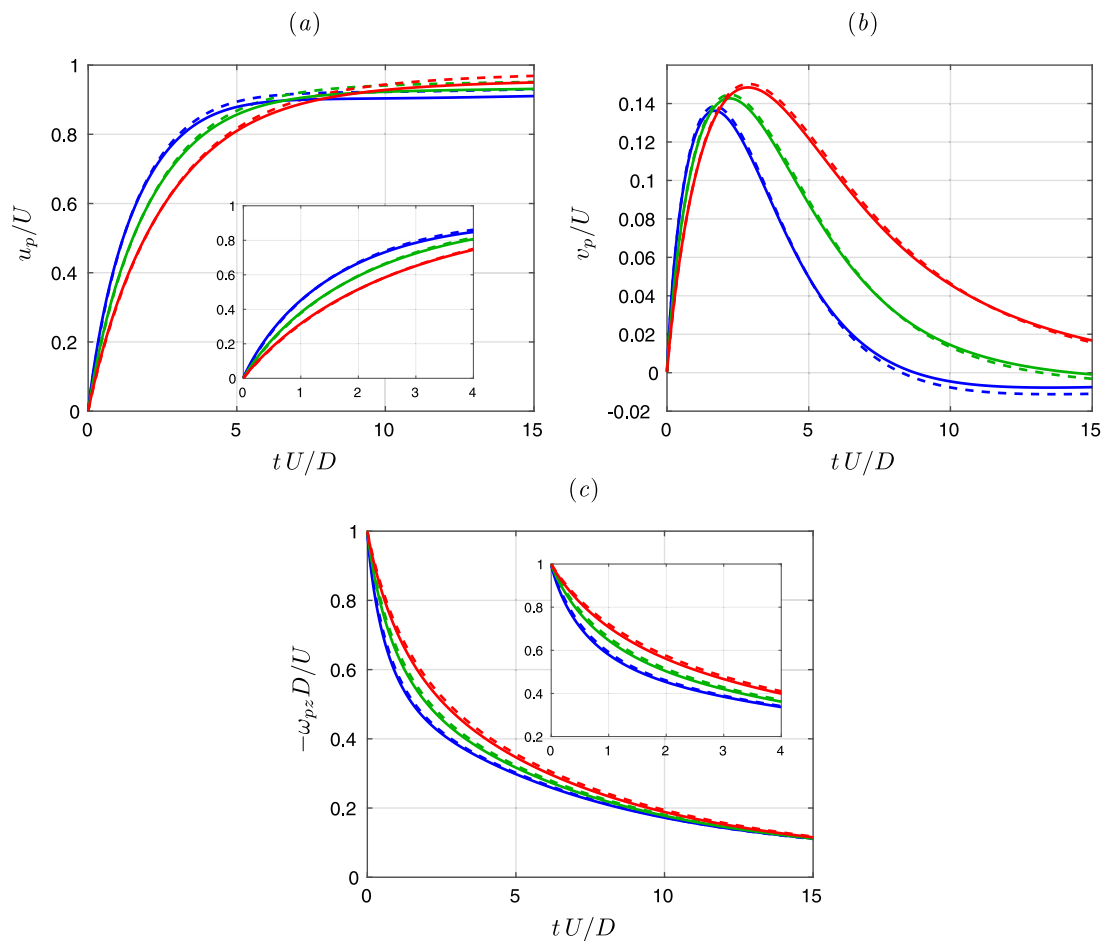


Fig. 8. Time evolution of (a) particle streamwise velocity component, u_p/U , (b) particle lateral velocity component, v_p/U , and (c) particle spanwise angular velocity component, $\omega_{pz}D/U$. Solid lines, reference data. Dashed lines, present results with $D/\Delta x = 24$. Blue, $\rho_p/\rho_f = 0.6$. Green, $\rho_p/\rho_f = 1$. Red, $\rho_p/\rho_f = 1.5$. Insets in (a) and (c) highlight the initial phase.

shown in panels Fig. 7(b, d, f), where we use $\phi_{norm} = U$ for the lateral velocity and $\phi_{norm} = U/D$ for the angular velocity. The agreement with the reference data is generally very good for all three quantities. The largest error amplitudes (smaller than 2.5% at any time during the simulation interval) are recorded for the streamwise velocity. This is in line with our observations in the case of a non-rotating sphere in a uniform flow at the same Reynolds number (Section 4.2), with even the temporal evolution of the errors featuring an analogous cross-over at some intermediate time, and the finest grid yielding the smallest error at later times. It can also be seen that the lateral linear particle velocity as well as its angular velocity are already extremely well captured on the coarsest grid.

Finally, we have performed analogous simulations for a particle with excess density ($\rho_p/\rho_f = 1.5$) and for a less dense particle ($\rho_p/\rho_f = 0.6$) at a fixed numerical resolution $D/\Delta x = 24$. Fig. 8 shows the time evolution of the Lagrangian quantities (the particle's streamwise and lateral velocity components and its angular velocity) for the three density ratios considered. The lower the density ratio the more rapid is the particle response after its release, i.e. a steeper slope is encountered for short times for all three of these quantities. This effect is very well captured by the IBM simulations, which yield a similarly good agreement with respect to the reference simulations irrespective of the value for the density ratio. This means that the present approach to treating density-matched particles can also be applied with very good results to non-density-matched particles.

5. Conclusions

In this work we have presented a simple modification of the direct-forcing immersed boundary method (IBM) originally proposed by Uhlmann [13] in order to enable it to be applied to particulate flows with solid-to-fluid density ratios around unity. Whereas the original formulation features a singularity for neutrally buoyant mobile particles, the present approach is free from such a singularity. The idea is similar to the approach taken by Tschisgale et al. [22]; however, in the present case we apply the volume forcing term to the entire space occupied by the immersed solid object (instead of to the vicinity of its interface only). This leads to a method which can be used seamlessly for density ratios down to $\rho_p/\rho_f > 0.5$. As compared to previous approaches the method is free from spurious oscillations. At the same time it is of a similar computational complexity as the method of Tschisgale et al. [22] which requires an integral over the volume of the particle to be evaluated.

The main difference with respect to the original formulation in [13] lies in the particle velocity update which is performed directly after the preliminary (composite) velocity field has been computed in the absence of any IBM forcing terms. Then the subsequent computation of the volume force at the Lagrangian positions already takes into account the updated velocity. One additional ingredient which arises through this procedure is the necessity to evaluate integrals of the linear and angular velocity field over the volume occupied by the solid

particle [similar to 17], which can be evaluated efficiently at second-order accuracy as sums over the respective quantities available at the Lagrangian force points.

In a first step, the new formulation has been validated using two flow configurations which have been previously established in the literature: (i) lateral migration of a neutrally buoyant circular particle in two-dimensional Couette flow for which reference data generated with a fictitious domain method with Lagrange multipliers and a finite-element discretization [36] is available; (ii) the release from rest of a neutrally buoyant sphere in a free stream with reference data from spectral-element computations using a body-conforming particle-attached mesh [22,39]. In the latter case, the resulting motion of the particle is unidirectional without rotation. In order to subject the present method to a more demanding test, we have introduced a new configuration in which a particle in a free stream is released after an initial phase in which it is translationally fixed with an imposed constant angular velocity. The ensuing unconstrained motion then features time-dependent streamwise and lateral motion as well as rotational dynamics. For this case we have generated high-fidelity reference data with the aid of the spectral-element method of Jenny and Dušek [39] which can henceforth be used for the cross-validation of numerical approaches. We have tested an implementation of our new IBM formulation in a fractional-step context, using a standard semi-implicit Runge–Kutta method in conjunction with second-order finite-differences on a staggered, uniform grid. In all three test cases the present formulation yields a very good agreement with the available reference data. We conclude from our study that the proposed approach is a cost-efficient and accurate modification of the original method which allows for the simulation of fluid systems involving density-matched solid particles. One of the questions which we will address with the aid of the present method in a future investigation is the occurrence of preferential concentration in homogeneous-isotropic turbulence for finite-size, neutrally buoyant particles [40]. In a future communication we will consider the case of neutrally-buoyant non-spherical particles. The present method appears well-suited to that case, as it can be applied with only minor adjustments.

CRediT authorship contribution statement

Manuel García-Villalba: Methodology, Conceptualization, Supervision, Software, Formal analysis, Writing – original draft. **Blanca Fuentes:** Software, Validation, Visualization, Writing – review & editing. **Jan Dušek:** Software, Validation, Writing – review & editing. **Manuel Moriche:** Software, Validation, Writing – review & editing. **Markus Uhlmann:** Methodology, Conceptualization, Funding acquisition, Project administration, Writing – original draft.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the German Research Foundation (DFG) under Project UH 242/11-1. The work was partially performed during visits of MGv to the Karlsruhe Institute of Technology funded by the aforementioned project under the Mercator Fellow scheme. BF was supported by a collaboration scholarship of the Spanish Ministry of Education. We thankfully acknowledge the help of Dr. Gonzalo Arranz with the code implementation. We also thank Prof. Tobias Kempe for

providing his data in electronic form. The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme.

Appendix. Description of reference computations with a spectral element method

The numerical method utilized for the reference computations is identical to the one employed in Uhlmann and Dušek [23], as previously developed in a series of papers [39,41,42]. It utilizes a body-conforming mesh which translates with the particle's center of mass. The computational domain is of cylindrical shape with the cylinder axis aligned in the vertical direction, a radius equal to $8D$ and a height of $37D$ (with the particle center located at a distance $12D$ from the cylinder base). The spatial discretization uses truncated Fourier series in the azimuthal direction and a spectral-element approach in the axial-radial plane (with two-dimensional Legendre polynomial approximation inside the elements). The particle motion is strongly coupled to the fluid solver, as proposed in Jenny and Dušek [39], and a third-order Adams–Bashforth method is used for the temporal discretization. Ambient flow conditions are imposed at the cylinder base, while a zero-stress condition is used at the upper cylinder boundary and at the sides. Likewise, the pressure is set to zero on the boundaries. The azimuthal complex Fourier expansion was truncated at the 7-th mode, and for all elements 6 collocation points were used in each of the two respective spatial directions. The element mesh was the same as that shown in figure 3b of Uhlmann and Dušek [23] around the sphere but was extended to larger radius and height resulting in a larger number of elements (203). The time step was adjusted such that the maximum CFL number equals 0.25. The accuracy of the reference simulations with this choice of numerical parameters has been demonstrated through extensive validation in previous publications [39,43].

References

- [1] Maxey M. Simulation methods for particulate flows and concentrated suspensions. *Annu Rev Fluid Mech* 2017;49:171–93. <http://dx.doi.org/10.1146/annurev-fluid-122414-034408>.
- [2] Wachs A. Particle-scale computational approaches to model dry and saturated granular flows of non-Brownian, non-cohesive, and non-spherical rigid bodies. *Acta Mech* 2019;230(6):1919–80. <http://dx.doi.org/10.1007/s00707-019-02389-9>.
- [3] Uhlmann M, Derksen J, Wachs A, Wang L-P, Moriche M. Efficient methods for particle-resolved direct numerical simulation. In: Subramaniam S, Balachandar S, editors. *Modelling approaches and computational methods for particle-laden turbulent flows*. Academic Press; 2022, p. 147–84.
- [4] Mittal R, Iaccarino G. Immersed boundary methods. *Annu Rev Fluid Mech* 2005;37(1):239–61. <http://dx.doi.org/10.1146/annurev.fluid.37.061903.175743>.
- [5] Verzicco R. Immersed boundary methods: Historical perspective and future outlook. *Annu Rev Fluid Mech* 2023;55:129–55. <http://dx.doi.org/10.1146/annurev-fluid-120720-022129>.
- [6] Peskin CS. Flow patterns around heart valves: A numerical method. *J Comput Phys* 1972;10(2):252–71. [http://dx.doi.org/10.1016/0021-9991\(72\)90065-4](http://dx.doi.org/10.1016/0021-9991(72)90065-4).
- [7] Peskin CS. The immersed boundary method. *Acta Numer* 2002;11:479–517. <http://dx.doi.org/10.1017/S0962492902000077>.
- [8] Sotiropoulos F, Yang X. Immersed boundary methods for simulating fluid–structure interaction. *Prog Aerosp Sci* 2014;65:1–21. <http://dx.doi.org/10.1016/j.paerosci.2013.09.003>.
- [9] Kim W, Choi H. Immersed boundary methods for fluid–structure interaction: A review. *Int J Heat Fluid Flow* 2019;75:301–9. <http://dx.doi.org/10.1016/j.ijheatfluidflow.2019.01.010>.
- [10] Griffith BE, Patankar NA. Immersed methods for fluid–structure interaction. *Annu Rev Fluid Mech* 2020;52:421. <http://dx.doi.org/10.1146/annurev-fluid-010719-060228>.
- [11] Mittal R, Bhardwaj R. Immersed boundary methods for thermofluid problems. *Annu Rev Heat Transf* 2022;24. <http://dx.doi.org/10.1615/AnnualRevHeatTransfer.2022041888>.
- [12] Arranz G, Martinez-Muriel C, Flores O, Garcia-Villalba M. Fluid–structure interaction of multi-body systems: Methodology and applications. *J Fluids Struct* 2022;110:103519. <http://dx.doi.org/10.1016/j.jfluidstructs.2022.103519>.

- [13] Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. *J Comput Phys* 2005;209(2):448–76. <http://dx.doi.org/10.1016/j.jcp.2005.03.017>.
- [14] Zhou K, Balachandar S. An analysis of the spatio-temporal resolution of the immersed boundary method with direct forcing. *J Comput Phys* 2021;424:109862. <http://dx.doi.org/10.1016/j.jcp.2020.109862>.
- [15] Vanella M, Balaras E. A moving-least-squares reconstruction for embedded-boundary formulations. *J Comput Phys* 2009;228(18):6617–28. <http://dx.doi.org/10.1016/j.jcp.2009.06.003>.
- [16] Pinelli A, Naqavi IZ, Piomelli U, Favier J. Immersed-boundary methods for general finite-difference and finite-volume Navier–Stokes solvers. *J Comput Phys* 2010;229(24):9073–91. <http://dx.doi.org/10.1016/j.jcp.2010.08.021>.
- [17] Kempe T, Fröhlich J. An improved immersed boundary method with direct forcing for the simulation of particle laden flows. *J Comput Phys* 2012;231(9):3663–84. <http://dx.doi.org/10.1016/j.jcp.2012.01.021>.
- [18] Breugem W-P. A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows. *J Comput Phys* 2012;231(13):4469–98. <http://dx.doi.org/10.1016/j.jcp.2012.02.026>.
- [19] Kim D, Choi H. Immersed boundary method for flow around an arbitrarily moving body. *J Comput Phys* 2006;212(2):662–80. <http://dx.doi.org/10.1016/j.jcp.2005.07.010>.
- [20] Schwarz S, Kempe T, Fröhlich J. A temporal discretization scheme to compute the motion of light particles in viscous flows by an immersed boundary method. *J Comput Phys* 2015;281:591–613. <http://dx.doi.org/10.1016/j.jcp.2014.10.039>.
- [21] Tavanashad V, Subramaniam S. Fully resolved simulation of dense suspensions of freely evolving buoyant particles using an improved immersed boundary method. *Int J Multiph Flow* 2020;132:103396. <http://dx.doi.org/10.1016/j.ijmultiphaseflow.2020.103396>.
- [22] Tschisgale S, Kempe T, Fröhlich J. A non-iterative immersed boundary method for spherical particles of arbitrary density ratio. *J Comput Phys* 2017;339:432–52. <http://dx.doi.org/10.1016/j.jcp.2017.03.026>.
- [23] Uhlmann M, Dušek J. The motion of a single heavy sphere in ambient fluid: A benchmark for interface-resolved particulate flow simulations with significant relative velocities. *Int J Multiph Flow* 2014;59:221–43. <http://dx.doi.org/10.1016/j.ijmultiphaseflow.2013.10.010>.
- [24] Tschisgale S, Kempe T, Fröhlich J. A general implicit direct forcing immersed boundary method for rigid particles. *Comput & Fluids* 2018;170:285–98. <http://dx.doi.org/10.1016/j.compfluid.2018.04.008>.
- [25] Arranz G, Moriche M, Uhlmann M, Flores O, García-Villalba M. Kinematics and dynamics of the auto-rotation of a model winged seed. *Bioinspir Biomim* 2018;13(3):036011. <http://dx.doi.org/10.1088/1748-3190/aab144>.
- [26] Moriche M, Uhlmann M, Dušek J. A single oblate spheroid settling in unbounded ambient fluid: A benchmark for simulations in steady and unsteady wake regimes. *Int J Multiph Flow* 2021;136:103519. <http://dx.doi.org/10.1016/j.ijmultiphaseflow.2020.103519>.
- [27] Yu Z, Shao X. A direct-forcing fictitious domain method for particulate flows. *J Comput Phys* 2007;227(1):292–314. <http://dx.doi.org/10.1016/j.jcp.2007.07.027>.
- [28] Ascher UM, Greif C. *A first course on numerical methods*. SIAM; 2011.
- [29] Sharma N, Patankar NA. A fast computation technique for the direct numerical simulation of rigid particulate flows. *J Comput Phys* 2005;205(2):439–57. <http://dx.doi.org/10.1016/j.jcp.2004.11.012>.
- [30] Roma AM, Peskin CS, Berger MJ. An adaptive version of the immersed boundary method. *J Comput Phys* 1999;153(2):509–34. <http://dx.doi.org/10.1006/jcph.1999.6293>.
- [31] Rai MM, Moin P. Direct simulations of turbulent flow using finite-difference schemes. *J Comput Phys* 1991;96(1):15–53. [http://dx.doi.org/10.1016/0021-9991\(91\)90264-L](http://dx.doi.org/10.1016/0021-9991(91)90264-L).
- [32] Wachs A. A DEM-DLM/FD method for direct numerical simulation of particulate flows: Sedimentation of polygonal isometric particles in a Newtonian fluid with collisions. *Comput & Fluids* 2009;38(8):1608–28. <http://dx.doi.org/10.1016/j.compfluid.2009.01.005>.
- [33] Kempe T, Fröhlich J. Collision modelling for the interface-resolved simulation of spherical particles in viscous fluids. *J Fluid Mech* 2012;709:445–89. <http://dx.doi.org/10.1017/jfm.2012.343>.
- [34] Kidanemariam A, Uhlmann M. Interface-resolved direct numerical simulations of the erosion of a granular bed sheared by laminar channel flow. *Int J Multiph Flow* 2014;67:174–88. <http://dx.doi.org/10.1016/j.ijmultiphaseflow.2014.08.008>, arXiv:1409.0339.
- [35] Ding E-J, Aidun CK. The dynamics and scaling law for particles suspended in shear flow with inertia. *J Fluid Mech* 2000;423:317–44. <http://dx.doi.org/10.1017/S0022112000001932>.
- [36] Pan T-W, Huang S-L, Chen S-D, Chu C-C, Chang C-C. A numerical study of the motion of a neutrally buoyant cylinder in two dimensional shear flow. *Comput Fluids* 2013;87:57–66. <http://dx.doi.org/10.1016/j.compfluid.2012.11.021>.
- [37] Fox AJ, Schneider JW, Khair AS. Inertial bifurcation of the equilibrium position of a neutrally-buoyant circular cylinder in shear flow between parallel walls. *Phys Rev Res* 2020;2(1):013009. <http://dx.doi.org/10.1103/PhysRevResearch.2.013009>.
- [38] Rettinger C, Råde U. A comparative study of fluid-particle coupling methods for fully resolved lattice Boltzmann simulations. *Comput & Fluids* 2017;154:74–89. <http://dx.doi.org/10.1016/j.compfluid.2017.05.033>.
- [39] Jenny M, Dušek J. Efficient numerical method for the direct numerical simulation of the flow past a single light moving spherical body in transitional regimes. *J Comput Phys* 2004;194:215–32. <http://dx.doi.org/10.1016/j.jcp.2003.09.004>.
- [40] Fiabane L, Zimmermann R, Volk R, Pinton J-F, Bourgoin M. Clustering of finite-size particles in turbulence. *Phys Rev E* 2012;86(035301(R)). <http://dx.doi.org/10.1103/PhysRevE.86.035301>.
- [41] Ghidersa B, Dušek J. Breaking of axisymmetry and onset of unsteadiness in the wake of a sphere. *J Fluid Mech* 2000;423:33–69. <http://dx.doi.org/10.1017/S0022112000001701>.
- [42] Kotouc M, Bouchet G, Dušek J. Loss of axisymmetry in the mixed convection, assisting flow past a heated sphere. *Int J Heat Mass Transfer* 2008;51(11):2686–700. <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2007.10.005>.
- [43] Bouchet G, Mebarek M, Dušek J. Hydrodynamic forces acting on a rigid fixed sphere in early transitional regimes. *Eur J Mech B Fluids* 2006;25:321–36. <http://dx.doi.org/10.1016/j.euromechflu.2005.10.001>.