Received: 6 January 2022 | Revised: 7 June 2022 | Accepted: 20 June 2022

DOI: 10.1002/fld.5128

RESEARCH ARTICLE

WILEY

Particle-resolved simulations of four-way coupled, polydispersed, particle-laden flows

Yinuo Yao[1,2] | Edward Biegert[3] | Bernhard Vowinckel[3,4] | Thomas Köllner[3,5] | Eckart Meiburg[3] | Sivaramakrishnan Balachandar[6] | Craig S. Criddle[2] | Oliver B. Fringer[1]

[1]The Bob and Norma Street Environmental Fluid Mechanics Laboratory, Department of Civil and Environmental Engineering, Stanford University, Stanford, California, USA

[2]Codiga Resource Recovery Center at Stanford, Department of Civil and Environmental Engineering, Stanford University, Stanford, California, USA

[3]Mechanical Engineering, University of California, Santa Barbara, Santa Barbara, California, USA

[4]Leichtweiß-Institute for Hydraulic Engineering and Water Resources, Technische Universität, Braunschweig, Braunschweig, Germany

[5]CADFEM GmbH, Munich, Germany

[6]Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida, USA

Correspondence
Yinuo Yao, The Bob and Norma Street Environmental Fluid Mechanics Laboratory, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305, USA.
Email: yaoyinuo@stanford.edu

Present address
Yinuo Yao, Department of Energy Resources Engineering, Stanford University, Stanford, California, USA

Funding information
California Energy Commission, Grant/Award Number: EPC-16-017; National Science Foundation, Grant/Award Numbers: ACI-1548562, 1028968; Office of Naval Research, Grant/Award Number: N00014-16-1-2256; German Research Foundation, Grant/Award Number: VO2413/2-1

Abstract

We present a collocated-grid framework for direct numerical simulations of polydisperse particles submerged in a viscous fluid. The fluid-particle forces are coupled with the immersed boundary method (IBM) while the particle-particle forces are modeled with a combination of contact and lubrication models, adapted for collocated grids. Our method is modified from the staggered-grid IBM of previous authors to a collocated-grid IBM by adapting the fluid and particle solvers. The method scales well on high-performance parallel computing platforms. It has been validated against various cases and is able to reproduce experimental results. Tuning parameters have been thoroughly calibrated to ensure the accuracy of the method. Finally, we demonstrate the capability of the method to simulate both monodispersed and bidispersed fluidized beds and reproduce the power law relationship between the inflow velocity and the porosity.

KEYWORDS

collocated, immersed boundary method, numerical methods, polydispersed particle-laden flow

1 | INTRODUCTION

Particle-laden flows are common in both industrial and natural systems. Typical examples of industrial applications in treating wastewater include sedimentation tanks and fluidized beds and so forth,[1-3] while examples of natural systems include sediment transport in estuaries.[4] Understanding the fluid-particle and particle-particle interactions enhances the design of various industrial systems. For instance, predicting the upflow velocity to achieve the desired porosity is critical in fluidized-bed applications. Hence, Richardson and Zaki[5] proposed a power law relationship to correlate the effect of superficial velocity (average inlet velocity) to the porosity of the particle phase by studying the macroscopic properties and behavior of the systems. However, due to limitations in the experimental data collection technologies, understanding and quantifying the microscale behavior has been a persistent challenge.[6] In recent years, with improvements in both computational power and numerical methods, particle-resolved simulations (PRS), or the direct numerical simulation (DNS) of particle-laden flows, has received constant attention and development, owing to its capability of fully resolving the flow around the particles and not relying on the accuracy of drag models.

Over the past few decades, different PRS methods have been developed. Zhang and Prosperetti[7] developed and Willen and Prosperetti[8] employed the PHYSALIS method that assumes flow near spheres to be Stokes flow regardless of the mean flow Reynolds number. As such, PHYSALIS utilizes the analytical solution for Stokes flow around a sphere to simulate particle-laden flows. Glowinski et al.[9,10] developed a distributed Lagrange multiplier (DLM) method that enforces rigid body motion of the fluid in a sphere. Ladd[11] developed the lattice-Boltzmann method (LBM) in which particles are represented by lattice nodes. Another family of methods is the immersed boundary method (IBM) first proposed by Peskin[12,13] and then extended to particle-laden flow.[14-22] One challenge associated with moving boundaries is the oscillation of the IBM forces due to (1) the spatial discontinuity in pressure, (2) the temporal discontinuity in the Eulerian velocity at the interface, and (3) mass conservation violation.[23] Uhlmann[14] introduced the concept of direct forcing and reduced the oscillation due to the temporal discontinuity by computing the IBM force at Lagrangian markers and interpolating between an Eulerian and Lagrangian marker with a regularized Dirac delta function. Kempe and Fröhlich[24] reduced the minimum stable particle-fluid density ratio from 1.2 to 0.2. Kempe and Fröhlich[24] and Wang et al.[18] further improved the accuracy of the direct forcing IBM by implementing iterative outer forcing loops to enforce a more accurate no-slip boundary conditions on particle surfaces. Akiki and Balachandar[15] extended IBM to non-uniform grids. Recently, Zhou and Balachandar[25] conduct a theoretical analysis on the optimum number of Lagrangian markers need to achieve most accurate simulations. Yang and Balachandar[26] developed a scalable parallel algorithm for IBM with the concept of double binned ghost particle (DBGP).

For systems with concentrated suspensions, particle-particle interactions are inevitable and collision models are necessary.[14,16,27-29] In the original direct forcing IBM, Uhlmann[14] employed the repulsion potential model proposed by Glowinski et al.[9] In this approach, an arbitrary force is added when particles are less than two grid cells apart, a situation not resolvable by IBM. The main purpose of this approach is to prevent particles from contacting one another. Studies that employ this model usually have small volume fractions where particle-particle interactions are not significant.[30] To obtain more accurate particle-particle interactions, collision models have been implemented in the form of lubrication and contact models. Lubrication models account for the forces exerted on particles when they are moving towards and away each other before and after coming into contact, while the contact model is used when the surfaces of the particles touch. For normal contact forces, two popular approaches are the hard- and soft-sphere based models. As discussed by Kempe and Fröhlich,[24] the former cannot model simultaneous collisions which can be significant in concentrated suspensions. In addition, both models usually require the time step size Δt to be small due to a high material stiffness coefficient k_n. To reduce the stiffness, Zaidi et al.[31] chooses a value of k_n that achieves a balance between accuracy and computational cost. Kempe and Fröhlich,[24] on the other hand, proposed an algorithm to dynamically optimize the stiffness coefficient k_n and damping coefficient d_n based on the dry restitution coefficient e_{dry} of the material and a calibrated collision time step T_c. For tangential contact models, Kempe and Fröhlich[24] designed a model to exactly enforce the no-slip condition between particles. As pointed out by Biegert et al.,[16] this model does not converge to a steady solution for enduring contact. Instead, Luding[32] and Thornton et al.[33] proposed and Biegert et al.[16] adopted a spring dashpot model that converged to a steady solution.

Although there have been significant efforts on developing and improving IBM and collision models, the aforementioned developments are based on the staggered grid formulation. Staggered grids have the advantage of pressure-velocity coupling and satisfying continuity with machine precision. The collocated grid, on the other hand, has the advantage that all variables are located at the same location and the finite-volume approach is straightforward to implement, thus simplifying its use in complex geometries.[34] Another significant advantage of the IBM method is its flexibility of implementation in any established Navier–Stokes solver. Lee and Balachandar[35,36] adapted the staggered formulation by Uhlmann[14] to a collocated formulation to simulate a single sphere in a shear flow. However, Uhlmann's method does not account for particle collisions and errors due to the explicit formulation of immersed boundary forces.[24] Various authors have developed collocated IBMs for complex geometries which do not include the effects of interactions between different geometries.[37-40] To date, no comprehensive collocated direct forcing IBM with collision models for polydispersed particles has been proposed. In the present work, we build on work by multiple researchers[14,16,27] and propose a simulation framework that combines the advantages of the collocated direct forcing IBM and the collision models for polydisperse particles, resulting in a comparable accuracy with the staggered-grid approach. This work will serve as a building block to couple IBM with existing collocated-grid Navier–Stokes solvers.

The article is organized as follows. In Section 2.1, we briefly discuss the original direct-forcing IBM on a staggered grid by Uhlmann.[14] In Section 2.2, we present the modifications to the original IBM for a collocated grid. Sections 2.2.1–2.2.4 focus on the Navier–Stokes solver, Sections 2.2.5–2.2.8 adapt the collision models to a collocated grid and Section 2.2.9 and 2.2.10 modify the particle solver for coupling of IBM with the collocated-grid Navier–Stokes solver. In Sections 3.1–3.3, simulations are presented to (1) validate the accuracy and computational efficiency of the proposed collocated-grid IBM and (2) determine the tuning parameters that are needed to achieve accurate simulations. In Section 3.5, monodispersed fluidization simulations are conducted and compared to predictions by existing models.

2 | METHODOLOGY

2.1 | Original direct-forcing IBM on a staggered grid

The governing equations are formulated as the modified unsteady Navier–Stokes equation for an incompressible fluid. The IBM force is accounted for with a source term, $\boldsymbol{f}_{\mathrm{IBM}}$, which is added to the Navier–Stokes equation to enforce no-slip boundary conditions on the particle surfaces. With this forcing, the modified unsteady, Navier–Stokes equation for an incompressible fluid is given by

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} = -\boldsymbol{\nabla} P + \nu_f \nabla^2 \boldsymbol{u} + \boldsymbol{f}_{\mathrm{IBM}}, \tag{1}$$

subject to continuity,

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

where $\boldsymbol{u} = \begin{bmatrix} u & v & w \end{bmatrix}^T$ is the fluid velocity vector in Cartesian coordinates, $P = (P_{tot} - \rho_f g z)/\rho_f$ is the perturbation pressure (relative to the hydrostatic pressure $\rho_f g z$, where g is the gravitational acceleration in the z direction) normalized by the fluid density ρ_f, P_{tot} is the total pressure, ν_f is the kinematic viscosity of the fluid and $\boldsymbol{f}_{\mathrm{IBM}} = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T$ is the IBM force vector. The original direct-forcing IBM method proposed by Uhlmann[14] eliminates strong oscillations arising from direct interpolation of $\boldsymbol{f}_{\mathrm{IBM}}$ from the neighboring Eulerian cells.[17,41] Uhlmann[14] proposed to compute the IBM force at the center of Lagrangian markers located at $\boldsymbol{X}_l$ which represent a thin shell of thickness h on the particle surface with volume

$$N_l \Delta V_l = \frac{4}{3}\pi \left[\left(\frac{d_p}{2} + \frac{h}{2} \right)^3 - \left(\frac{d_p}{2} - \frac{h}{2} \right)^3 \right], \tag{3}$$

where N_l is the total number of Lagrangian markers, $\Delta V_l \approx h^3$ is the approximate volume of each Lagrangian marker, d_p is the particle diameter and h is the Eulerian grid spacing, which is isotropic in the three Cartesian coordinate directions (x, y, z) such that $\Delta x = \Delta y = \Delta z = h$. The desired motion of particles at particle surface locations $\boldsymbol{X}_l$ is defined as

$$\boldsymbol{u}_p^d (\boldsymbol{X}_l) = \boldsymbol{u}_p + \boldsymbol{\omega}_p \times \left(\boldsymbol{X}_l - \boldsymbol{x}_p \right), \tag{4}$$

where $\boldsymbol{u}_p$ and $\boldsymbol{\omega}_p$ are governed by Newton's second law governing linear and angular momentum of a spherical particle

$$m_p \frac{\mathrm{d}\boldsymbol{u}_p}{\mathrm{d}t} = \oint_S \boldsymbol{\tau} \cdot \boldsymbol{n} \ \mathrm{d}S + V_p(\rho_p - \rho_f)\boldsymbol{g} + \boldsymbol{F}_{c,p}, \tag{5a}$$

$$I_p \frac{\mathrm{d}\boldsymbol{\omega}_p}{\mathrm{d}t} = \oint_S \boldsymbol{r} \times (\boldsymbol{\tau} \cdot \boldsymbol{n}) \ \mathrm{d}S + \boldsymbol{T}_{c,p}, \tag{5b}$$

where $m_p = \rho_p V_p$ is the mass of the particle, ρ_p and V_p are the particle density and volume, $\boldsymbol{\tau}$ is the hydrodynamic stress tensor, $\boldsymbol{n}$ is the outward-pointing normal vector on the particle surface S, $\boldsymbol{g} = -g\boldsymbol{e}_z$ is the gravitational acceleration vector in the $\boldsymbol{e}_z$ direction, I_p is the particle moment of inertia, $\boldsymbol{r}$ is the position vector between the particle center $\boldsymbol{x}_p$ and particle surface and $\boldsymbol{F}_{c,p}$ and $\boldsymbol{T}_{c,p}$ are the forces and torques exerted on the particle due to collisions including lubrication and contact forces. Substitution of the Navier–Stokes equation (1) (with the full pressure and gravity terms included) into Equation (5a) and employing Gauss's theorem, the particle motion equations are given by

$$m_p \frac{\mathrm{d}\boldsymbol{u}_p}{\mathrm{d}t} = -\rho_f \int_{\Omega_p} \boldsymbol{f}_{\mathrm{IBM}}\ \mathrm{d}V + \rho_f \frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_p} \boldsymbol{u}\ \mathrm{d}V + V_p \left(\rho_p - \rho_f\right) \boldsymbol{g} + \boldsymbol{F}_{c,p}, \tag{6a}$$

$$I_p \frac{\mathrm{d}\boldsymbol{\omega}_p}{\mathrm{d}t} = \rho_f \int_{\Omega_p} \boldsymbol{r} \times \boldsymbol{f}_{\mathrm{IBM}}\ \mathrm{d}V + \rho_f \frac{\mathrm{d}}{\mathrm{d}t} \int_{\Gamma_p} \boldsymbol{r} \times \boldsymbol{u}\ \mathrm{d}V + \boldsymbol{T}_{c,p}, \tag{6b}$$

where Ω_p is the particle volume. Following Uhlmann,[14] the rate-of-change terms can be simplified by assuming rigid-body motion such that

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_p} \boldsymbol{u}\ \mathrm{d}V = V_p \frac{\mathrm{d}\boldsymbol{u}_p}{\mathrm{d}t}, \tag{7a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_p} \boldsymbol{r} \times \boldsymbol{u}\ \mathrm{d}V = \frac{I_p}{\rho_p} \frac{\mathrm{d}\boldsymbol{\omega}_{\boldsymbol{p}}}{\mathrm{d}t}. \tag{7b}$$

which gives the governing equations

$$m_p \left(1 - \frac{\rho_f}{\rho_p}\right) \frac{\mathrm{d}\boldsymbol{u}_p}{\mathrm{d}t} = -\rho_f \int_{\Omega_p} \boldsymbol{f}_{\mathrm{IBM}}\ \mathrm{d}V + V_p \left(\rho_p - \rho_f\right) \boldsymbol{g} + \boldsymbol{F}_{c,p}, \tag{8a}$$

$$I_p \left(1 - \frac{\rho_f}{\rho_p}\right) \frac{\mathrm{d}\boldsymbol{\omega}_p}{\mathrm{d}t} = \rho_f \int_{\Omega_p} \boldsymbol{r} \times \boldsymbol{f}_{\mathrm{IBM}}\ \mathrm{d}V + \boldsymbol{T}_{c,p}. \tag{8b}$$

However, this can lead to a singularity when $\rho_p = \rho_f$, a problem that is addressed in Section 2.2.9.

Figure 1 shows an example of the Eulerian grid and the Lagrangian markers representing the particle on both staggered and collocated grids. To interpolate between Eulerian and Lagrangian quantities, the one-dimensional kernel in the x-direction based on the three-point regularized Dirac delta function δ_h^{3p} is utilized and defined as

$$\delta_{h,x}^{3p} \left(x - X_l\right) = \frac{1}{h} \phi_3 \left(r\right), \tag{9}$$

where $\phi_3\left(r\right)$ is the one-dimensional, three-point function

$$\phi_3\left(r\right) = \begin{cases} \frac{1}{6}\left(5 - 3|r| - \sqrt{-3(1-|r|)^2 + 1}\right), & 0.5 \le |r| \le 1.5, \\ \frac{1}{3}\left(1 + \sqrt{-3|r|^2 + 1}\right), & |r| \le 0.5, \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

and $r = \left(x - X_l\right)/h$ is the normalized distance from the Lagrangian marker.[42] In three dimensions, the three-point regularized Dirac delta function $\delta_{h,3D}^{3p}$ is then given by

$$\delta_{h,3D}^{3p} \left(\boldsymbol{x} - \boldsymbol{X}_l\right) = \delta_{h,x}^{3p} \left(x - X_l\right) \delta_{h,y}^{3p} \left(y - Y_l\right) \delta_{h,z}^{3p} \left(z - Z_l\right). \tag{11}$$

This is used to interpolate quantities from the Eulerian grid onto the Lagrangian marker and vice-versa. In what follows, $\delta_{h,3D}$ (without the superscript) implies either the three- or four-point functions which are defined in Equations (10) and (19) respectively. The choice is clarified in the test case.

In Uhlmann,[14] the advection term is integrated in time with the explicit, three-step Runge–Kutta scheme described in Rai and Moin.[43] The viscous term is time integrated with the second-order implicit Crank–Nicolson scheme to eliminate the associated stability constraints. The fractional step method by Rai and Moin[43] is used to couple the velocity and the pressure terms and enforces continuity. Overall, the fluid-solver with direct-forcing IBM on a uniform staggered grid by Uhlmann[14] is given by the following steps:

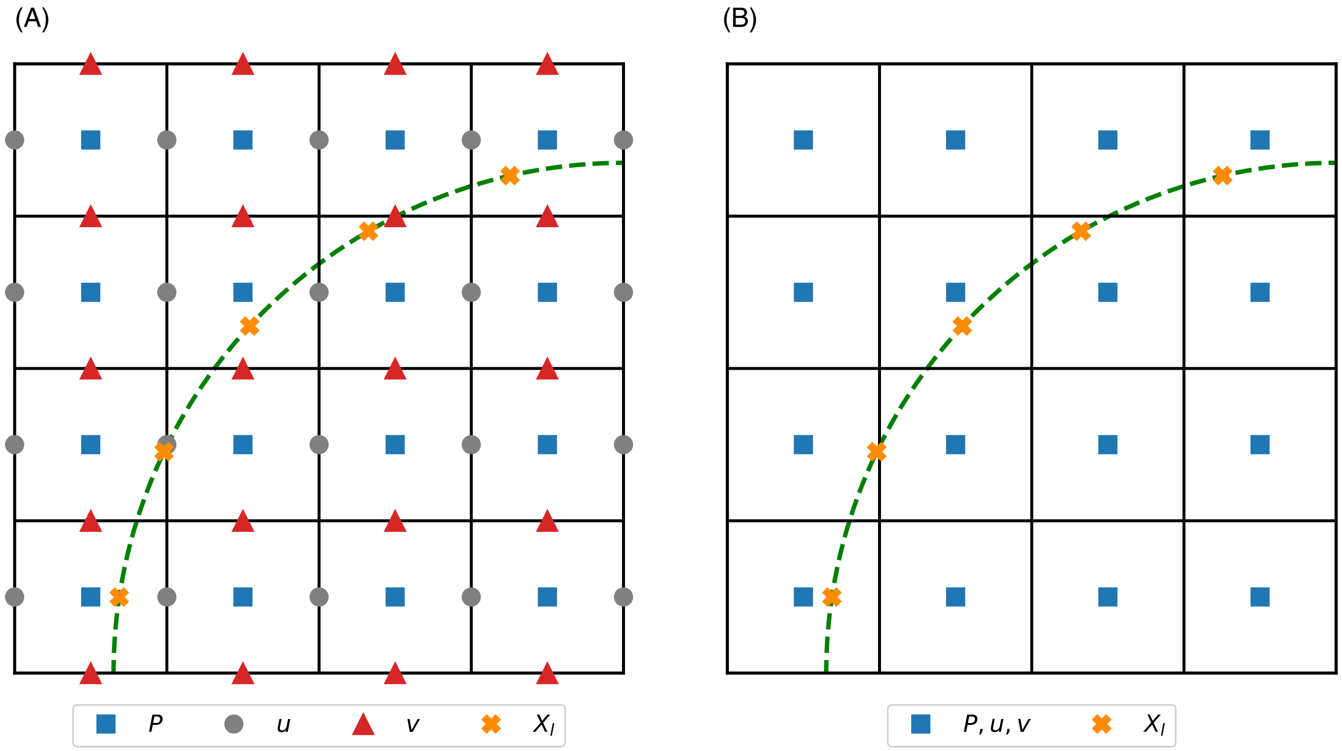

FIGURE 1 Variable locations on (A) staggered and (B) collocated grids. Green dashed line represents the particle surface. [Colour figure can be viewed at wileyonlinelibrary.com]

1. Predictor step without direct forcing

$$\frac{\widetilde{\boldsymbol{u}}-\boldsymbol{u}^{k-1}}{\Delta t} = 2\alpha_k \nu_f \nabla^2 \boldsymbol{u}^{k-1} - 2\alpha_k \boldsymbol{\nabla} P^{k-1} - \gamma_k(\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u})^{k-1} - \zeta_k(\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u})^{k-2}, \tag{12a}$$

2. Project predicted Eulerian grid velocities onto the Lagrangian marker

$$\widetilde{\boldsymbol{U}}\left(\boldsymbol{X}_{l,n}\right) = \sum_{i,j,k}^{N_i,N_j,N_k} \widetilde{\boldsymbol{u}}\left(\boldsymbol{x}_{i,j,k}\right) \delta_{h,3D}^{3p}\left(\boldsymbol{x}_{\boldsymbol{i},\boldsymbol{j},\boldsymbol{k}} - \boldsymbol{X}_{l,n}\right) \Delta x \Delta y \Delta z, \tag{12b}$$

where $\delta_{h,3D}^{3p}$ is the three-point regularized Dirac-delta function (Equation 9).

3. Determine the Lagrangian marker force to enforce the no-slip condition

$$\boldsymbol{F}\left(\boldsymbol{X}_{l,n}\right) = \frac{\boldsymbol{u}_p^d\left(\boldsymbol{X}_{l,n}\right) - \widetilde{\boldsymbol{U}}\left(\boldsymbol{X}_{l,n}\right)}{2\alpha_k \Delta t}, \tag{12c}$$

4. Interpolate the Lagrangian marker force back onto the Eulerian grid

$$\boldsymbol{f}_{\mathrm{IBM}}(\boldsymbol{x}_{i,j,k}) = \sum_{n,l}^{N_p,N_l} \boldsymbol{F}\left(\boldsymbol{X}_{l,n}\right) \delta_{h,3D}^{3p}\left(\boldsymbol{x}_{i,j,k} - \boldsymbol{X}_{l,n}\right) \Delta V_{l,n}, \tag{12d}$$

5. Predictor step with direct forcing

$$\boldsymbol{u}^* = \widetilde{\boldsymbol{u}} + \Delta t\left(2\alpha_k \boldsymbol{f}_{\mathrm{IBM}} - \alpha_k \nu_f \nabla^2 \boldsymbol{u}^{k-1} + \alpha_k \nu_f \nabla^2 \boldsymbol{u}^*\right), \tag{12e}$$

6. Pressure Poisson equation to obtain the pseudopressure ϕ

$$\nabla^2 \phi = \frac{1}{2\alpha_k \Delta t} \boldsymbol{\nabla} \cdot \boldsymbol{u}^*, \tag{12f}$$

7. Corrector step

$$\boldsymbol{u}^k = \boldsymbol{u}^* - 2\alpha_k \Delta t \boldsymbol{\nabla} \phi, \tag{12g}$$

8. Compute the real pressure P from the pseudopressure ϕ

$$P^k = P^{k-1} + \phi - \alpha_k \Delta t \nu_f \nabla^2 \phi. \tag{12h}$$

In this method, α_k, γ_k, and ζ_k for $k = 1, 2, 3$ are Runge Kutta coefficients in Rai and Moin.[43] To couple the interactions between fluid and markers, $\boldsymbol{u}_p^d(\boldsymbol{X}_l)$ is computed from the discrete equations that govern the linear and angular particle momentum

$$\boldsymbol{u}_p^k = \boldsymbol{u}_p^{k-1} + 2\alpha_k \Delta t \frac{\rho_p \rho_f}{m_p(\rho_p - \rho_f)} \left[-\sum_l^{N_l} \boldsymbol{F}(\boldsymbol{X}_l) \Delta V_l + \frac{\boldsymbol{F}_{c,p}}{\rho_f} \right] + 2\alpha_k \Delta t \mathbf{g}, \tag{13a}$$

$$\boldsymbol{\omega}_p^k = \boldsymbol{\omega}_p^{k-1} + 2\alpha_k \Delta t \frac{\rho_p \rho_f}{I_p(\rho_p - \rho_f)} \left[-\sum_l^{N_l} \left(\boldsymbol{X}_l - \boldsymbol{x}_p^{k-1} \right) \times \boldsymbol{F}(\boldsymbol{X}_l) \Delta V_l + \boldsymbol{T}_{c,p} \right], \tag{13b}$$

$$\boldsymbol{x}_p^k = \boldsymbol{x}_p^{k-1} + \alpha_k \Delta t \left(\boldsymbol{u}_p^k + \boldsymbol{u}_p^{k-1} \right), \tag{13c}$$

$$\boldsymbol{u}_p^d(\boldsymbol{X}_l) = \boldsymbol{u}_p^k + \boldsymbol{\omega}_p^k \times \left(\boldsymbol{X}_l - \boldsymbol{x}_p^k \right). \tag{13d}$$

In what follows, Equations (12a)–(12h) are referred to as the original fluid solver and Equations (13a)–(13d) are referred to as the original particle solver. Thorough validations with both two- and three-dimensional[14,44,45] cases have been conducted to demonstrate that the original direct-forcing IBM is second-order accurate in both time and space to simulate particle-flow interactions.

2.2 | Finite-volume, IBM on a collocated grid

2.2.1 | Collocated fluid solver: Pressure-momentum coupling on a collocated grid

Instead of a finite-difference Navier–Stokes solver on a staggered grid, Equation (1) is discretized on a collocated grid using the finite-volume approach. The main disadvantage of collocated grids is the lack of coupling between momentum and pressure when solving the pressure Poisson equation as defined in Equation (12f). A collocated grid results in wider stencils to compute the Laplacian term $\nabla^2 \phi$ leading to decoupling between velocity and pressure and only an approximately divergence-free flow.[34,46] This decoupling results in a "checkerboard" pressure field and grid-scale oscillations in the velocity field.[46,47]

A common method to eliminate the checkerboarding is to use a staggered formulation of the pressure Poisson equation by interpolating quantities from cell centers to faces.[34,46] Here, we adopt the method of Zang et al.[34] in using interpolated face values of the velocity field $\boldsymbol{u}_f$ to solve the pressure Poisson equation and then correcting $\boldsymbol{u}_f$ and $\boldsymbol{u}$ separately with the following steps:

1. Interpolate center to face values

$$\boldsymbol{u}_f^* = \mathcal{I}(\boldsymbol{u}^*), \tag{14a}$$

2. Pressure Poisson equation to obtain the pseudopressure

$$\nabla^2 \phi = \frac{1}{2\alpha_k \Delta t} \boldsymbol{\nabla} \cdot \boldsymbol{u}_f^*, \tag{14b}$$

3. Corrector step for the cell-centered velocity

$$\boldsymbol{u}^{k} = \boldsymbol{u}^{*} - 2\alpha_{k}\Delta t\boldsymbol{\nabla}\phi, \tag{14c}$$

4. Corrector step for the face-centered velocity

$$\boldsymbol{u}_{f}^{k} = \boldsymbol{u}_{f}^{*} - 2\alpha_{k}\Delta t\left(\boldsymbol{\nabla}\phi\right)_{f}. \tag{14d}$$

In step 1, $\mathcal{I}(\cdot)$ is an interpolation scheme used to obtain $\boldsymbol{u}_f$ from $\boldsymbol{u}$. To ensure overall second-order accuracy, the interpolation scheme used in our modified method is a second-order accurate linear interpolation.

2.2.2 | Collocated fluid solver: Outer forcing

In the original direct-forcing IBM, no-slip boundary conditions are approximate and incur an error of $\epsilon = |\boldsymbol{u}_p^d(\boldsymbol{X}_l) - \widetilde{\boldsymbol{U}}(\boldsymbol{X}_l)|$ due to the explicit formulation of $\boldsymbol{f}_{\mathrm{IBM}}$ in calculating the predictor velocity field $\boldsymbol{u}^*$.[24] Following Kempe and Fröhlich,[24] the difference between the explicit and implicit $\boldsymbol{f}_{\mathrm{IBM}}$ is defined as

$$\Delta\boldsymbol{f}_{IBM} = -\Delta t\left[\alpha_k\nu_f\nabla^2\left(\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{f}_{\mathrm{IBM}}\right) + 2\alpha_k^2\nu_f^2\nabla^2\left(\nabla^2\boldsymbol{u}^k\right)\right]. \tag{15}$$

Equation (15) implies first-order error with respect to the time-step size Δt. Therefore, sufficiently small Δt is required to achieve satisfactory results which leads to an increase in computational cost. To remove the limitation of Δt imposed by this error, outer forcing loops proposed by various researchers[24,48] are implemented with n_f steps as follows:

for $m = 1, n_f$

1. Project Eulerian grid velocities onto the Lagrangian marker

$$\boldsymbol{U}^{m-1}\left(\boldsymbol{X}_{l,n}\right) = \sum_{i,j,k}^{N_i,N_j,N_k}\boldsymbol{u}^{m-1}\left(\boldsymbol{x}_{i,j,k}\right)\delta_{h,3D}\left(\boldsymbol{x}_{i,j,k} - \boldsymbol{X}_{l,n}\right)h^3, \tag{16a}$$

2. Determine the Lagrangian marker force to enforce the no-slip condition

$$\boldsymbol{F}^{m-1}\left(\boldsymbol{X}_{l,n}\right) = \frac{\boldsymbol{U}^d\left(\boldsymbol{X}_{l,n}\right) - \boldsymbol{U}_{l,n}^{m-1}}{2\alpha_k\Delta t}, \tag{16b}$$

3. Interpolate Lagrangian marker force onto the Eulerian grid

$$\boldsymbol{f}_{\mathrm{IBM}}^{m-1}(\boldsymbol{x}_{i,j,k}) = \sum_{n,l}^{N_p,N_l}\boldsymbol{F}_{l,n}^{m-1}\delta_{h,3D}\left(\boldsymbol{x}_{i,j,k} - \boldsymbol{X}_{l,n}\right)\Delta V_{l,n}, \tag{16c}$$

4. Update Eulerian velocity with the computed force

$$\boldsymbol{u}^{m} = \boldsymbol{u}^{m-1} + 2\alpha_k\Delta t\boldsymbol{f}^{m-1}, \tag{16d}$$

$$\boldsymbol{u}^{m-1} = \boldsymbol{u}^{m}, \tag{16e}$$

end

In principle, Equation (16d) must be solved implicitly with the viscous term which would require a matrix inversion. However, since the Runge–Kutta time step is relatively small, an explicit update is a valid approximation. Although this method alleviates the time-step constraint, the number of outer forcing loops n_f is a tuning parameter. Kempe and

Fröhlich[24] reported that large n_f will eventually eliminate the error, although $n_f = 3$ represents a good tradeoff between computational cost and accuracy. However, Biegert[49] stated that results can be oscillatory and negatively impact the collision accuracy. Biegert et al.[16] reported $n_f = 1$ is sufficient to obtain accurate results for a single sphere settling in an approximately unbounded (periodic) domain. In Section 3.3, we show that n_f has a strong effect on collision model accuracy and hence must be calibrated based on both the fluid-particle interaction and collision models in our collocated method.

2.2.3 | Collocated fluid solver: Three- and four-point Dirac delta function

In the original direct forcing IBM on a staggered grid, the three-point regularized Dirac delta function (Equation 9) is used. However, on a collocated grid, a four-point function that ensures both odd and even grid cells receive the same forcing may be needed to reduce oscillations that occur with the three-point function. To enforce this constraint, the four-point function must satisfy

$$\sum_{i\text{ even}} \phi_4\left(r_i\right) = \sum_{i\text{ odd}} \phi_4\left(r_i\right) = \frac{1}{2}, \tag{17}$$

while the three-point function (Equation 10) only enforces

$$\sum_{i} \phi_3\left(r_i\right) = 1, \tag{18}$$

where i is an integer indicating one of the cells on the Eulerian grid. Uhlmann[45] suggests that a collocated formulation should use the four-point function proposed by Peskin[13] to reduce oscillations that occur with the three-point function. In the four-point function,

$$\phi_4\left(r\right) = \begin{cases} \frac{1}{8}\left(5 + 2r - \sqrt{-7 - 12r - 4r^2}\right), & -2 \le r \le 1 \\ \frac{1}{8}\left(3 + 2r + \sqrt{1 - 4r - 4r^2}\right), & -1 \le r \le 0 \\ \frac{1}{8}\left(3 - 2r + \sqrt{1 + 4r - 4r^2}\right), & 0 \le r \le 1 \\ \frac{1}{8}\left(5 - 2r - \sqrt{-7 + 12r - 4r^2}\right), & 1 \le r \le 2 \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

The three-dimensional, four-point regularized Dirac-delta function is then given by

$$\delta^{4p}_{h,3D}\left(\boldsymbol{x} - \boldsymbol{X}_l\right) = \delta^{4p}_{h,x}\left(x - X_l\right) \delta^{4p}_{h,y}\left(y - Y_l\right) \delta^{4p}_{h,z}\left(z - Z_l\right), \tag{20}$$

where

$$\delta^{4p}_{h,x}\left(x - X_l\right) = \frac{1}{h}\phi_4\left(r\right). \tag{21}$$

In Section 3.2, we compare the effects of the three- and four-point regularized Dirac delta functions on the accuracy of direct-forcing IBM in our collocated grid approach.

2.2.4 | Collocated fluid solver: Triply-periodic boundary conditions

To simulate particle suspensions in a triply periodic domain, the compatibility condition ($\int_\Omega \boldsymbol{\nabla} P \ \mathrm{d}\Omega = 0$) must be satisfied.[14] As demonstrated by Höfler and Schwarzer,[50] by assuming periodicity and zero net acceleration and decomposing the pressure gradient with

$$\boldsymbol{\nabla} P = \boldsymbol{\nabla} P_{\text{aperiodic}} + \boldsymbol{\nabla} P_{\text{periodic}}, \tag{22}$$

where $\nabla P_{\text{aperiodic}}$ and $\nabla P_{\text{periodic}}$ are the aperiodic and periodic components of the pressure gradient, integrating the Navier–Stokes equation (1) over the computational domain volume Ω gives

$$\nabla P_{\text{aperiodic}} = \frac{1}{V_d}\int_{\Omega} \boldsymbol{f}_{\mathbf{IBM}}\ \mathrm{d}V = \bar{\boldsymbol{f}}_{\text{IBM}}, \tag{23}$$

where $\bar{\boldsymbol{f}}_{\text{IBM}}$ is the volume-averaged direct-forcing vector and Ω is the computational domain with volume V_d. To ensure compatibility, $\bar{\boldsymbol{f}}_{\text{IBM}}$ must be subtracted from Equation (1) for triply-periodic cases. Since the aperiodic pressure gradient arises from the buoyancy force due to the particles, the force can be computed as the submerged weight of the N_p particles in the system with

$$\bar{\boldsymbol{f}}_{e,sw} = -\sum_{n}^{N_p} \frac{\pi}{6} d_{p,n}^3 \left(\frac{\rho_{p,n}}{\rho_f} - 1\right) \mathbf{g}, \tag{24a}$$

where $d_{p,n}$ and $\rho_{p,n}$ are, respectively, the diameter and density of particle n. Alternatively, it can be directly computed discretely as the average direct IBM force over the Eulerian grid cells in the domain with

$$\bar{\boldsymbol{f}}_{\text{IBM}} = -\frac{1}{N_i N_j N_k} \sum_{i,j,k}^{N_i,N_j,N_k} \boldsymbol{f}_{\text{IBM},ijk}, \tag{24b}$$

which is the average discrete IBM force added to the system due to the presence of particles. Both equations yield the same result but Equation (24b) incurs slightly more computational cost than Equation (24a) since Equation (24a) is computed only once. Equation (24b) is adopted for our proposed method so as to not restrict to spherical particles for possible future extension.

Including the modifications in this section, the modified finite-volume fluid solver for direct-forcing IBM on a collocated grid is given by

1. Predictor step without direct forcing

$$\begin{aligned}\frac{\tilde{\boldsymbol{u}} - \boldsymbol{u}^{k-1}}{\Delta t} &= 2\alpha_k \nu_f \nabla^2 \boldsymbol{u}^{k-1} - 2\alpha_k \nabla P^{k-1} \\ &\quad - \gamma_k (\boldsymbol{u}\cdot\nabla\boldsymbol{u})^{k-1} - \zeta_k (\boldsymbol{u}\cdot\nabla\boldsymbol{u})^{k-2},\end{aligned} \tag{25a}$$

   where $\boldsymbol{u}\cdot\nabla\boldsymbol{u}$ and $\nabla^2\boldsymbol{u}$ are evaluated with second-order accurate finite differences on the collocated grid.
2. Project the predicted Eulerian grid velocities onto the Lagrangian marker

$$\tilde{\boldsymbol{U}}\left(\boldsymbol{X}_{l,n}\right) = \sum_{i,j,k}^{N_i,N_j,N_k} \tilde{\boldsymbol{u}}\left(\boldsymbol{x}_{i,j,k}\right) \delta_{h,3D}\left(\boldsymbol{x}_{i,j,k} - \boldsymbol{X}_{l,n}\right) h^3, \tag{25b}$$

3. Determine the Lagrangian marker force to enforce the no-slip condition

$$\boldsymbol{F}\left(\boldsymbol{X}_{l,n}\right) = \frac{\boldsymbol{u}_p^d\left(\boldsymbol{X}_{l,n}\right) - \tilde{\boldsymbol{U}}\left(\boldsymbol{X}_{l,n}\right)}{2\alpha_k \Delta t}, \tag{25c}$$

4. Interpolate Lagrangian marker force back onto the Eulerian grid

$$\boldsymbol{f}_{\text{IBM}}\left(\boldsymbol{x}_{i,j,k}\right) = \sum_{n,l}^{N_p,N_l} \boldsymbol{F}\left(\boldsymbol{X}_{l,n}\right) \delta_{h,3D}\left(\boldsymbol{x}_{i,j,k} - \boldsymbol{X}_{l,n}\right) \Delta V_{l,n}. \tag{25d}$$

   The three- and four-point delta functions are compared in Section 3.2.

5. Predictor step with direct forcing

$$\frac{\boldsymbol{u}^{*}-\widetilde{\boldsymbol{u}}}{\Delta t}=2\alpha_{k}\left(\boldsymbol{f}_{\mathrm{IBM}}-\bar{\boldsymbol{f}}_{\mathrm{IBM}}\right)-\alpha_{k}\nu_{f}\nabla^{2}\boldsymbol{u}^{k-1} \\ +\alpha_{k}\nu_{f}\nabla^{2}\boldsymbol{u}^{*}+2\alpha_{k}\theta\boldsymbol{\nabla}P^{k-1}, \tag{25e}$$

where $\bar{\boldsymbol{f}}_{\mathrm{IBM}}$ is the submerged weight of the particles from Equation (24b) and only non-zero for triply-periodic cases.

6. Interpolate the cell-centered velocities onto the faces

$$\boldsymbol{u}_{f}^{*}=\mathcal{I}\left(\boldsymbol{u}^{*}\right), \tag{25f}$$

7. Solve the pressure Poisson equation to obtain the pseudopressure ϕ

$$\nabla^{2}\phi=\frac{1}{2\alpha_{k}\Delta t}\boldsymbol{\nabla}\cdot\boldsymbol{u}_{f}^{*}, \tag{25g}$$

8. Corrector step to obtain the cell- and face-centered quantities

$$\boldsymbol{u}^{k}=\boldsymbol{u}^{*}-2\alpha_{k}\Delta t\boldsymbol{\nabla}\phi, \tag{25h}$$

$$\boldsymbol{u}_{f}^{k}=\boldsymbol{u}_{f}^{*}-2\alpha_{k}\Delta t\boldsymbol{\nabla}\phi_{f}, \tag{25i}$$

9. Compute the full pressure P using the pseudopressure ϕ

$$P^{k}=(1-\theta)\,P^{k-1}+\theta\left(\phi-\alpha_{k}\Delta t\nu_{f}\nabla^{2}\phi\right). \tag{25j}$$

Here, $\theta=0$ for pressure projection and $\theta=1$ for pressure correction. Compared to the staggered formulation, Equations (25a)–(25d) are formulated on a collocated grid. Pressure and momentum have been coupled through Equations (25e)–(25j). The original direct-forcing IBM[14] was thoroughly validated with test cases involving fluid-particle interactions. However, in particle suspensions, particle-particle and particle-wall collisions are inevitable. When particles come into contact with one another or a wall, two problems arise. First, the Lagrangian marker cells overlap, rendering the Dirac delta function invalid. Second, the direct-forcing IBM cannot resolve the flow in the small gaps between the particles or particle and wall. To resolve these issues, we adapt the collision models proposed by Biegert et al.[16] for a collocated grid. In this approach, the collision force $\boldsymbol{F}_{c,p}$ and torque $\boldsymbol{T}_{c,p}$ imposed on particle p are given by

$$\boldsymbol{F}_{c,p}=\sum_{p,q\neq p}^{N_{p}}\left(\boldsymbol{F}_{n,pq}+\boldsymbol{F}_{t,pq}\right)+\boldsymbol{F}_{n,pw}+\boldsymbol{F}_{t,pw}, \tag{26a}$$

$$\boldsymbol{T}_{c,p}=\sum_{p,q\neq p}^{N_{p}}R_{pq,cp}\boldsymbol{n}_{pq}\times\boldsymbol{F}_{t,pq}+R_{pw,cp}\boldsymbol{n}_{pw}\times\boldsymbol{F}_{t,pw}, \tag{26b}$$

where $\boldsymbol{F}_{n,pq}$ and $\boldsymbol{F}_{t,pq}$ are the normal and tangential collision forces between particle p and q, $\boldsymbol{F}_{n,pw}$ and $\boldsymbol{F}_{t,pw}$ are the normal and tangential collision forces between particle p and a wall, $\boldsymbol{n}_{pq}$ is the vector normal to the plane of contact between particles p and q, $\boldsymbol{n}_{pw}$ is the vector normal to the wall at the point of contact with particle q, $R_{pq,cp}=0.5\|\left(\|\boldsymbol{x}_{q}-\boldsymbol{x}_{p}\|+R_{p}-R_{q}\right)\boldsymbol{n}_{pq}\|$ is the effective radius between particle p and q and $R_{pw,cp}=\|\boldsymbol{x}_{w}-\boldsymbol{x}_{p}\|$ is the effective radius between particle p and wall w. The normal and tangential collisional forces on particle p are defined as

$$\boldsymbol{F}_{n}=\begin{cases}0, & \zeta_{n}\geq\epsilon_{sep}\\ \boldsymbol{F}_{n,lub}, & 0\leq\zeta_{n}\leq\epsilon_{sep},\\ \boldsymbol{F}_{n,con}, & \zeta_{n}\leq 0,\end{cases} \tag{27a}$$

$$\boldsymbol{F}_{t}=\begin{cases}0, & \zeta_{n}\geq\epsilon_{sep}\\ \boldsymbol{F}_{t,con}, & \zeta_{n}\leq 0,\end{cases} \tag{27b}$$

where $\boldsymbol{F}_{n,lub}$ is the lubrication force, $\boldsymbol{F}_{n,con}$ is the normal contact force, ζ_n is the separation distance between the surfaces and $\epsilon_{sep} = 1.5h$ for the three-point Dirac-delta function and $\epsilon_{sep} = 2h$ for the four-point Dirac-delta function.

2.2.5 | Collision models: Disabling the Lagrangian markers

To resolve issues associated with overlapping Eulerian grid points, we adopt an approach similar to Kempe and Fröhlich[24] and Biegert et al.[16] by excluding Lagrangian markers in calculating $\boldsymbol{f}_{\text{IBM}}$ when the distance between a particle and wall satisfies $\zeta_{n,pw} < \epsilon_{sep}$ or the distance between particles p and q satisfies $\zeta_{n,pq} < 2\epsilon_{sep}$. In addition, for particle-particle collisions, instead of excluding Lagrangian markers for both particles as in Kempe and Fröhlich[24] and Biegert et al.,[16] we exclude the Lagrangian markers from just one particle, chosen at random. This avoids the scenario where the particle that is surrounded by many neighbors has none of its Lagrangian markers included in the calculation and better enforces no-slip condition on the particle surfaces.

2.2.6 | Collision models: Lubrication model

When the separation distance ζ_n between particles is smaller than the threshold separation distance ϵ_{sep}, direct-forcing IBM can no longer resolve the flow. Therefore, a lubrication model is used to model the force exerted by the fluid on the particles. We adopt the analytical lubrication model by Cox and Brenner[51] and modified by Biegert et al.,[16] in which the lubrication force is given by

$$\boldsymbol{F}_{n,lub} = -\frac{6\pi\rho_f \nu_f R_{\text{eff}}^2 \psi\left(\zeta_n\right)}{\max\left(\zeta_{\min}, \zeta_n\right)} \boldsymbol{g}_{n,cp}, \tag{28}$$

where $R_{\text{eff}} = R_p R_q/(R_p + R_q)$ is the effective radius that is defined based on the particles p and q, $\boldsymbol{g}_{n,cp}$ is the normal component of the relative velocity of the particle surface at the contact point and $\zeta_{\min}$ is the minimum separation distance to prevent a singularity as $\zeta_n \to 0$.[16] In Biegert et al.,[16] $\psi\left(\zeta_n\right) = \zeta_n$ was used. However, this formulation results in a discontinuity at the interface where $\boldsymbol{F}_{n,lub} \neq 0$ (Figure 2). To avoid the discontinuity, we introduce a function $\psi\left(\zeta_n\right) = 0.5\text{erf}\left(6 - C\max\left(\zeta_{\min}, \zeta_n\right)/h\right) + 0.5$ to enforce a continuous force when particles come close to one another where $C = 5.5$ or 4 for three and four-point function respectively.

2.2.7 | Collision models: Normal contact model

To account for the normal contact force during collisions, we implemented the approach by Biegert et al.[16] who employ the adaptive collision time model (ACTM) proposed by Kempe and Fröhlich.[27] The idea behind ACTM is to derive an optimized stiffness coefficient k_n and damping coefficient d_n to achieve the desired dry restitution coefficient e_{dry} over a collision time T_c, as discussed below. The normal contact force $F_{n,con}$ is defined as

$$\boldsymbol{F}_{n,con} = -k_n|\zeta_n|^{3/2}\boldsymbol{n} - d_n\boldsymbol{g}_{n,cp}, \tag{29}$$

where k_n and d_n are the optimized stiffness and damping coefficients. Equation (29) describes the collision model based on the contact theory of Hertz.[27] In the past, k_n and d_n were typically chosen based on material properties. However, the chosen k_n is usually large, making the equations very stiff. Therefore, the time step size Δt must be based on the collision model in which the collision time T_c is much smaller than Δt for the flow calculation. To overcome this problem, instead of fixing k_n and d_n, Kempe and Fröhlich[27] proposed to fix e_{dry} and T_c by dynamically optimizing k_n and d_n. e_{dry} is a parameter based on the material property of the particle and T_c is a tuning parameter. Large T_c will lead to extensive overlap between colliding particles, making collisions unrealistic, while small T_c will increase the stiffness of the equation. Based on the studies conducted by Kempe and Fröhlich,[27] $T_c = 10\Delta t$ represents a good balance between accuracy and computational cost. To obtain k_n and d_n, the nonlinear ordinary differential equations representing the interparticle spacing are solved, such that

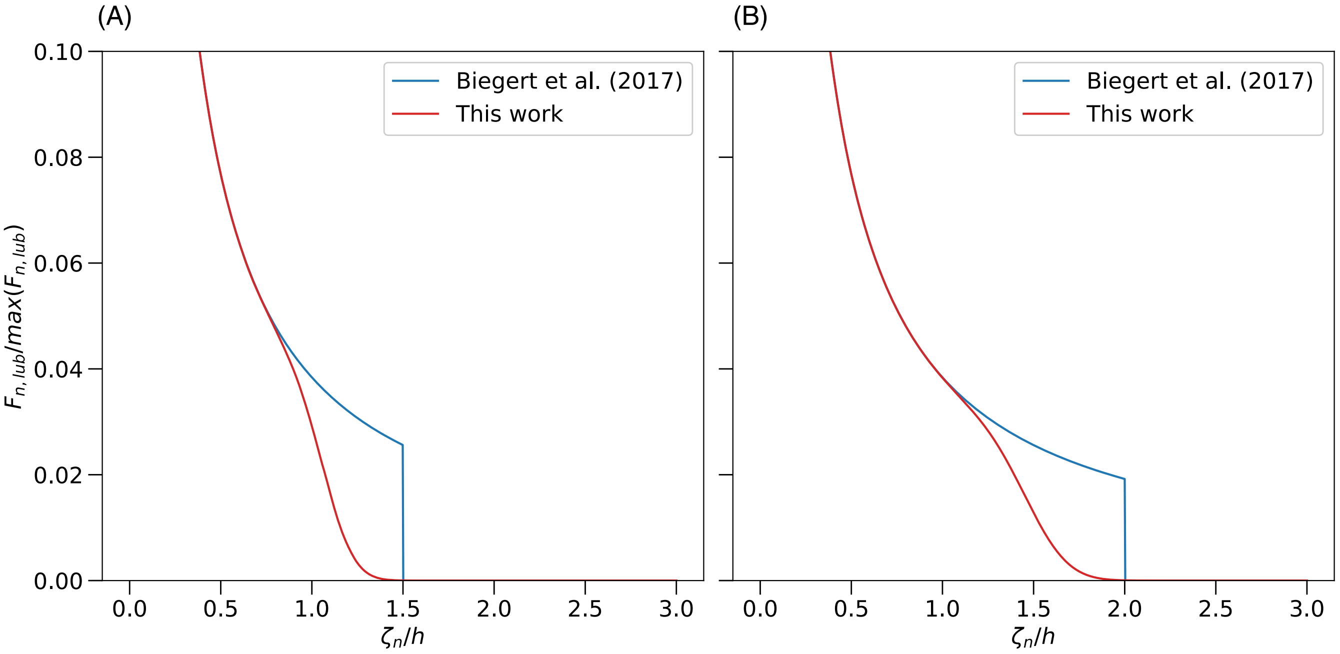

FIGURE 2 Lubrication force as a function of separation distance between particle p and q for (A) three- and (B) four-point function [Colour figure can be viewed at wileyonlinelibrary.com]

$$m_{\text{eff}}\frac{\mathrm{d}^2\zeta_n}{\mathrm{d}t^2} + d_n\frac{\mathrm{d}\zeta_n}{\mathrm{d}t} + k_n\zeta_n^{3/2} = 0, \tag{30a}$$

$$\frac{\mathrm{d}\zeta_n}{\mathrm{d}t} = -\boldsymbol{g}_{n,cp}\cdot\boldsymbol{n}, \tag{30b}$$

where $m_{\text{eff}} = m_p m_q/(m_p + m_q)$ is the effective mass accounting for polydisperse particles, and the equations are subject to two conditions at $t = T_c$, $\zeta_n(T_c) = 0$ and $g_{n,cp}(T_c) = e_{dry}\boldsymbol{g}_{n,cp}\cdot\boldsymbol{n}$. Kempe and Fröhlich[27] used the Newton–Ralphson method to obtain a solution to Equation (30a) while Ray et al.[52] developed an analytical approach. The method of Ray et al.[52] incurs less than 1.3% error in the rebound velocity when $e_{dry} > 0.7$. In our simulations, since $e_{dry} > 0.9$ is typically used, we adopt their approach due to its low computational cost and ease of implementation.

A potential issue with ACTM as pointed out by Biegert et al.[16] is large k_n for weak collisions. As the impact velocity between collisions $u_{in} \to 0$, the Stokes number defined as

$$St = \frac{u_{in}\rho_p d_p}{9\rho_f \nu_f}, \tag{31}$$

also approaches 0, making $k_n \to \infty$. Both Kempe and Fröhlich[27] and Biegert et al.[16] introduce a critical St_{crit} where u_{in} is based on a prescribed St_{crit}. In our approach, we set $St_{crit} = 5$ following Biegert et al.[16] In addition, Biegert et al.[16] also introduces a threshold $k_{n,grav}$ to prevent extensive overlap when $u_{in,crit}$ is large relative to the particle size and relevant time scales for a low Reynolds number flow. Therefore, k_n is defined as

$$k_n = \begin{cases} k_{n,ACTM}, & u_{in} > u_{in,crit}, \\ \max\left(k_{n,crit}, k_{n,grav}\right), & u_{in} \leq u_{in,crit}, \end{cases} \tag{32}$$

where

$$k_{n,crit} = \frac{m_{\text{eff}}}{\sqrt{u_{in,crit}t_*^5}}, \tag{33}$$

and

$$k_{n,grav} = \max\left(m_p g\left(\epsilon d_p/2\right)^{-3/2}, m_q g\left(\epsilon d_q/2\right)^{-3/2}\right), \tag{34}$$

where $\epsilon = 10^{-3}$.

2.2.8 | Collision models: Tangential contact model

To account for the tangential contact force during collisions, we follow the approach by Biegert et al.[16] who employ the model in a review paper by Thornton et al.[53] This model uses a spring-dashpot model in which $\boldsymbol{F}_{t,con}$ is defined as

$$\boldsymbol{F}_{t,con} = \min\left(\left\|\boldsymbol{F}_{t,dp}\right\|, \left\|\mu_{fri}\boldsymbol{F}_n\right\|\right)\boldsymbol{t}, \tag{35}$$

where μ_{fri} is the coefficient of friction between two surfaces and $\boldsymbol{t} = \boldsymbol{F}_{t,dp}/\left\|\boldsymbol{F}_{t,dp}\right\|$ is the direction vector of the tangential force. $\boldsymbol{F}_{t,dp}$ is defined as

$$\boldsymbol{F}_{t,dp} = -k_t\boldsymbol{\zeta}_t - d_t\boldsymbol{g}_{t,cp}, \tag{36}$$

where k_t and d_t are the stiffness and damping coefficients, $\boldsymbol{g}_{t,cp}$ is the tangential velocity relative to the surface of contact and $\boldsymbol{\zeta}_t$ is the time-cumulative tangential spring displacement defined as

$$\boldsymbol{\zeta}_t = \int_{t_0}^{t} \boldsymbol{g}_{t,cp}(\tau)\ \mathrm{d}\tau, \tag{37}$$

where t_0 is the impact time. An approach similar to ACTM is adopted for k_t and d_t which are calculated dynamically at each time step. The tangential stiffness coefficient is defined as

$$k_t = \frac{2\left(1 - \nu_{poi}\right)}{2 - \nu_{poi}}, \tag{38}$$

and the tangential damping coefficient is defined as

$$d_t = 2\sqrt{m_{\text{eff}}k_t}\frac{-\ln e_{dry}}{\sqrt{\pi^2 + \left(\ln e_{dry}^2\right)}}, \tag{39}$$

where ν_{poi} is Poisson's ratio of the particle material. When compared to the model proposed by Kempe and Fröhlich[24] that enforces slip conditions, this model allows the particles to interact smoothly and stably. To differentiate between rolling/sticking and sliding motions, we adopted the methods by Biegert et al.[16] who employed the formulation of Luding[32] in which $\mu_{fri} = \mu_s$ when particles are sticking ($\left\|\boldsymbol{F}_{t,dp}\right\| < \left\|\mu_{fri}\boldsymbol{F}_n\right\|$) and $\mu_{fri} = \mu_k$ when slipping occurs ($\left\|\boldsymbol{F}_{t,dp}\right\| > \left\|\mu_{fri}\boldsymbol{F}_n\right\|$).

2.2.9 | Collocated particle solver: Direct computation of fluid inertia within the particle

In the original direct-forcing IBM, Uhlmann[14] used a rigid body approximation to calculate the rate-of-change term that describes the effect of fluid inertia within the particle. As a result, Equation (13a) has a singularity at $\rho_p/\rho_f = 1$ when $\rho_p - \rho_f = 0$, and the method becomes unstable when $\rho_p/\rho_f < 1.2$. To resolve this issue, Kempe and Fröhlich[24] adopted a level-set approximation to compute the rate-of-change term directly which eliminates both the singularity and improves the stability related to fluid-particle coupling when $\rho_p/\rho_f > 0.2$. Using second-order midpoint quadrature rules, the integrals in the rate-of-change terms are approximated with

$$\int_{\Omega_p} \boldsymbol{u} \ \mathrm{d}\boldsymbol{V} \approx \sum_{i,j,k}^{N_i,N_j,N_k} \alpha_{i,j,k} \Delta\Omega_{i,j,k} \boldsymbol{u}_{i,j,k}, \tag{40a}$$

$$\int_{\Omega_p} \boldsymbol{r} \times \boldsymbol{u} \ \mathrm{d}\boldsymbol{V} \approx \sum_{i,j,k}^{N_i,N_j,N_k} \alpha_{i,j,k} \Delta\Omega_{i,j,k} \left(\boldsymbol{r} \times \boldsymbol{u}\right), \tag{40b}$$

where $\Delta\Omega$ is the volume of a grid cell and $\alpha_{i,j,k}$ is the volume fraction of the cell with indices i, j, k and is defined as

$$\alpha_{i,j,k} = \frac{\Delta\Omega^p_{i,j,k}}{\Delta\Omega_{i,j,k}}, \tag{41}$$

where $\Delta\Omega^p_{i,j,k}$ is the volume of the cell occupied by particle p. $\alpha_{i,j,k}$ can be calculated with the level-set approximation

$$\alpha_{i,j,k} = \frac{\sum_m^8 - \phi_m H\left(-\phi_m\right)}{\sum_m^8 |\phi_m|}, \tag{42}$$

where $H\left(\cdot\right)$ is the Heaviside function

$$H\left(-\phi\right) = \begin{cases} 1, & \phi \leq 0, \\ 0, & \phi > 0, \end{cases} \tag{43}$$

and ϕ is the distance from the corner of each grid cell to the center of the particle and $m = 1, 2, \dots, 8$ is the index of the eight corners of a Cartesian grid cell.

2.2.10 | Collocated particle solver: High-order time integration scheme and sub-stepping

In the original direct-forcing IBM, the linear and angular momentum equations for the particle motion are integrated in time with the first-order forward Euler scheme or the second-order Crank–Nicolson scheme. Various researchers[14,24] have shown that these discretizations can produce accurate results of fluid-particle interactions. However, Biegert et al.[16] demonstrated that lower-order schemes do not produce accurate particle rebound velocities. As a result, a collision time of $T_c = 1000\Delta t$ is required to reduce the error in the rebound velocity to 0.1%. Therefore, we followed the approach by Biegert et al.[16] by adopting a higher-order time-stepping scheme with predictor-corrector steps.

Another issue identified by many researchers[16,30,54] is that a time-step size that accurately resolves fluid-particle interactions may fail to resolve the lubrication force. To overcome this issue, Costa et al.[54] and Biegert et al.[16] proposed sub-iterations for the particle motion solver. Costa et al.[54] conducted a total of 50 sub-iteration by using $\Delta t_{sub} = \Delta t/50$ while Biegert et al.[16] conducted a total of 15 sub-iterations with $\Delta t_{sub} = \Delta t/15$. With the approach by Biegert et al.,[16] each sub-step employs a three-step Runge–Kutta scheme, resulting in a total of 45 iterations. The Runge–Kutta sub-step k of the improved form of the original update given in Equations (13a)–(13d) is given by

$$\frac{\widetilde{\boldsymbol{u}}_p - \boldsymbol{u}_p^{k-1}}{\Delta t} = \frac{1}{m_p}\left(2\alpha_k \boldsymbol{F}^k_{h,p} + \gamma_k \boldsymbol{F}_{c,p}\left(\boldsymbol{x}_p^{k-1}, \boldsymbol{u}_p^{k-1}\right) + \zeta_k \boldsymbol{F}_{c,p}\left(\boldsymbol{x}_p^{k-2}, \boldsymbol{u}_p^{k-2}\right)\right) + 2\alpha_k \boldsymbol{g}', \tag{44a}$$

$$\frac{\widetilde{\boldsymbol{\omega}}_p - \boldsymbol{\omega}_p^{k-1}}{\Delta t} = \frac{1}{I_p}\left(2\alpha_k \boldsymbol{T}^k_{h,p} + \gamma_k \boldsymbol{T}_{c,p}\left(\boldsymbol{x}_p^{k-1}, \boldsymbol{u}_p^{k-1}\right) + \zeta_k \boldsymbol{T}_{c,p}\left(\boldsymbol{x}_p^{k-2}, \boldsymbol{u}_p^{k-2}\right)\right), \tag{44b}$$

$$\frac{\widetilde{\boldsymbol{x}}_p - \boldsymbol{x}_p^{k-1}}{\Delta t} = \alpha_k\left(\widetilde{\boldsymbol{u}}_p + \boldsymbol{u}_p^{k-1}\right), \tag{44c}$$

$$\frac{\boldsymbol{u}_p^k - \boldsymbol{u}_p^{k-1}}{\Delta t} = \frac{1}{m_p}\left(2\alpha_k \boldsymbol{F}^k_{h,p} + \gamma_k \boldsymbol{F}_{c,p}\left(\widetilde{\boldsymbol{x}}_p, \widetilde{\boldsymbol{u}}_p\right) + \zeta_k \boldsymbol{F}_{c,p}\left(\boldsymbol{x}_p^{k-1}, \boldsymbol{u}_p^{k-1}\right)\right) + 2\alpha_k \boldsymbol{g}', \tag{44d}$$

$$\frac{\boldsymbol{\omega}_p^k-\boldsymbol{\omega}_p^{k-1}}{\Delta t}=\frac{1}{I_p}\left(2\alpha_k\boldsymbol{T}_{h,p}^k+\gamma_k\boldsymbol{T}_{c,p}\left(\widetilde{\boldsymbol{x}}_p,\widetilde{\boldsymbol{u}}_p\right)+\zeta_k\boldsymbol{T}_{c,p}\left(\boldsymbol{x}_p^{k-1},\boldsymbol{u}_p^{k-1}\right)\right), \tag{44e}$$

$$\frac{\boldsymbol{x}_p^k-\boldsymbol{x}_p^{k-1}}{\Delta t}=\alpha_k\left(\boldsymbol{u}_p^k+\boldsymbol{u}_p^{k-1}\right), \tag{44f}$$

where

$$\boldsymbol{F}_{h,p}^k=\begin{cases}-\rho_f\sum_l^{N_l}\boldsymbol{F}_l\left(\boldsymbol{X}_l\right)\Delta V_l+\rho_f\left[\frac{\mathrm{d}t}{\mathrm{d}\int_{\Gamma_p}}\boldsymbol{u}\ \mathrm{d}\boldsymbol{x}\right]^k, & \max\left(St\right)>St_{crit}\\ 0, & \max\left(St\right)\le St_{crit}\end{cases} \tag{45}$$

$$\mathbf{g}'=\left(1-\rho_f/\rho_g\right)\mathbf{g}, \tag{46}$$

$$\boldsymbol{T}_{h,p}^k=\begin{cases}-\rho_f\sum_l^{N_l}\left(\boldsymbol{X}_l-\boldsymbol{x}_p^{k-1}\right)\times\boldsymbol{F}\left(\boldsymbol{X}_l\right)\Delta V_l+\rho_f\left[\frac{\mathrm{d}t}{\mathrm{d}\int_{\Gamma_p}}\boldsymbol{r}\times\boldsymbol{u}\ \mathrm{d}\boldsymbol{x}\right]^k, & \max\left(St\right)>St_{crit}\\ 0. & \max\left(St\right)\le St_{crit}\end{cases} \tag{47}$$

3 | RESULTS AND DISCUSSIONS

3.1 | Verification with analytical Taylor Green vortices

Following Uhlmann,[14] to verify the accuracy of the fluid solver of the finite-volume-based IBM on a collocated grid, we computed the errors associated with computing the flow around a particle located in a flow field given by Taylor Green decaying vortices, for which the analytical solution is given by

$$u\left(x,y,t\right)=\sin\left(k_xx\right)\cos\left(k_yy\right)\exp\left(-\left(k_x^2+k_y^2\right)\nu_ft\right), \tag{48a}$$

$$v\left(x,y,t\right)=-\frac{k_x}{k_y}\cos\left(k_xx\right)\sin\left(k_yy\right)\exp\left(-\left(k_x^2+k_y^2\right)\nu_ft\right), \tag{48b}$$

$$P\left(x,y,t\right)=\frac{1}{2}\left(-\sin^2\left(k_xx\right)+\frac{k_x^2}{k_y^2}\cos^2\left(k_yy\right)\right)\exp\left(-2\left(k_x^2+k_y^2\right)\nu_ft\right), \tag{48c}$$

where $k_x=k_y=\pi$ m^{-1} is assumed. The flow is initialized at time $t=0$ with Equation (48) and a two-dimensional circular disk with a diameter $d_p=2$ m and particle-fluid density ratio $s=\rho_p/\rho_f=1$ is located at the center of the computational domain of size $1.5d_p\times1.5d_p$. The kinematic viscosity $\nu_f=0.2$ m^2 s^{-1} is used. The simulation time is 0.5 s with a time step size $\Delta t=0.001$ s. The desired velocity at the disk surface $\boldsymbol{u}_p^d$ is computed with Equation (48), and hence the desired velocity at the Lagrangian markers is the same as the exact Eulerian velocity at those points. Therefore, the particle should not translate or rotate because there is no viscous stress on the particle surface. An accurate IBM method therefore should give the velocity and pressure fields given by Equation (48) since these represent the fluid motion in the absence of a particle.

Figure 3 shows the computed errors in the velocity and pressure fields as a function of the grid resolution. The error is given by the L_∞ norm as

$$\text{error}_\infty=||\alpha-\alpha_{true}||_\infty, \tag{49}$$

where α and α_{true} are the quantities of interest from simulations and Equation (48), respectively. In this verification analysis, the pressure projection method ($\theta=0$ in Equation (25)) and three-point Dirac delta function are used. Similar trends were observed with other combinations (i.e., pressure projection with four-point Dirac delta function). By comparing to the reference line (dashed-line), second-order convergence in both velocity and pressure fields was observed. For cases with and without the particle, second-order convergence was also observed (Figure 3B vs. Figure 3A), showing that the inclusion of interpolation using the discrete delta function does not have an impact on the overall accuracy of the fluid solver.

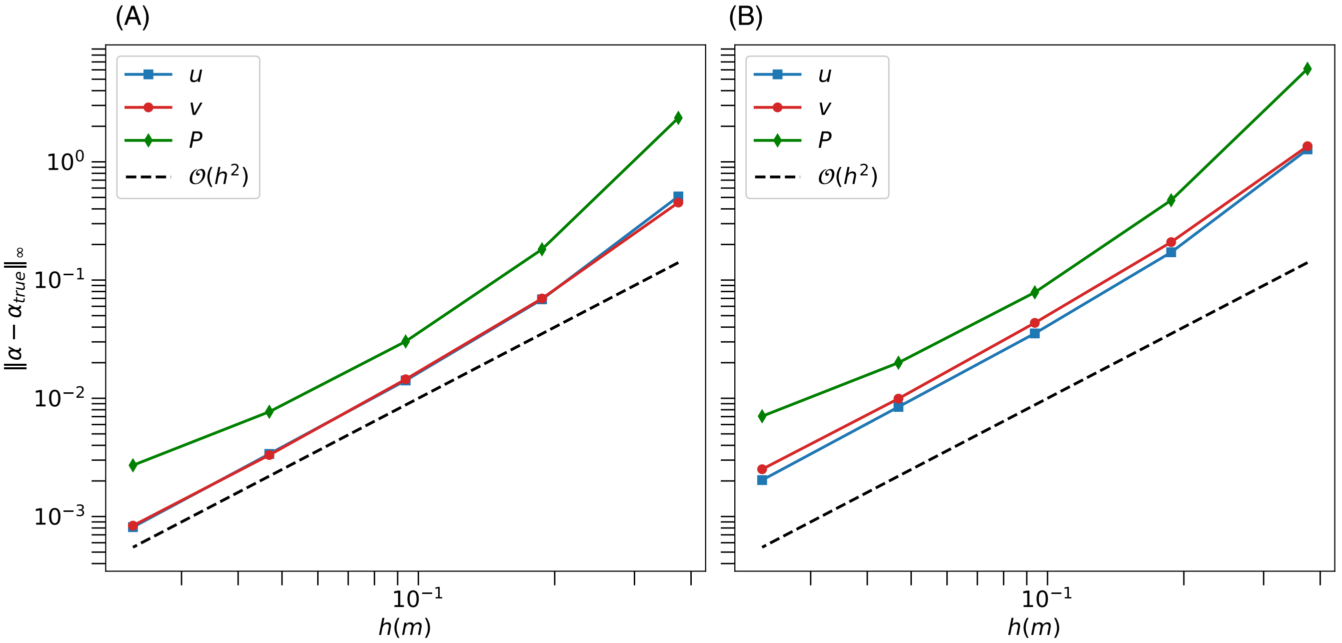

FIGURE 3 Error of velocity and pressure field for simulating two-dimensional decaying vortices. The error is shown as a function of mesh resolution h (A) without and (B) with a particle [Colour figure can be viewed at wileyonlinelibrary.com]

To further verify the accuracy of the discrete delta function, we used the same configuration, except that we enforced a no-slip boundary condition on the particle surface such that $\boldsymbol{u}_p^d = 0$. Since exact solutions are not available due to the presence of the particle, we use the simulation with the highest resolution ($d_p/h = 683$) as a reference solution, where the highest resolution used is much finer than the typical resolution used for IBM methods ($\sim d_p/h = 20$). Both L_2 and L_∞ norms were computed with

$$\text{error}_2 = \|\alpha - \alpha_{d_p/h=683}\|_2, \tag{50a}$$

$$\text{error}_\infty = \|\alpha - \alpha_{d_p/h=683}\|_\infty. \tag{50b}$$

By comparing to the reference lines (dashed-line for $\mathcal{O}(h^2)$ and dashed-dotted-line for $\mathcal{O}(h)$), approximately first-order convergence in both velocity and pressure fields was observed. Overall, the finite-volume-based IBM on a collocated grid is approximately first-order accurate (Figure 4). This is consistent with other similar approaches.[19]

3.2 | Fluid-particle interactions

We simulated the settling of a single particle to validate the accuracy of the fluid-particle interactions using our collocated direct-forcing IBM approach. The primary parameter of interest is the terminal Reynolds number

$$Re_{t,\infty} = \frac{w_{t,\infty} d_p}{\nu_f}, \tag{51}$$

where $w_{t,\infty}$ is the terminal velocity of a single particle in an approximately unbounded (periodic) domain. In the results, the settling velocity and time are normalized by $w_{ref} = \sqrt{g d_p}$ and $t_{ref} = \sqrt{d_p/g}$, respectively, where $g = 9.81$ m s^{-2}. For all simulations in this section, the time step is determined based on a maximum Courant number $C_{\max} = u_{\max}\Delta t/h = 0.4$, where $u_{\max}$ is the maximum magnitude of the fluid velocity vector over the course of the simulation. We also assume $n_f = 2$ outer forcing loops (Section 2.2.2), the three-point Dirac delta function (Equation 10), and the pressure projection scheme ($\theta = 0$ in Equation (25)). Results of a particle settling onto a bottom wall are compared to the experiments of

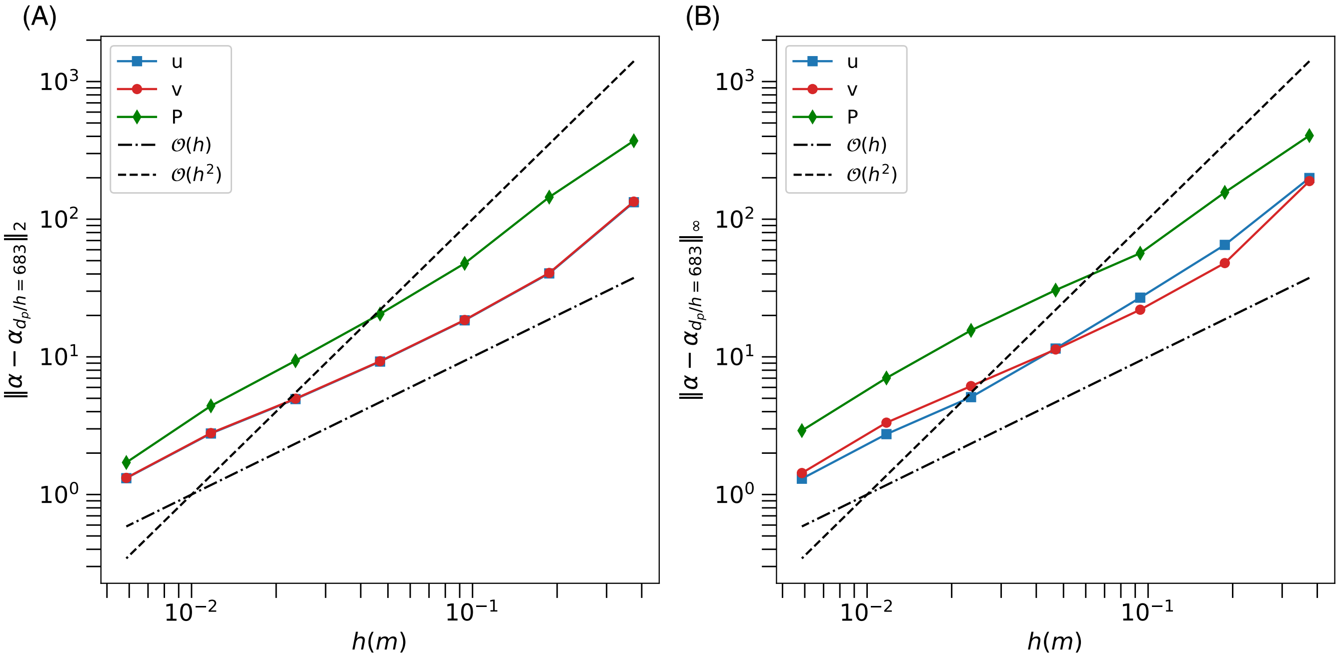

FIGURE 4 Error of velocity and pressure when simulating two-dimensional decaying vortices with a no-slip condition on a particle. Errors are shown as functions of mesh resolution h for (A) L_2 and (B) L_∞ norms [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Simulation parameters and setup for test cases to validate against experiments results by Ten Cate et al.[55] and Mordant and Pinton.[56]

$Re_{t,\infty}$	12	32	41	360
Particle diameter d_p (m)	0.015	0.015	1/6	1/6
Density ratio ρ_p/ρ_f	1.16	1.16	2.56	2.56
Fluid kinematic viscosity ν_f (10^{-4} m^2/s)	1.17	6.04	54.2	10.4
Domain size (m)	0.1 × 0.1 × 0.2	0.1 × 0.1 × 0.2	1.25 × 1.25 × 10	1.25 × 1.25 × 10
Grid resolution d_p/h	14, 19, 29	14, 19, 29	10, 20, 30, 43	10, 20, 30, 43
Particle initial vertical position z_0 (m)	0.13	0.13	9.5	9.5
Boundary conditions	p × p × ns	p × p × ns	p × p × p	p × p × p
Outer forcing loops n_f	2	2	2	2
Delta function	Three-point	Three-point	Three-point	Three-point
Pressure scheme (θ)	Correction	Correction	Correction	Correction
Reference	Ten Cate et al.[55]	Ten Cate et al.[55]	Mordant et al.[56]	Mordant et al.[56]

Note: Boundary conditions are periodic (p) or no-slip (ns).

Ten Cate et al.,[55] while results of a particle settling in an approximately unbounded (periodic) domain are compared to experiments of Mordant and Pinton.[56] Simulation parameters and setup are summarized in Table 1.

Figures 5 and 6 show the settling velocity, w_t, of a single particle settling against a wall and in an approximately unbounded (periodic) domain, respectively, demonstrating the effect of grid resolution d_p/h. As the grid is refined and d_p/h increases, the settling velocity w_t converges monotonically to the experimental values. In Figure 6B, the simulated settling velocity appears to exceed the experimental values, particularly for the higher $Re_{t,\infty}$ case. The experimental results by Mordant and Pinton[56] are ensemble averages of many different instances to average out small discrepancies in particle sizes and unsteadiness in the particle motion, which leads to a small discrepancy between experimental and simulation results and parameters. Similar discrepancies were also found by various authors.[14,19,49] Overall, based on these results,

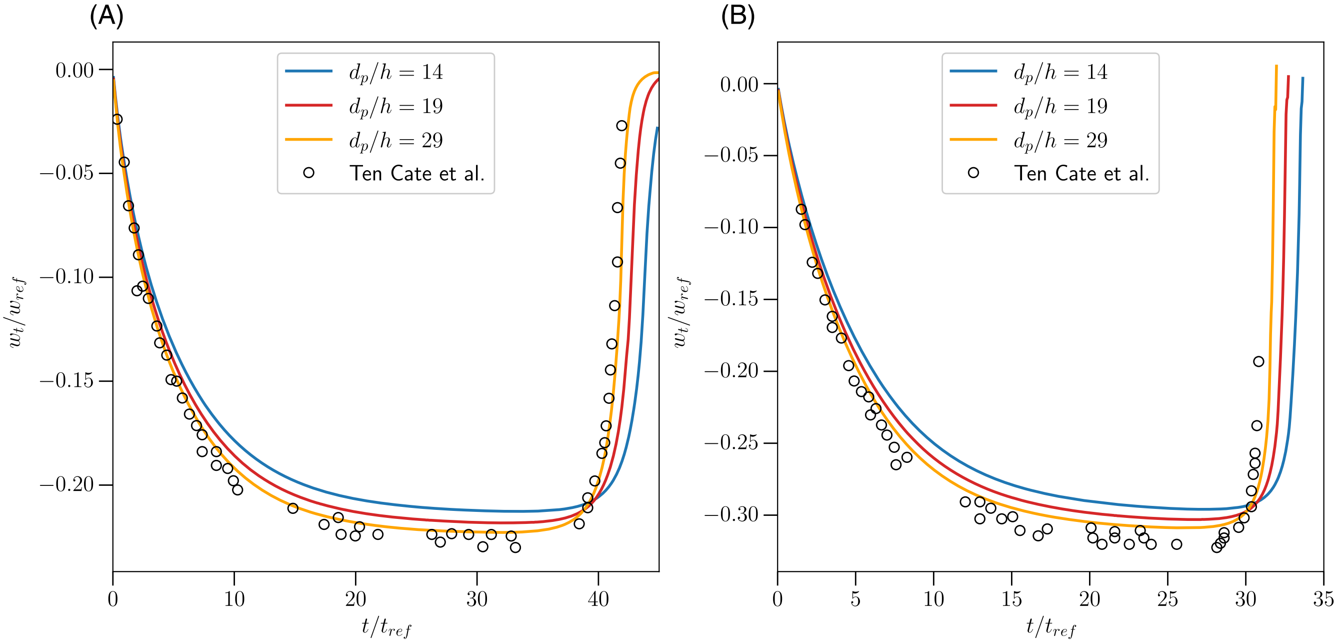

FIGURE 5 Time series of the simulated settling velocity w_t of a single particle settling and coming to rest on a bottom wall with $Re_{t,\infty} = 12$ (A) and 32 (B) and different grid resolutions h used to resolve the particle diameter d_p, compared to published results. [Colour figure can be viewed at wileyonlinelibrary.com]

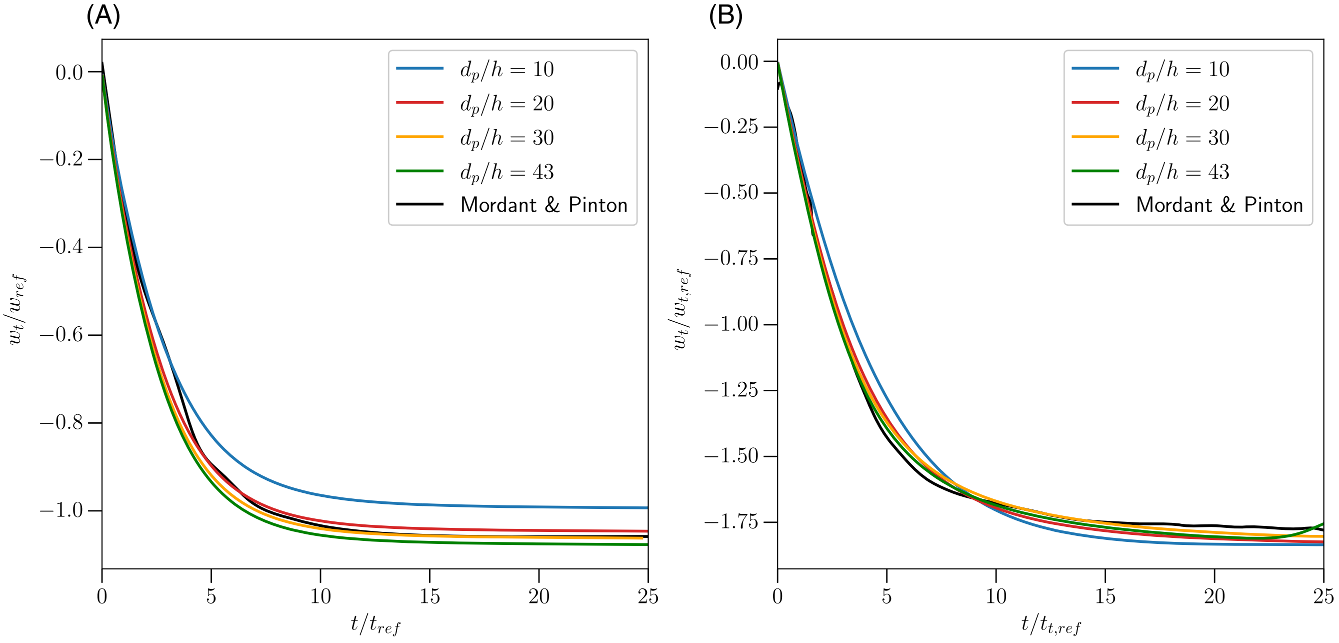

FIGURE 6 Time series of the simulated settling velocity w_t of a single particle in an approximately unbounded (periodic) domain with $Re_{t,\infty} = 41$ (A) and 360 (B) and different grid resolutions h used to resolve the particle diameter d_p, compared to published results. [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Simulation setup to study the effect of the number of outer forcing loops n_f, pressure scheme, and Dirac delta function on simulations of particle settling.

$Re_{t,\infty}$	41	360
Particle diameter d_p (m)	1/6	1/6
Density ratio ρ_p/ρ_f	2.56	2.56
Fluid kinematic viscosity ν_f (10^{-4} m^2/s)	54.2	10.4
Domain size (m × m × m)	1.25 × 1.25 × 10	1.25 × 1.25 × 10
Grid resolution d_p/h	20	20
Particle initial vertical position z_0 (m)	9.5	9.5
Outer forcing loops n_f	0,1,2,3	0,1,2,3
Delta function	Three- or four-point	Three- or four-point
Pressure scheme (θ)	Correction, projection	Correction, projection

Note: Boundary conditions are periodic in all directions.

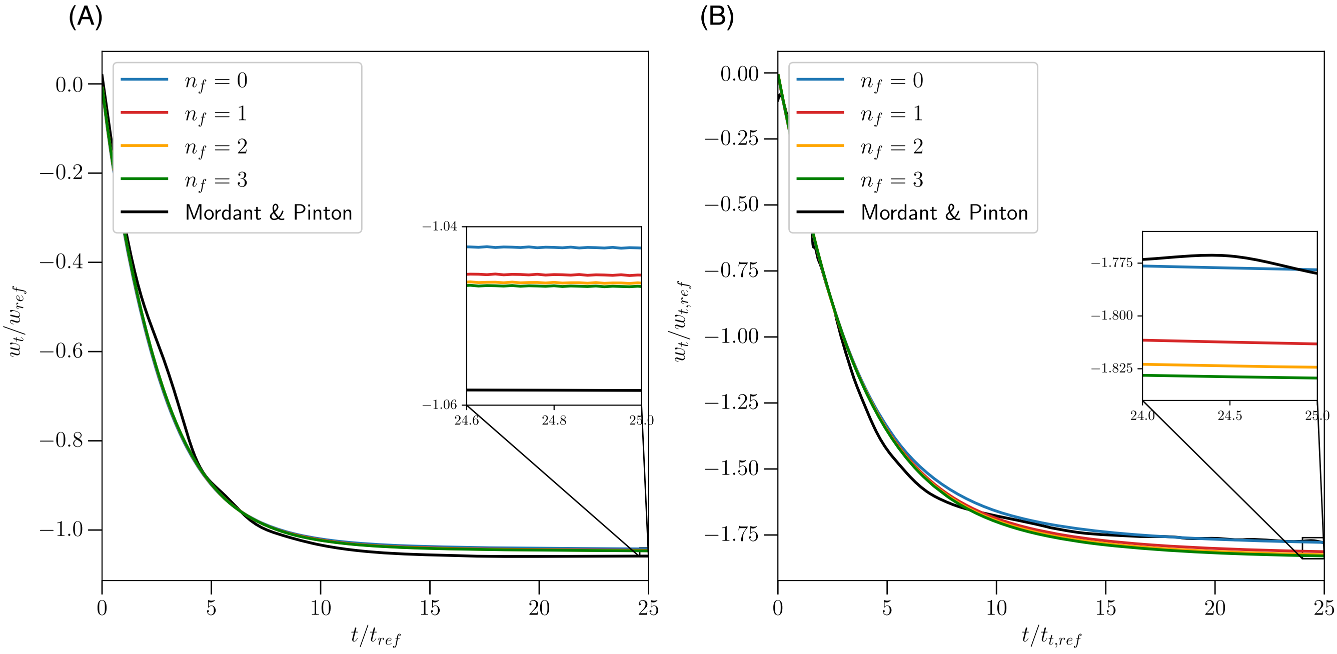

FIGURE 7 Time series of the simulated settling velocity w_t of a single particle with $Re_{t,\infty} = 41$ (A) and 360 (B) and different outer forcing loops n_f to enforce the no-slip condition on the particle surface, compared to published results. [Colour figure can be viewed at wileyonlinelibrary.com]

a grid resolution of $d_p/h = 20$ achieves a good balance between computational cost and accuracy. Therefore, in what follows, we will use $d_p/h = 20$ as the default resolution.

In Section 2.2.2, the number of outer forcing loops n_f was introduced as a tuning parameter. To understand the effect of n_f on the accuracy of fluid-particle interactions, simulations with different n_f are conducted. Table 2 summarizes the simulation parameters and Figure 7 shows the effects of n_f on the settling velocity w_t in an approximately unbounded (periodic) domain. For the low Reynolds number case, w_t converges towards the experimental results as n_f increases (Figure 7A). Similar trends have been also observed by Kempe and Fröhlich[24] and Biegert.[49] When $Re_{t,\infty} = 360$, the improvement from $n_f = 0$ to $n_f = 1$ is significant although further increasing n_f does not demonstrate significant improvement (Figure 7B). The agreement between $n = 0$ and experimental results in Figure 7B is likely to be coincidence since unsteadiness is introduced for higher Reynolds number. The experimental result for $Re_{t,\infty} = 360$ is obtained by averaging numerous repetitions. Based on these simulations, $n_f = 1$ is sufficient to accurately simulate fluid-particle interactions.

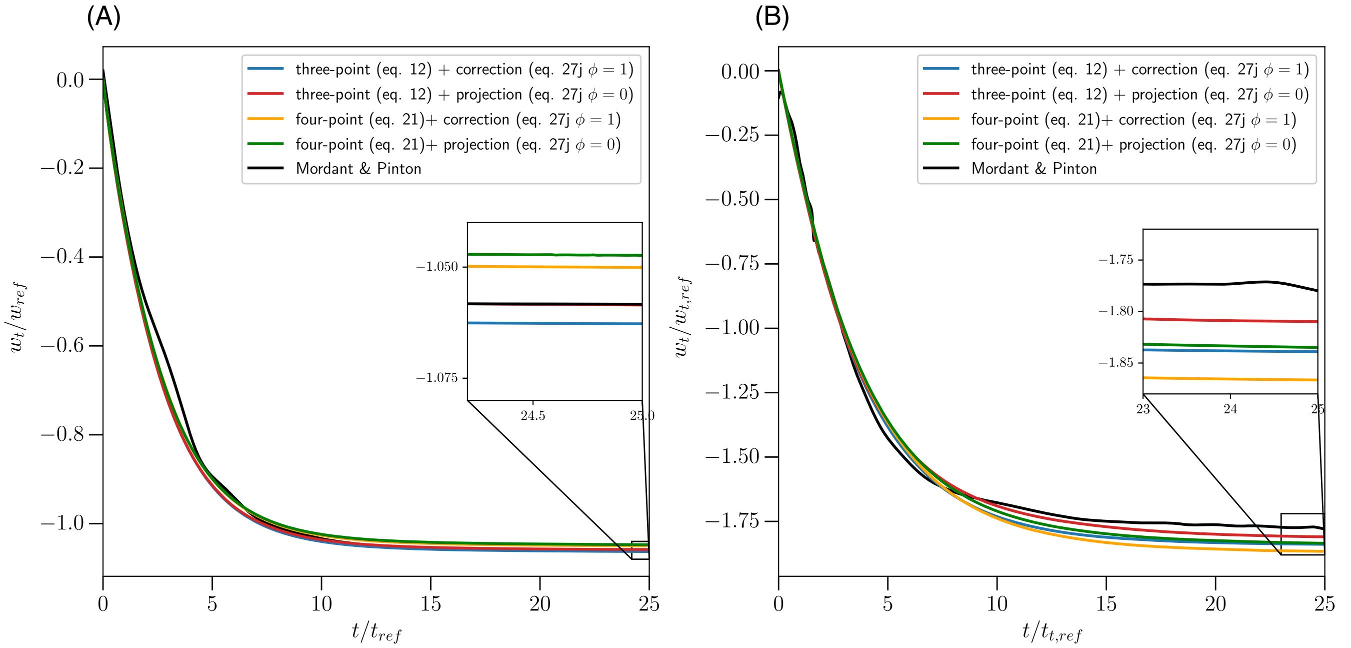

FIGURE 8 Time series of the simulated settling velocity w_t of a single particle with $Re_{t,\infty} = 41$ (A) and 360 (B) and different pressure schemes and delta functions, compared to published results. [Colour figure can be viewed at wileyonlinelibrary.com]

However, the number of outer forcing loops will also affect the accuracy of the collision models. Combined with studies conducted in Section 3.3, $n_f = 2$ is required for accurate collision models while $n_f \geq 1$ is needed to obtain accurate fluid-particle interactions. Therefore, $n_f = 2$ is appropriate to accurately simulate both fluid-particle and particle-particle interactions.

In addition to the grid resolution and n_f, choosing an appropriate interpolation kernel and pressure scheme has been shown to be critical, as discussed in Section 2.2. Figure 8 demonstrates the effects of (1) the pressure correction versus projection scheme (Equation 25k) and (2) the three- versus four-point regularized Dirac delta functions (Equation 10 vs. 19). Although these parameters have been shown to have important effects by past authors (i.e., Armfield and Street[57] compared pressure correction and projection in staggered formulation of fractional step method without IBM and Uhlmann[44] tested three- and four-point Dirac delta functions in a staggered grid without collisions), thorough studies with the collocated IBM method have not been conducted. Our results show that a combination of the three-point delta function and pressure projection scheme produces results that most closely match the experiments for both low and high Reynolds numbers. In what follows, the three-point Dirac delta function and pressure projection scheme will be used.

3.3 | Particle-particle interactions

To validate and calibrate the normal collision model, simulation results are compared to the experiments of Gondret et al.,[58] in which a particle bounces off the wall of a tank. The experiments focus on the effect of the Stokes number on the maximum height of the particle after bouncing. The Stokes number is defined in Equation (31). Table 3 summarizes parameters and setup used in the simulations which were identical to Biegert et al.,[16] and we simulate cases with $St = 27$ and $St = 152$. Two tuning parameters to be calibrated are the number of outer forcing loops n_f and minimum separation distance for the lubrication model $\zeta_{\min}$. The former governs the accuracy of the no-slip condition enforced on the particle surface while the latter determines the extent of deceleration due to lubrication forces. In the staggered direct forcing IBM,[16] $n_f = 1$ and $\zeta_{\min} = 3.0 \times 10^{-3} r_p$ were chosen to reproduce experimental results by Gondret et al.[58] ($r_p = d_p/2$ is the particle radius).

To calibrate n_f and $\zeta_{\min}$ for the collocated IBM in this article, the particle is initially placed at a height 2 d_p below the top wall and its velocity in the z-direction is prescribed according to Biegert et al.[16] as

TABLE 3 Simulation parameters and setup to validate the collision models against the wall bounce experiments by Gondret et al.[58]

Stokes number *St*	**27**	**152**
Particle diameter d_p(m)	0.006	0.003
Density ratio ρ_p/ρ_f	8.083	8.342
Kinematic viscosity $\nu_f(10^{-4}\ \mathrm{m}^2/\mathrm{s})$	1.036	0.107
Restitution coefficient e_{dry}	0.97	0.97
Impact velocity u_{imp}(m/s)	0.518	0.585
Domain size (m × m × m)	0.08 × 0.08 × 0.16	0.02 × 0.02 × 0.2
Grid resolution d_p/h	19.2	19.2
Particle initial vertical position z_0 (m)	0.075	0.197
Boundary conditions	p × p × ns	p × p × ns
Outer forcing loops n_f	0,1,2,3,5,10,20	0,1,2,3,5,10
Dirac delta function	Three-point	Three-point
Pressure scheme (θ)	Projection	Projection
Time step Δt (s)	2.5×10^{-4}	8.9×10^{-5}

Note: Boundary conditions are periodic (p) or no-slip (ns).

$$u(t) = -u_{imp}\left(1 - e^{-40t}\right). \tag{52}$$

This ensures a smooth acceleration to the desired impact velocity u_{imp}. The particle is then allowed to move freely once the distance to the bottom wall satisfies $\zeta_n < d_p/2$, whereupon the particle is subject to interaction with the fluid and wall.

To understand the effect of the number of outer forcing loops n_f on the collision model, we conducted simulations with fixed $\zeta_{\min}$ and varied n_f to reproduce the particle rebound height. Figure 9 shows the effect of n_f on the rebound height with $\zeta_{\min} = 2.5 \times 10^{-3} r_p$. As shown in Figure 10, the maximum rebound height initially increases and then decreases with further increases in n_f, demonstrating the existence of a value of n_f that achieves a maximum rebound height for a given $\zeta_{\min}$.

Although there is a value of n_f that maximizes the rebound height, the maximum value may not be the value that most closely matches the experiments because the value of $\zeta_{\min}$ is also important. To assess the effects of $\zeta_{\min}$, we simulated cases with varying $\zeta_{\min}$ and fixing $n_f = 1$ to study its effects on particle-particle interactions (the parameters are the same as those for case $St = 27$ in Table 3). As shown in Figure 11, $\zeta_{\min}$ has a significant effect on the rebound height because it dictates the lubrication force exerted on the particle. The smaller $\zeta_{\min}$, the greater the lubrication force experienced by the particle, hence the lower the particle rebound height. Unlike the effect of n_f, the rebound height decreases monotonically with decreasing $\zeta_{\min}$. The best values of n_f and $\zeta_{\min}$ are obtained by conducting simulations with different $\zeta_{\min}$ for $n_f = 1, 2$ and 3 and choosing values that best match the experiments for both $St = 27$ and $St = 152$. The computed errors for all cases simulated are summarized in Table 4. For $n_f = 2$ and $n_f = 3$ as shown in Figure 12, the values of $\zeta_{\min}$ to obtain the most accurate rebound heights are $3.0 \times 10^{-3} r_p$ and $3.5 \times 10^{-3} r_p$, respectively. Since $n_f = 2$ is already satisfactory for fluid-particle interaction, $n_f = 2$ and $\zeta_{\min} = 3.0 \times 10^{-3} r_p$ are chosen for both computational cost and accuracy.

3.4 | Strong and weak computational scaling

The code is parallelized with MPI (openmpi-4.0) and we used the Hypre libraries developed by Lawrence Livermore National Laboratories[59] to invert the large linear systems associated with the fluid pressure and viscous terms. Lagrangian particle information is transferred between processors using an MPI struct. We used Stampede2 KNL (TACC at University of Texas, Austin) with Intel Xeon Phi 7250 processors (1.4 GHz) from XSEDE[60] to obtain the scaling results outlined below.

The scaling test cases were conducted with a fluidized-bed reactor that is periodic in the x- and y-directions with inflow and outflow conditions at the top and bottom boundaries (in the z-direction; see Section 3.5 for details). In all simulations, the particle diameter is d_p =2 mm, the upward flow velocity is 0.05 m s^{-1} and the domain size in the x, y, and

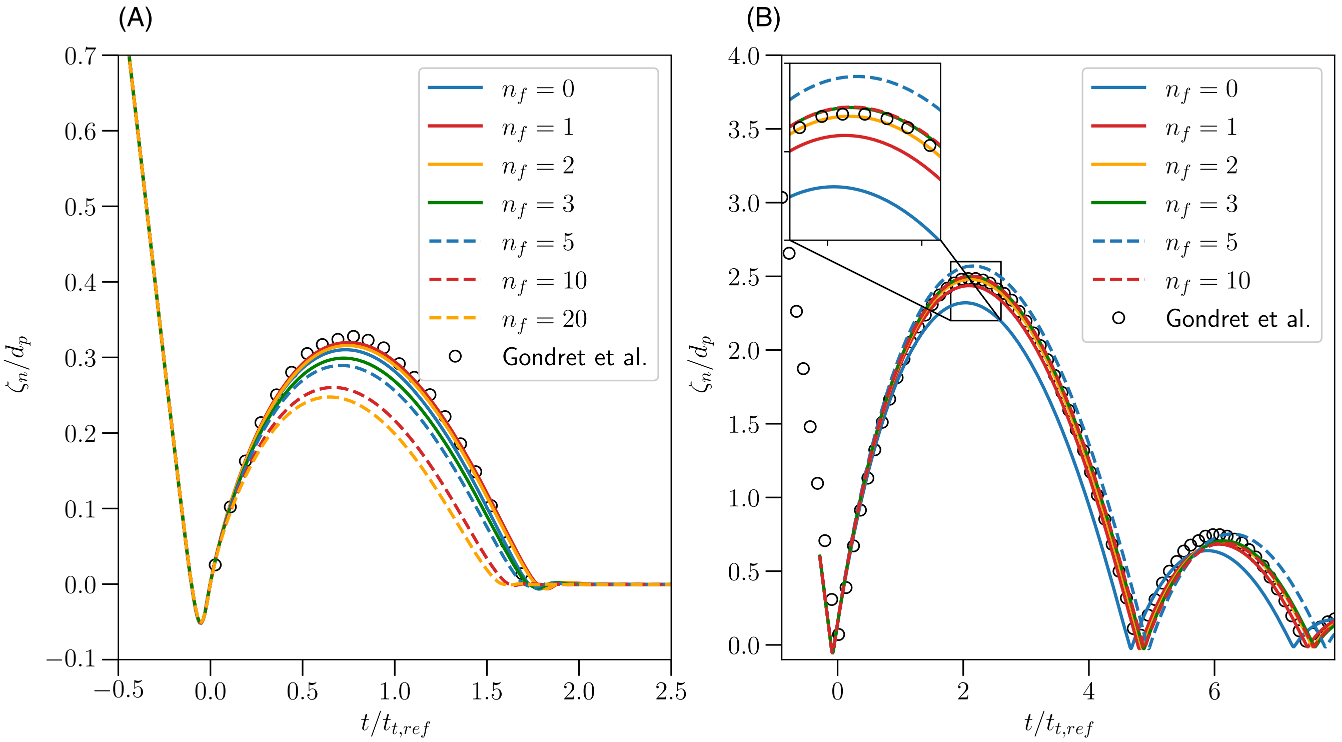

FIGURE 9 Simulated height of the particle surface (ζ_n) bouncing off of a wall with *St* = 27 (A) and 152 (B) with different number of outer forcing loops n_f compared to published values. The minimum distance from the wall is $\zeta_{\min} = 2.5 \times 10^{-3} r_p$. [Colour figure can be viewed at wileyonlinelibrary.com]

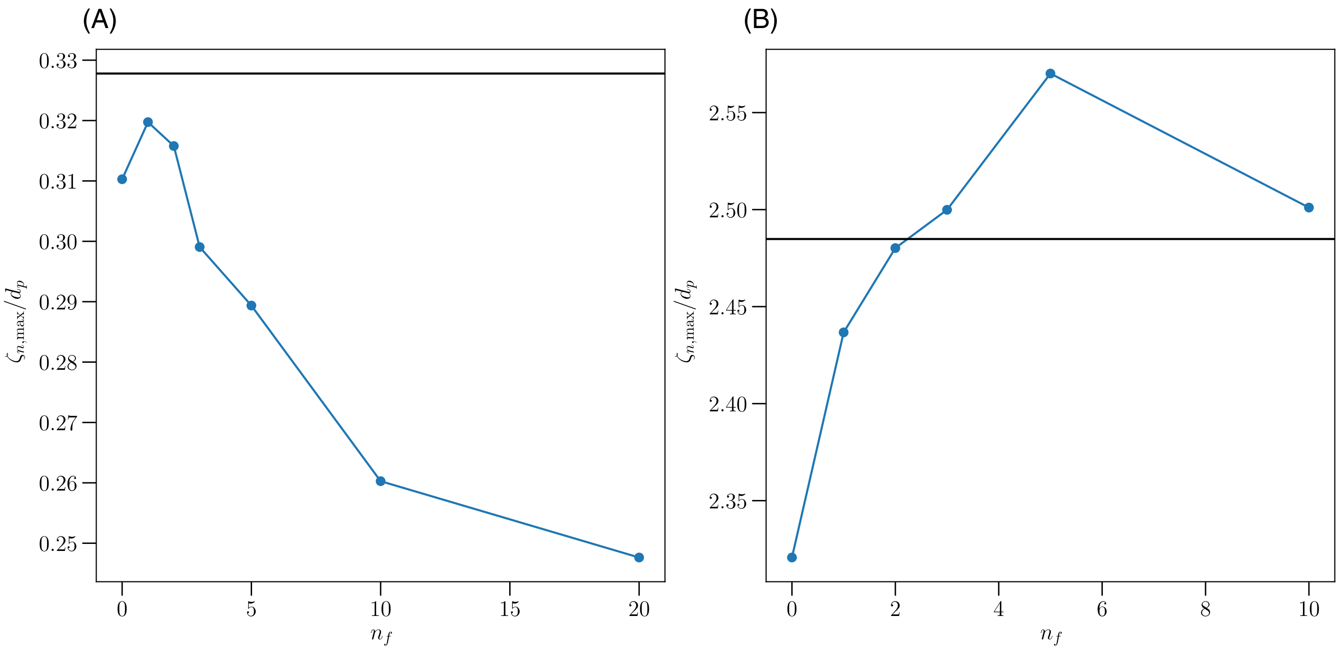

FIGURE 10 Maximum rebounding height of the center of a particle ($\zeta_{n,\max}$) with *St* = 27 (A) and 152 (B) as a function of number of outer forcing loops n_f. The black line represents the maximum rebounding height from experiments by Gondret et al.[58] [Colour figure can be viewed at wileyonlinelibrary.com]

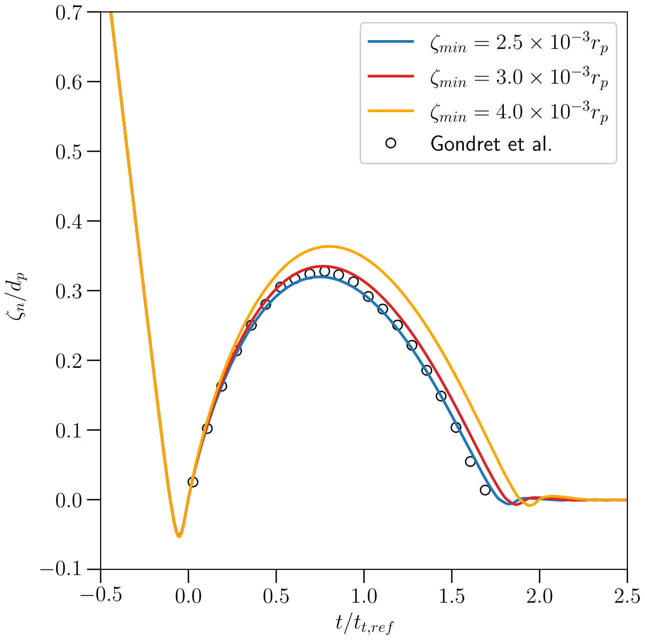

FIGURE 11 Simulated height of the particle surface (ζ_n) bouncing off of a wall with $St = 27$ with different $\zeta_{\min}$ and $n_f = 1$. [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 4 Summary of error $|\zeta_{n,\max,sim} - \zeta_{n,\max,exp}|$ between simulations and experiments by Gondret et al.[58] as a function of $\zeta_{\min}$ and n_f.

	$n_f = 1$	$n_f = 2$		$n_f = 3$	
$\zeta_{\min}$	$St = 27$	$St = 27$	$St = 152$	$St = 27$	$St = 152$
$2.5 \times 10^{-3} r_p$	0.008	0.012	0.005	0.029	0.015
$3.0 \times 10^{-3} r_p$	0.007	0.002	0.007	-	-
$3.5 \times 10^{-3} r_p$	-	0.014	0.016	0.003	0.036
$4.0 \times 10^{-3} r_p$	0.036	0.025	0.025	0.008	0.045
$5.0 \times 10^{-3} r_p$	-	-	-	0.033	0.059

z directions is $15d_p \times 15d_p \times 45d_p$. Ten time steps were computed with a time-step size of 10^{-4} s, and results are reported as the average wall-clock time per time step.

Strong scaling of the code is assessed by fixing the number of particles and number of grid points, and varying the total number of MPI tasks. To demonstrate strong scaling behavior of our code, we conducted three-dimensional simulations with 2000 spherical particles. The particles were uniformly distributed in the domain or closely packed near the bed to provide worst- and best-case scaling scenarios. The total problem size was either $256 \times 256 \times 768$ or $512 \times 512 \times 1536$ Eulerian grid points.

In order to compare the parallel performance of the components of the code related to the flow and particle solvers, we define the total wallclock time with n MPI tasks as the sum of the time needed for the flow and particle calculations as $t_n = t_{\text{flow},n} + t_{\text{particles},n}$. The speedup related to calculation of the flow+particles is then given by $S_{\text{flow+particles},n} = t_{24}/t_n$, while the speedup related to the flow solver only is $S_{\text{flow},n} = t_{\text{flow},24}/t_{\text{flow},n}$, where $n \geq 24$ is chosen due to memory limitation. As shown in Figure 13, since there are substantially fewer particles than flow grid cells, the parallel efficiency of the particle solver is smaller owing to the relatively fine-grained parallelism for the particle calculations. This effect is pronounced when the simulations are initialized with a closely packed bed, in which case load balancing is less efficient because the particle workload is disproportionately assigned to processors containing the closely packed particles.

In weak scaling, the typical approach is to keep the number of grid points on each processor constant. This is achieved by varying the number of processors and total number of grid points simultaneously. In our method, weak scaling is demonstrated by keeping both the number of particles and grid points on each subdomain constant. However, a challenge

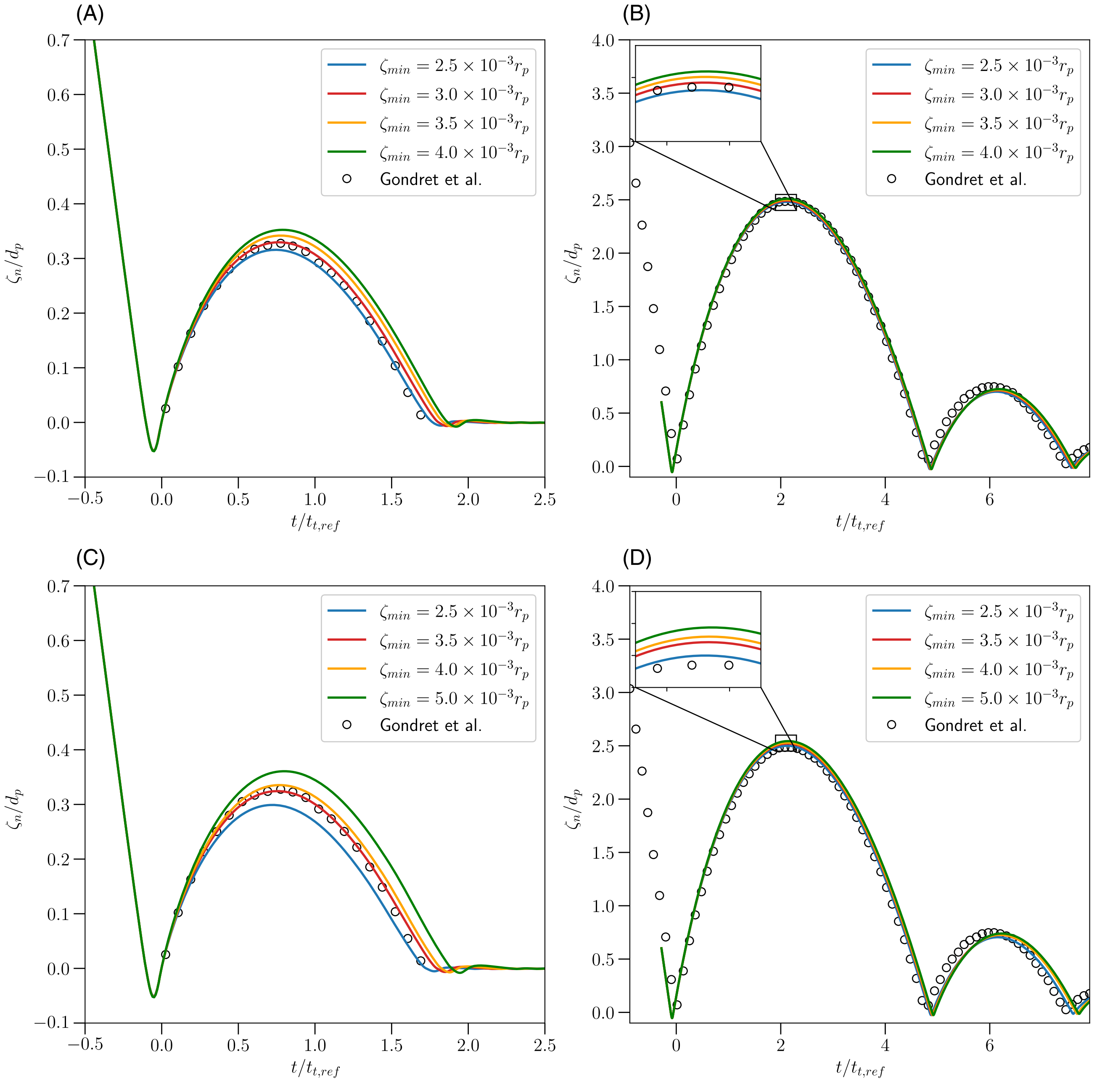

FIGURE 12 Simulated height of the particle surface (ζ_n) bouncing off of a wall with (A) $n_f = 2$ and $St = 27$, (B) $n_f = 2$ and $St = 152$, (C) $n_f = 3$ and $St = 27$ and (D) $n_f = 3$ and $St = 152$ with different $\zeta_{\min}$. [Colour figure can be viewed at wileyonlinelibrary.com]

with the weak scaling is that the computational cost of the particle solver increases with grid refinement related to the flow solver because the particle surfaces are resolved with more Lagrangian markers that must be coincident with the Eulerian grid cells. Therefore, weak scaling disproportionally adds more work to the particle solver when refining the Eulerian grid. To ensure that the workload for each processor related to the particle solver is fair, we reduce the particle diameter in proportion to the grid spacing when refining the grid. For example, refining the grid from $64 \times 64 \times 192$ to $128 \times 128 \times 384$ would reduce the particle diameter d_p from 2 to 1 mm. This ensures that the number of Lagrangian markers needed to simulate the fluid-particle interactions remains constant on each processor.

Weak scaling is demonstrated for different grid sizes on each processor and with two different initial locations of the 125 particles per processor: (a) all particles are clustered in the center of each subdomain so that no information is exchanged, and (b) most particles are distributed along the boundaries of the processor so that particle information on the boundaries must be exchanged. As shown in Figure 14, scaling is best when using 64^3 grid points per MPI task,

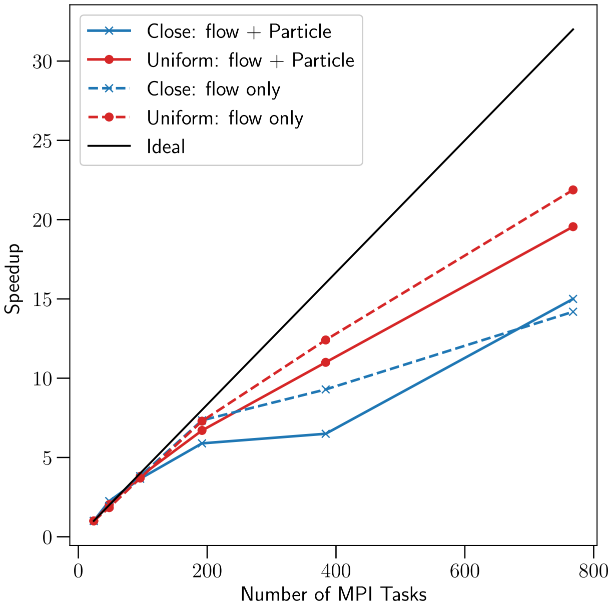

FIGURE 13 Strong scaling with 2000 particles in a doubly-periodic cubic domain, showing how the particle solver is not as efficient as the flow solver, and the efficiency is better for the uniform distribution of particles owing to improved load balancing. [Colour figure can be viewed at wileyonlinelibrary.com]

although the performance is not as good for case (b). However, typical cases are expected to have scaling behavior that is somewhere between cases (a) and (b).

Overall, the weak scaling results indicate that a simulation with 400 million grid cells is expected to take roughly 7 s per time step using 1500 MPI tasks (assuming behavior that is half-way between cases (a) and (b)). Extrapolation indicates that a simulation with 1 billion grid cells should take 9 s per time step with 3000 MPI tasks, and a simulation with 5 billion grid cells should take roughly 20 s per time step.

3.5 | Monodispersed and bidispersed particle fluidization

To demonstrate the capability of the method to simulate fluidized bed reactors, we conducted simulations showing that we can match the porosity ($1-\phi$ where ϕ is the volume fraction) predicted by experiments. In a fluidized bed, the porosity can be predicted with the power law relationship

$$u^* = \frac{u_0}{w_{t,ref}} = k(1-\phi)^n, \tag{53}$$

where u^* is the normalized velocity, u_0 is the superficial or upflow velocity, $w_{t,ref}$ is settling velocity of a single particle in the domain of interest, $k = 0.7 - 0.9$ is a constant prefactor[2,6,8,61] and n is the power law exponent. The general consensus in the literature is that Equation (53) can predict the hindered settling velocity for a particle suspension or the porosity for fluidization. Researchers have established various relationships to relate n to the terminal Reynolds number of a single particle given by Equation (51).[5,62] Richardson and Zaki[5] employ the stepwise function

$$n = \begin{cases} 4.65, & Re_{t,\infty} < 0.2, \\ 4.35 Re_{t,\infty}^{-0.03}, & 0.2 \leq Re_{t,\infty} < 1, \\ 4.45 Re_{t,\infty}^{-0.1}, & 1 \leq Re_{t,\infty} < 500, \\ 2.39, & Re_{t,\infty} \geq 500. \end{cases} \tag{54}$$

Garside and Al-Dibouni[62] improved the relationship with a continuous sigmoid function proposed to relate n and the terminal Reynolds number of a single particle in the domain of interest, $Re_{t,ref} = w_{t,ref} d_p / \nu_f$ where $w_{t,ref}$ is the terminal

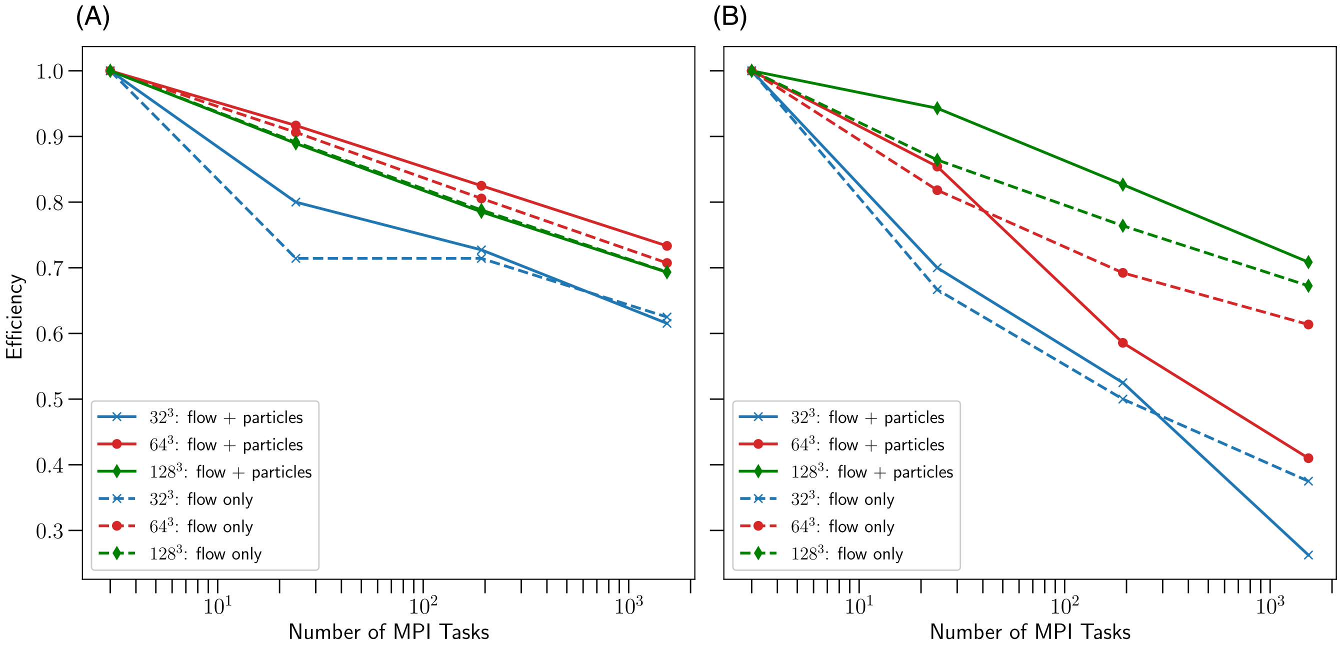

FIGURE 14 Weak scaling for the flow solver and for the flow+particle solver with different grid sizes on each processor for a doubly-periodic cubic domain. The scaling for actual simulations is expected to be somewhere between the best-case (A) and worst-case (B) particle scenarios. [Colour figure can be viewed at wileyonlinelibrary.com]

velocity of a single particle in the domain of interest, as

$$\frac{5.1-n}{n-2.7}=0.1Re_{t,ref}^{0.9}, \tag{55}$$

which is 20% more accurate than Equation (54).[6]

To verify that our method can reproduce the power law 53, three-dimensional simulations are conducted with $N_p=2000$ particles in the reactor channel shown in Figure 15. The particles have a uniform diameter $d_p=0.002$ m and density $\rho_p=1300\,\mathrm{kg\,m^{-3}}$, and the fluid has a kinematic viscosity $\nu_f=1.0037\times10^{-6}\,\mathrm{m^2\,s^{-1}}$ and density $\rho_f=998.21\,\mathrm{kg\,m^{-3}}$ ($\rho_p/\rho_f=1.3$). The grid spacing is uniform in the x, y, and z directions and given by $\Delta x=\Delta y=\Delta z=h=d_p/25.6$, which is sufficient to resolve the flow-particle interactions as demonstrated in Section 3.1. The square channel dimensions are given by $L_x=L_y=10d_p$ and its length is $L_z=60d_p$, giving a three-dimensional grid with 256×256×1536 grid points. The time-step size is $\Delta t=1.5\times10^{-4}$ s, resulting in a maximum Courant number of 0.4 for the cases with the highest upflow velocities. In all simulations, 384 processors were used ensuring a 64^3 computational domain for each processor. Each time step requires 10 s wall clock which is consistent with the scaling results presented in Section 3.4.

Simulations are initialized with a uniform distribution of close-packed particles with a spacing of $1d_p$ at the bottom of the domain and the flow is impulsively started from rest. The upflow velocity leads to expansion of the bed and random motion of the particles until statistical equilibrium is reached, at which time the dynamics are independent of the initial particle distribution. Simulations are run for a total of 100 τ_T, where $\tau_T=d_p/u_0$ is the particle turnover time. As it takes roughly 30 τ_T to reach statistical equilibrium, results are time averaged over the last 70 τ_T.

The average upflow velocity at the inlet, u_0, is varied to investigate Reynolds number effects. A total of six simulations were conducted with $0.01\le u_0\le0.35\,\mathrm{m\,s^{-1}}$, giving $20\le Re_p\le70$ where $Re_p=u_0d_p/\nu_f$. For all cases, the pressure is specified at the top boundary as $p=0$, while at the bottom boundary the inflow velocity is specified and all side wall boundary conditions are periodic. Due to the absence of walls which would produce a Poiseuille velocity profile at the inlet, the inflow velocity is uniform and given by u_0. For collision models, the minimum separation distance in the lubrication model is set to $\zeta_{\min}=3.0\times10^{-3}r_p$ and the dry restitution coefficient $e_{dry}=0.97$. The parameters dictating rolling/sticking $\mu_s=0.11$ and sliding $\mu_k=0.8$ are used.

Figure 16 demonstrates the relationship between the normalized velocity u^* and the time-averaged porosity $1-\overline{\phi}$. A straight line on a log-log scale indicates the results follow a power law relationship. To assess the accuracy of the

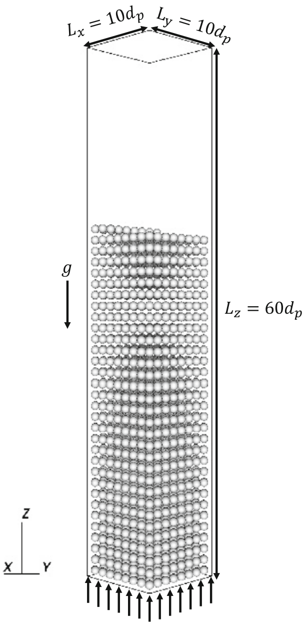

FIGURE 15 Schematic of the three-dimensional computational domain, showing the initial particle locations making up the fluidized bed.

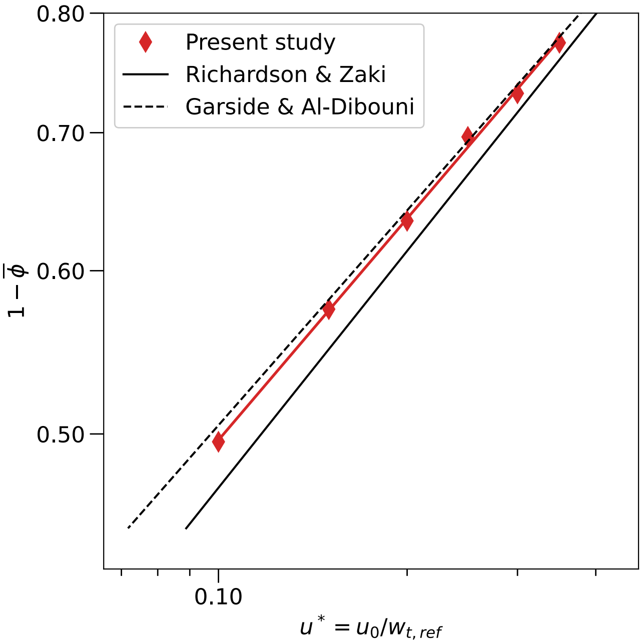

FIGURE 16 Time-averaged porosity $1-\overline{\phi}$ as a function of superficial velocity u_0 normalized by the settling velocity of a single particle in the domain of interest $w_{t,ref}$ for the simulated cases. The lines were constructed based on fitting to the power law Equation (53). [Colour figure can be viewed at wileyonlinelibrary.com]

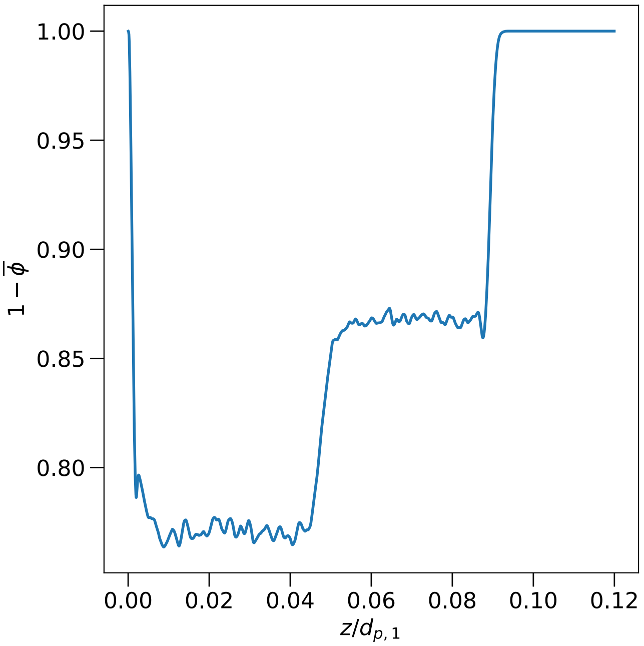

FIGURE 17 Time-averaged porosity $1-\overline{\phi}$ as a function of vertical position z for an upflow velocity of 0.35 m s^{-1} in a bidispersed fluidized bed. [Colour figure can be viewed at wileyonlinelibrary.com]

simulations, we regressed the power law (Equation 53) which gives $k = 0.72$ and $n = 2.82$. This is in close agreement with the values of $n = 2.62$ (Equation 54) and $n = 2.88$ (Equation 55) and trends reported by other low-porosity PRS.[6,8]

To demonstrate the capability to simulate polydispersed particles, we increased the simulation complexity from monodispersed to bidispersed fluidization. Here, we performed a three-dimensional simulation with $N_{p,\mathrm{total}} = 3376$ particles in the reactor. The particles have two uniform diameters $d_{p,1} = 0.002$ m and $d_{p,2} = d_{p,1}/1.4$ and density $\rho_p = 1300$ kg m^{-3}. The number of particles for $d_{p,1}$ and $d_{p,2}$ are 1000 and 2376, respectively. The fluid properties are identical to the monodispersed fluidization simulations. A uniform grid spacing with $h/d_{p,1} = 25.6$ was used to ensure accuracy. The square channel dimensions were identical, giving a grid with 256×256×1536 points. The inflow velocity was 0.35 m s^{-1}, giving Re_p calculated based on $d_{p,1}$ as 70. The time-step size Δt is calculated with both advection and diffusion Courant numbers that are defined as $C_{\mathrm{adv}} = u_0\Delta t/h$ and $C_{\mathrm{diff}} = \nu_f \Delta t/h^2$, respectively. We ensure that the maximum Courant number $C_{\max} = \max(C_{\mathrm{adv}}, C_{\mathrm{diff}}) = 0.25$. Figure 17 shows the time-averaged porosity $1-\overline{\phi}$ as a function of z. The segregated bidispersed fluidized bed consists of three regions with the following properties, in order from the bottom to the top of the reactor: (1) monodispersed fluidization with particles of size $d_{p,1}$, (2) bidispersed fluidization consisting of both particle sizes, and (3) monodispersed fluidization with particles of size $d_{p,2}$. We computed the spatially-averaged porosity for the lower and upper regions as $1-\overline{\phi} = 0.771$ and 0.868, respectively which are approximately equivalent to the time-averaged porosity obtained for the equivalent monodispersed fluidization simulations ($1-\overline{\phi} = 0.773$ and 0.869). These results indicate that bidispersed fluidized beds essentially segregate into layers that behave like their monodispersed equivalents, as discussed extensively in a article using the present method by Yao et al.[3]

4 | CONCLUSION

We develop a modified particle-resolved simulation framework that combines the collocated-grid IBM and collision models for polydisperse particles. An improved IBM[24] is coupled with the fractional step method[34] to solve the Navier Stokes equations. The method is shown to be slightly higher than first-order accurate in space and scales well on hundreds of processors on high-performance parallel computing platforms. To enforce no-slip boundary conditions on particle surfaces, a tuning parameter, the number of outer forcing loops n_f, was introduced and calibrated based on experiments. The accuracy of fluid-particle interactions was validated against various test cases corresponding to a single particle settling in a periodic domain with different terminal settling velocities. These cases showed that the tuning parameter $n_f \geq 1$. Since the IBM does not resolve the fluid-particle interactions when particles are very close to or in contact with one another, collision models are implemented to simulate the interactions between particles and flow. The normal contact model we implement is based on the adaptive collision time step model proposed by Kempe and Fröhlich[27] to remove the constraint resulting from the collision time-step size which is typically much smaller than the fluid time-step size. The tangential contact model utilizes a spring-dash-pot model[16] while the lubrication model is modified based on an algebraic relationship. To ensure accurate collisions between particles, specification of the minimum separation distance between particles, $\zeta_{\min}$, is needed. Since n_f affects both fluid-particle and particle-particle interactions, n_f and $\zeta_{\min}$ must be calibrated to reproduce experimental results of particle-wall interactions. The collision models are calibrated with experimental results of a single particle colliding with a wall, in which the rebound height of the particle matches the experimental rebound trajectories with tuning parameters $n_f = 2$ and $\zeta_{\min} = 3.0 \times 10^{-3} r_p$. Our results also show that a grid resolution of $d_p/h = 20$ can produce results that closely match the experiments. These parameters represent a good balance between computational cost and accuracy. Finally, our collocated-grid simulation framework for polydispersed particles was used to simulate both monodispersed and bidispersed fluidized-bed expansion, demonstrating its capability in reproducing the power law relationship between superficial velocity and porosity.

ACKNOWLEDGMENTS

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation Grant number ACI-1548562. Simulations were conducted with supercomputer resources under XSEDE Project CTS190063. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this article. This work was funded by the California Energy Commission (CEC) under CEC project number EPC-16-017, the U.S. NSF Engineering Center for Reinventing of the Nation's Urban Water Infrastructure (ReNUWIt) under Award No. 1028968, and Office of Naval Research Grant N00014-16-1-2256. Bernhard Vowinckel gratefully acknowledges the support through the German Research Foundation, (DFG) Grant VO2413/2-1. This article was prepared as a result of work sponsored in part by the California Energy Commission. It does not necessarily represent the views of the Energy Commission, its employees, or the State of California. Neither the Commission, the State of California, nor the Commission's employees, contractors, or subcontractors makes any warranty, express or implied, or assumes any legal liability for the information in this article; nor does any party represent that the use of this information will not infringe upon privately owned rights. This article has not been approved or disapproved by the Commission, nor has the Commission passed upon the accuracy of the information in this article.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

ORCID

Yinuo Yao https://orcid.org/0000-0001-8328-6072
Bernhard Vowinckel https://orcid.org/0000-0001-6853-7750
Craig S. Criddle https://orcid.org/0000-0002-2750-8547
Oliver B. Fringer https://orcid.org/0000-0003-3176-6925

REFERENCES

1. Shin C, McCarty PL, Kim J, Bae J. Pilot-scale temperate-climate treatment of domestic wastewater with a staged anaerobic fluidized membrane bioreactor (SAF-MBR). *Bioresour Technol*. 2014;159:95-103. doi:10.1016/j.biortech.2014.02.060
2. Yao Y, Criddle CS, Fringer OB. The effects of particle clustering on hindered settling in high-concentration particle suspensions. *J Fluid Mech*. 2021;920:A40. doi:10.1017/jfm.2021.470
3. Yao Y, Criddle CS, Fringer OB. Comparison of the properties of segregated layers in a bidispersed fluidized bed to those of a monodispersed fluidized bed. *Phys Rev Fluids*. 2021;6(8):084306. doi:10.1103/PhysRevFluids.6.084306
4. Fringer OB, Dawson CN, He R, Ralston DK, Zhang YJ. The future of coastal and estuarine modeling: findings from a workshop. *Ocean Model*. 2019;143:101458. doi:10.1016/j.ocemod.2019.101458
5. Richardson JF, Zaki WN. Sedimentation and fluidisation: part I. *Trans Inst Chem Eng*. 1954;32:35-53.
6. Yin X, Koch DL. Hindered settling velocity and microstructure in suspensions of solid spheres with moderate Reynolds numbers. *Phys Fluids*. 2007;19(9):093302. doi:10.1063/1.2764109
7. Zhang Z, Prosperetti A. A second-order method for three-dimensional particle simulation. *J Comput Phys*. 2005;210(1):292-324. doi:10.1016/j.jcp.2005.04.009

8. Willen DP, Prosperetti A. Resolved simulations of sedimenting suspensions of spheres. *Phys Rev Fluids*. 2019;4(1):014304. doi:10.1103/PhysRevFluids.4.014304
9. Glowinski R, Pan TW, Hesla TI, Joseph DD. A distributed Lagrange multiplier/fictitious domain method for particulate flows. *Int J Multiphase Flow*. 1999;25(5):755-794. doi:10.1016/S0301-9322(98)00048-2
10. Glowinski R, Pan TW, Hesla TI, Joseph DD, Périaux J. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow. *J Comput Phys*. 2001;169(2):363-426. doi:10.1006/jcph.2000.6542
11. Ladd AJC. Hydrodynamic interactions and the viscosity of suspensions of freely moving spheres. *J Chem Phys*. 1989;90(2):1149-1157. doi:10.1063/1.456170
12. Peskin CS. Numerical analysis of blood flow in the heart. *J Comput Phys*. 1977;25(3):220-252. doi:10.1016/0021-9991(77)90100-0
13. Peskin CS. The immersed boundary method. *Acta Numer*. 2002;11:479-517. doi:10.1017/S0962492902000077
14. Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. *J Comput Phys*. 2005;209(2):448-476. doi:10.1016/j.jcp.2005.03.017
15. Akiki G, Balachandar S. Immersed boundary method with non-uniform distribution of Lagrangian markers for a non-uniform Eulerian mesh. *J Comput Phys*. 2016;307:34-59. doi:10.1016/j.jcp.2015.11.019
16. Biegert E, Vowinckel B, Meiburg E. A collision model for grain-resolving simulations of flows over dense, mobile, polydisperse granular sediment beds. *J Comput Phys*. 2017;340:105-127. doi:10.1016/j.jcp.2017.03.035
17. Kim J, Kim D, Choi H. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *J Comput Phys*. 2001;171(1):132-150. doi:10.1006/jcph.2001.6778
18. Wang Z, Fan J, Luo K. Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles. *Int J Multiphase Flow*. 2008;34(3):283-302. doi:10.1016/j.ijmultiphaseflow.2007.10.004
19. Breugem WP. A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows. *J Comput Phys*. 2012;231(13):4469-4498. doi:10.1016/j.jcp.2012.02.026
20. Apte SV, Martin M, Patankar NA. A numerical method for fully resolved simulation (FRS) of rigid particle–flow interactions in complex flows. *J Comput Phys*. 2009;228(8):2712-2738. doi:10.1016/j.jcp.2008.11.034
21. Apte SV, Finn JR. A variable-density fictitious domain method for particulate flows with broad range of particle–fluid density ratios. *J Comput Phys*. 2013;243:109-129. doi:10.1016/j.jcp.2012.12.021
22. Yu Z, Shao X. A direct-forcing fictitious domain method for particulate flows. *J Comput Phys*. 2007;227(1):292-314. doi:10.1016/j.jcp.2007.07.027
23. Huang WX, Tian FB. Recent trends and progress in the immersed boundary method. *Proc Inst Mech Eng Part C*. 2019;233(23-24):7617-7636. doi:10.1177/0954406219842606
24. 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):3663-3684. doi:10.1016/j.jcp.2012.01.021
25. 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. doi:10.1016/j.jcp.2020.109862
26. Yang Y, Balachandar S. A scalable parallel algorithm for direct-forcing immersed boundary method for multiphase flow simulation on spectral elements. *J Supercomput*. 2021;77(3):2897-2927. doi:10.1007/s11227-020-03371-2
27. Kempe T, Fröhlich J. Collision modelling for the interface-resolved simulation of spherical particles in viscous fluids. *J Fluid Mech*. 2012;709:445-489. doi:10.1017/jfm.2012.343
28. Yu Z, Lin Z, Shao X, Wang LP. A parallel fictitious domain method for the interface-resolved simulation of particle-laden flows and its application to the turbulent channel flow. *Eng Appl Comput Fluid Mech*. 2016;10(1):160-170. doi:10.1080/19942060.2015.1092268
29. Xia Y, Xiong H, Yu Z, Zhu C. Effects of the collision model in interface-resolved simulations of particle-laden turbulent channel flows. *Phys Fluids*. 2020;32(10):103303. doi:10.1063/5.0020995
30. Kidanemariam AG, Uhlmann M. Interface-resolved direct numerical simulation of the erosion of a sediment bed sheared by laminar channel flow. *Int J Multiphase Flow*. 2014;67:174-188. doi:10.1016/j.ijmultiphaseflow.2014.08.008
31. Zaidi AA, Tsuji T, Tanaka T. Hindered settling velocity & structure formation during particle settling by direct numerical simulation. *Proc Eng*. 2015;102:1656-1666. doi:10.1016/j.proeng.2015.01.302
32. Luding S. Cohesive, frictional powders: contact models for tension. *Granul Matter*. 2008;10(4):235. doi:10.1007/s10035-008-0099-x
33. Thornton C, Cummins SJ, Cleary PW. An investigation of the comparative behaviour of alternative contact force models during inelastic collisions. *Powder Technol*. 2013;233:30-46. doi:10.1016/j.powtec.2012.08.012
34. Zang Y, Street RL, Koseff JR. A non-staggered grid, fractional step method for time-dependent incompressible navier-stokes equations in curvilinear coordinates. *J Comput Phys*. 1994;114(1):18-33. doi:10.1006/jcph.1994.1146
35. Lee H, Balachandar S. Drag and lift forces on a spherical particle moving on a wall in a shear flow at finite Re. *J Fluid Mech*. 2010;657:89-125. doi:10.1017/S0022112010001382
36. Lee H, Ha MY, Balachandar S. Rolling/sliding of a particle on a flat wall in a linear shear flow at finite Re. *Int J Multiphase Flow*. 2011;37(2):108-124. doi:10.1016/j.ijmultiphaseflow.2010.10.005
37. Zhang X, Gu X, Ma N. A ghost-cell immersed boundary method on preventing spurious oscillations for incompressible flows with a momentum interpolation method. *Comput Fluids*. 2021;220:104871. doi:10.1016/j.compfluid.2021.104871
38. Mittal R, Dong H, Bozkurttas M, Najjar FM, Vargas A, van Loebbecke A. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *J Comput Phys*. 2008;227(10):4825-4852. doi:10.1016/j.jcp.2008.01.028

39. Tseng YH, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. *J Comput Phys*. 2003;192(2):593-623. doi:10.1016/j.jcp.2003.07.024
40. Gilmanov A, Sotiropoulos F. A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *J Comput Phys*. 2005;207(2):457-492. doi:10.1016/j.jcp.2005.01.020
41. Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *J Comput Phys*. 2000;161(1):35-60. doi:10.1006/jcph.2000.6484
42. Roma AM, Peskin CS, Berger MJ. An adaptive version of the immersed boundary method. *J Comput Phys*. 1999;153(2):509-534. doi:10.1006/jcph.1999.6293
43. Rai M, Moin P. Direct simulations of turbulent flow using finite-difference schemes. *J Comput Phys*. 1991;96(1):15-53. doi:10.1016/0021-9991(91)90264-l
44. Uhlmann M. First experiments with the simulation of particulate flows. Centro de Investigaciones Energeticas Medioambientales y Tecnologicas (CIEMAT); 2003.
45. Uhlmann M. New results on the simulation of particulate flows. Centro de Investigaciones Energeticas Medioambientales y Tecnologicas (CIEMAT) 2004.
46. Rhie CM, Chow WL. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA J*. 1983;21(11):1525-1532. doi:10.2514/3.8284
47. Ferziger JH, Perić M, Street RL. *Computational Methods for Fluid Dynamics*. Springer; 2019.
48. Luo K, Wang Z, Fan J, Cen K. Full-scale solutions to particle-laden flows: multidirect forcing and immersed boundary method. *Phys Rev E Stat Nonlinear Soft Matter Phys*. 2007;76(6 Pt 2):066709. doi:10.1103/PhysRevE.76.066709
49. Biegert EK. *Eroding Uncertainty: Towards Understanding Flows Interacting with Mobile Sediment Beds Using Grain-Resolving Simulations*. PhD thesis. UC Santa Barbara, 2018.
50. Höfler K, Schwarzer S. Navier-Stokes simulation with constraint forces: finite-difference method for particle-laden flows and complex geometries. *Phys Rev E*. 2000;61(6 Pt B):7146-7160. doi:10.1103/physreve.61.7146
51. Cox RG, Brenner H. The slow motion of a sphere through a viscous fluid towards a plane surface—II small gap widths, including inertial effects. *Chem Eng Sci*. 1967;22(12):1753-1777. doi:10.1016/0009-2509(67)80208-2
52. Ray S, Kempe T, Fröhlich J. Efficient modelling of particle collisions using a non-linear viscoelastic contact force. *Int J Multiphase Flow*. 2015;76:101-110. doi:10.1016/j.ijmultiphaseflow.2015.06.006
53. Thornton C, Cummins SJ, Cleary PW. An investigation of the comparative behaviour of alternative contact force models during elastic collisions. *Powder Technol*. 2011;210(3):189-197. doi:10.1016/j.powtec.2011.01.013
54. Costa P, Boersma BJ, Westerweel J, Breugem WP. Collision model for fully resolved simulations of flows laden with finite-size particles. *Phys Rev E Stat Nonlinear Soft Matter Phys*. 2015;92(5):053012. doi:10.1103/PhysRevE.92.053012
55. Ten Cate A, Nieuwstad CH, Derksen JJ, Van den Akker HE. Particle imaging velocimetry experiments and lattice-Boltzmann simulations on a single sphere settling under gravity. *Phys Fluids*. 2002;14(11):4012-4025. doi:10.1063/1.1512918
56. Mordant N, Pinton JF. Velocity measurement of a settling sphere. *Eur Phys J B*. 2000;18(2):343-352. doi:10.1007/PL00011074
57. Armfield S, Street R. An analysis and comparison of the time accuracy of fractional-step methods for the Navier-stokes equations on staggered grids. *Int J Numer Methods Fluids*. 2002;38(3):255-282. doi:10.1002/fld.217
58. Gondret P, Lance M, Petit L. Bouncing motion of spherical particles in fluids. *Phys Fluids*. 2002;14(2):643-652. doi:10.1063/1.1427920
59. Falgout RD. An introduction to algebraic multigrid. *Comput Sci Eng*. 2006;8(6):24-33. doi:10.1109/MCSE.2006.105
60. Towns J, Cockerill T, Dahan M, et al. XSEDE: accelerating scientific discovery. *Comput Sci Eng*. 2014;16(5):62-74. doi:10.1109/MCSE.2014.80
61. Yao Y, Criddle CS, Fringer OB. Competing flow and collision effects in a monodispersed liquid–solid fluidized bed at a moderate Archimedes number. *J Fluid Mech*. 2021;927:A28. doi:10.1017/jfm.2021.780
62. Garside J, Al-Dibouni MR. Velocity-voidage relationships for fluidization and sedimentation in solid-liquid systems. *Ind Eng Chem Proc Des Dev*. 1977;16(2):206-214. doi:10.1021/i260062a008

How to cite this article: Yao Y, Biegert E, Vowinckel B, et al. Particle-resolved simulations of four-way coupled, polydispersed, particle-laden flows. *Int J Numer Meth Fluids*. 2022;1-31. doi: 10.1002/fld.5128