Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2021-0166

Vol. **31**, No. 4, pp. 1242-1271
April 2022

Hybrid Diffuse and Sharp Interface Immersed Boundary Methods for Particulate Flows in the Presence of Complex Boundaries

Jianhua Qin[1,2], Xiaolei Yang[1,2,*] and Zhaobin Li[1,2]

[1] *The State Key Laboratory of Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China.*
[2] *School of Engineering Sciences, University of Chinese Academy of Sciences, Beijing 100049, China.*

Communicated by Kun Xu

Received 9 August 2021; Accepted (in revised version) 16 February 2022

Abstract. A coupling framework that leverages the advantages of the diffuse and sharp interface immersed boundary (IB) methods is presented for handling the interaction among particles and particles with the static complex geometries of the environment. In the proposed coupling approach, the curvilinear IB method is employed to represent the static complex geometries, a variant of the direct forcing IB method is proposed for simulating particles, and the discrete element method is employed for particle-particle and particle-wall collisions. The proposed approach is validated using several classical benchmark problems, which include flow around a sphere, sedimentation of a sphere, collision of two sedimenting spheres, and collision between a particle and a flat wall, with the present predictions showing an overall good agreement with the results reported in the literature. The capability of the proposed framework is further demonstrated by simulating the interaction between multiple particles and a wall-mounted cylinder, and the particle-laden turbulent flow over periodic hills. The proposed method provides an efficient way to simulate particle-laden turbulent flows in environments with complex boundaries.

AMS subject classifications: 76T99, 76F65, 76M20
Key words: Immersed boundary method, particle-laden flow, complex geometry.

1 Introduction

Particle transport in a carrier fluid is often encountered in engineering and environmental applications. Examples of such flows include transmission of droplets during a cough [1,

*Corresponding author. *Email addresses:* xyang@imech.ac.cn (X. Yang), qinjianhua@imech.ac.cn (J. Qin), zhaobin.li@imech.ac.cn (Z. Li)

2], suspension and sedimentation of aspherical particles [3, 4], sediment transport [5] and dispersion of contaminants [6]. Approaches for simulating particle-fluid interaction can be classified into two categories according to whether the particles are modeled or resolved. The particle-modeled methods include the fully Eulerian method in which only one-way interaction is considered [4, 5], the Eulerian-Eulerian method [7, 8] in which the governing equations of the fluid phase and the particle phase are both solved by using the Eulerian approach, the point-particle Eulerian-Lagrangian method [9] in which the particles are treated as point sources to the fluid momentum and are evolved via a given drag law. The immersed boundary (IB) method [10], which can resolve the geometry of particles, belongs to the particle-resolved category.

The classic IB method was proposed by Peskin to simulate blood flows [10]. Up to date, various IB methods have been proposed [11–13], which can be classified into the diffuse interface and the sharp interface IB methods. The diffuse interface IB methods regularize the discontinuous across the fluid-structure interface by using kernel functions, which include the regularized Dirac delta functions [10, 14–18], the reproducing kernel particle method delta function [19] and the moving-least-squares reconstructions [20, 21]. Different variants of the diffuse interface IB method also differ in how the forces on the IB are computed. In the classical IB method [10], the IB forces on the flexible boundary are computed via the constitutive law, e.g., Hook's law. When applying such methods to rigid boundaries, spring constants of very high magnitudes have to be employed, causing issues of numerical instability. To solve this problem, the diffuse interface direct forcing IB method [14], which calculates the forces on the boundary by satisfying the no-slip boundary conditions, was then proposed and developed [17]. As an alternative to the diffuse interface IB methods, the sharp interface IB methods include the curvilinear immersed boundary (CURVIB) method [22], the immersed interface method [23–26], the Lagrange-multiplier based fictitious domain methods [27], the cut cell methods [28, 29], and the embedded boundary methods [30, 31], which differ in the ways of applying the boundary conditions. In the cut cell method, the cells cut by the boundary are reshaped according the actual geometry, where the boundary conditions are directly imposed. In the immersed interface method, the boundary conditions are applied based on stress discontinuities via jump conditions for the velocity, the pressure and their derivatives, which enables simulating also interface problems not restricted to fluid-solid boundaries. In the CURVIB method and others, the boundary conditions are imposed by reconstructing the velocity (and pressure) field near the boundary.

Different approaches have been developed to satisfy the no-slip boundary condition on the IB more accurately for the diffuse interface IB method. Su et al. [32] proposed to take into account the effect of force distribution process in the determination of forces on the IB by solving a linear equation with coefficients forming a banded matrix so that the no-slip boundary condition can be completely satisfied. Wang and Zhang [33] extended the matrix inversion idea to the discrete stream function flow solver. Wang et al. [34] introduced an iterative direct forcing IB method to avoid the penetration of streamlines across the boundary. To minimize the effect of the Dirac delta function on the simulated

results, Breugem [35] used an inward retraction technique to improve the accuracy near the IB. In the present work, a variant of the direct forcing IB method with subiterations is proposed to satisfy the no-slip boundary conditions more accurately. The proposed approach can be employed for discrete delta functions of any widths, without the need to construct the banded matrix as in Su et al. [32], which can be cumbersome for discrete delta functions other than the two-point hat function.

There are two advantages of the diffuse interface IB method: 1) a smooth representation of the boundary relieves the requirement on background grid resolution in order to model the boundary smoothly; (2) no pre-process is required for classifying the background grid nodes, making it naturally suitable for moving boundary problems. However, one drawback of the diffuse interface IB method is the relatively lower accuracy near the IB boundary caused by the smear effect of the distributed force. In cases where the flow near the boundary matters, the diffuse interface IB method might not be the ideal choice. The key advantage of the sharp interface IB method is the sharp representation of the IB, which enables approximately the same accuracy of the predicted flow near the IB when compared with that away from it. In the sharp interface IB method, the background grid nodes need to be classified as fluid nodes, solid nodes and those nodes near the interface. For the IB method based on the velocity reconstruction, the fluid nodes and nodes on the boundary, which are employed for the velocity interpolation, need to be identified. This process can be time-consuming for large-scale simulations with immersed boundaries with a large number of discrete points. Such pre-process needs to be performed for every time steps, making it undesirable for moving boundary problems with many bodies, e.g., particulate flows. Furthermore, for moving boundary problems, some types of sharp interface IB methods often show nonphysical force oscillations caused by the change of types of the background grid nodes. Methods for suppressing these oscillations can be found in these references [24,26,36,37].

For moving boundaries problems where contact between boundaries exists, body-fitted methods like the arbitrary-Lagrangian-Eulerian method is cumbersome. For the IB methods, three workarounds have been proven to be effective. One simple solution is to resolve the fluids near the contact region with sufficient high resolution using the adaptive mesh refinement (AMR) technique [25]. The so called lubricated IB method, on the other hand, employs the jump conditions based on the lubrication theory to simulate the contact between a vesicle and a narrow channel [38]. The above two approaches, which is either relied on complicated programming for the AMR technique or complicated formulations for the IB method, cannot be easily applied to particle-laden flows with arbitrarily complex boundaries. The third treatment of the contact, which was proposed by Biegert et al. [39] and dubbed as the 'dry' contact approach, excludes the near wall Lagrangian points to prevent the non-physical adhesion of the solid body on the wall. The third approach is easier to implement when compared with the other two. However, it is yet to be extended to particle-laden flows when the particles collide with the complex boundaries of the environment.

The objective of this work is to develop a framework of hybrid diffuse interface and

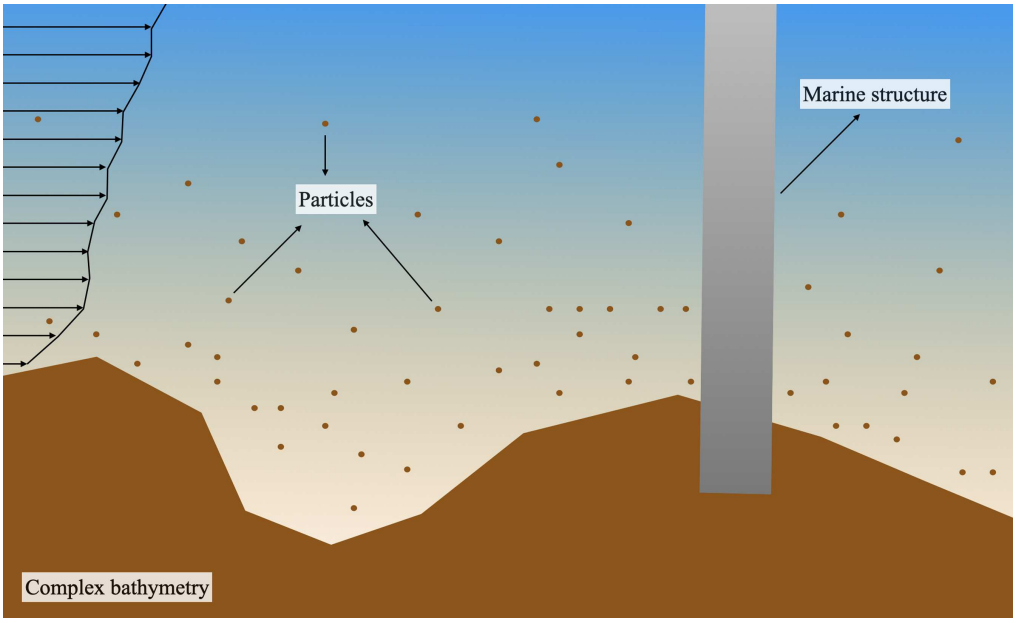


Figure 1: Schematic for particulate flows in environments with complex boundaries. In the proposed method, the complex bathymetry and marine structures are solved using the sharp interface immersed boundary method, while the particles are simulated using the diffuse interface immersed boundary method.

sharp interface IB methods, which takes advantages of both methods, for simulating fluid transportation of rigid particles in environments with complex boundaries. A typical scenario is demonstrated in Fig. 1, where sand particles collide on a cylinder mounted on a complex bathymetry. Other scenarios include particle-wall interactions with complex geometries existing in the transport gasifier, the fluidized bed, and the process of inhaled drug delivery. In this paper, we propose to simulate the stationary complex boundaries (e.g., the bathymetry and the wall-mounted cylinder) and the particles using the sharp interface CURVIB method [22] and the diffuse interface direct forcing IB method, respectively. When combining the diffuse interface direct forcing IB method and the CURVIB method, the 'dry' contact approach of Biegert et al. [39] is employed to prevent the non-physical adhesion of particles on the complex geometry. Because of the large number of particles in some applications with particulate flows and the low particle Reynolds number, the employed background grid might be relatively coarse in terms of resolving the particle. Because of this low resolution, the no-slip boundary conditions cannot be satisfied as desired. To enforce the no-slip boundary conditions on the IBs more accurately, we use the above mentioned variant of the direct forcing IB method with subiterations that is developed in this work. In comparison with previous approaches, the proposed framework is advantageous in the following two aspects: 1) arbitrarily complex boundaries of the environment can be handled easily with the CURVIB method [40], which is cumbersome or even impossible for the methods based on body-fitted grids, and is smeared for the diffuse interface IB methods, respectively; 2) particle transport can be simulated

accurately and efficiently with the diffuse interface direct forcing IB method with iterations, which requires complicated mesh techniques for the methods based on body-fitted grids, and may suffer from non-physical force oscillations and requires computationally expensive preprocessing (e.g., for the classification of the background grid nodes) for the sharp interface IB methods, respectively. With the above two advantages, particle-resolved simulations of particulate flows in the presence of complex boundaries, which are rarely reported in the literature, can be easily handled with the proposed method (which is arduous for the conventional approaches).

The rest of the paper is structured as follows. In Section 2 we introduce the hybrid diffuse interface and sharp interface IB method followed by the validation using classic benchmark problems in Section 3. Then the proposed approach is applied to simulate flow past a wall-mounted circular cylinder with multiple particles. In Section 4 we simulate and analyze particle-laden turbulent flows around periodic hills. At last, conclusions are drawn in Section 5.

2 Hybrid diffuse and sharp interface IB methods

In this section, we present the framework of hybrid diffuse and sharp interface immersed boundary methods. A schematic for this hybrid framework is shown in Fig. 2, where a particle collides with a structure. The structure is represented using the CURVIB method, which is proposed by Ge and Sotiropoulos [22] and will be briefly described in Subsection 2.1 together with the flow solver. The particle is simulated using the direct forcing IB method with subiterations, which will be introduced in Subsection 2.2. The collision among particles, and the collision between the particles and the wall are handled using the discrete element method (DEM), which will be described in Subsection 2.3. The present framework is an extension of the VWiS code [41], which has been successfully applied to turbulent flows in wind energy applications [42, 43]. Apart from the well-tested CURVIB module that includes the flow solver, the new development of the present work includes the module of the diffuse interface IB method with subiterations and the coupling with the DEM library.

Before detailed descriptions for different components of the proposed framework, its overall computational procedure is introduced in the following, where the subscripts n and $n+1$ denote the previous and current time steps, respectively.

1. Enforce the boundary conditions on the complex boundaries of the environment using the CURVIB method based on the flow at step n.

2. Compute the forces on the surface of the particles using the diffuse interface IB method presented in Section 2.2.

3. Advance the discretized Navier-Stokes equations to obtain the flow field at step $n+1$, with the boundary conditions enforced in the first step and the Eulerian forces distributed from the surface of the particles in the second step.

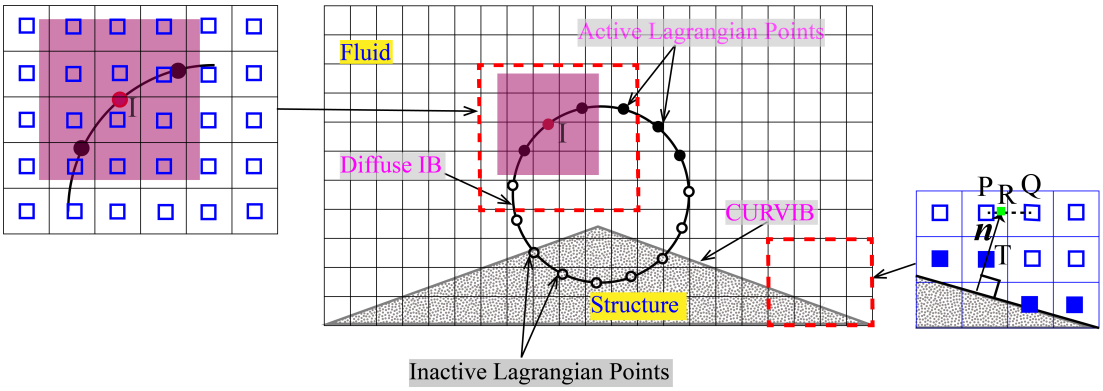


Figure 2: Schematic of the coupling of the sharp interface (zoomed part in the right part) and diffuse interface (zoomed part in the left part) IB methods implemented in two spatial dimensions. The particle is represented by a circle and simulated by the diffuse IB method. The active and inactive Lagrangian points of the diffuse IB are represented by open and filled circles, respectively. The fluid variables are interpolated to cell centers before they are used in IB calculations.

4. Advance the governing equations for the motion of the particles to step $n+1$ using the discrete element method described in Section 2.3.

5. Go to the next time step.

It is noted that although the loose coupling approach is adopted in this study as demonstrated by the above procedure, a strong coupling approach with subiterations within each time step can be directly implemented if necessary.

2.1 Flow solver and the curvilinear immersed boundary method

The governing equations for the flow are the three-dimensional, incompressible Navier-Stokes equations in non-orthogonal, generalized, curvilinear coordinates, which read in compact tensor notation (repeated indices imply summation) as follows ($i,j,l=1,2$ and 3 representing x,y and z directions, respectively):

$$\frac{1}{J}\frac{\partial U^i}{\partial t}=\frac{\xi_l^i}{J}\left(-\frac{\partial}{\partial\xi^j}(U^j u_l)+\frac{\mu_\mathrm{f}}{\rho_\mathrm{f}}\frac{\partial}{\partial\xi^j}\left(\frac{g^{jk}}{J}\frac{\partial u_l}{\partial\xi^k}\right)-\frac{1}{\rho_\mathrm{f}}\frac{\partial}{\partial\xi^j}\left(\frac{\xi_l^j p}{J}\right)+f_l\right), \tag{2.1}$$

$$J\frac{\partial U^j}{\partial\xi^j}=0, \tag{2.2}$$

where J is the Jacobian of the geometric transformation, x_i and ξ^i are the Cartesian and curvilinear coordinates, respectively, $\xi_l^i=\partial\xi^i/\partial x_l$ are the transformation metrics, u_i is the i^{th} component of the velocity vector in Cartesian coordinates, $U^i=(\xi_m^i/J)u_m$ is the contravariant volume flux, $g^{jk}=\xi_l^j\xi_l^k$ are the components of the contravariant metric tensor, ρ_f is the density of the fluid, μ_f is the dynamic viscosity of the fluid, p is the pressure, and f_l are the body forces introduced by the direct forcing IB method for particles.

The governing equations are discretized using a second-order central difference scheme and advanced in time using a second order fractional step method with the second-order Crank-Nicholson scheme. Steps for solving the governing equations to facilitate the description of the IB methods are listed as follows (variables on the Cartesian coordinate system are employed for simplifying the notation):

1. Solve Eq. (2.1) for the intermediate velocity as follows

$$\frac{u_i^*-u_i^n}{\Delta t}=\text{rhs}_i^*+f_i^{n+\frac{1}{2}}, \tag{2.3}$$

in which u_i^* is the intermediate velocity, and

$$\text{rhs}^*=-\frac{u_i^*+u_i^n}{2}\text{Grad}\left(\frac{u_i^*+u_i^n}{2}\right)+\text{Grad}\left(\frac{p^n}{\rho_f}\right)+\frac{\mu_\text{f}}{\rho_\text{f}}\text{Lap}\left(\frac{u_i^*+u^n}{2}\right) \tag{2.4}$$

denotes the collection of the convection term, the viscous term, and the pressure gradient term. Here, Lap and Grad are discretizations of the Laplacian operator ($\Delta=\nabla^2$) and the gradient operator (∇). In this step, the velocity reconstruction and computation of $f_i^{n+\frac{1}{2}}$ for enforcing boundary conditions for the CURVIB method and the direct forcing IB method, respectively, are carried out, in which the latter will be explained in detail in Section 2.2. In the expression for the volume force, i.e., $f_i^{n+1/2}$, the superscript is to denote the fact that it is computed in the direct forcing IB method by explicitly advancing the momentum equation for one time step, which will be explained in Section 2.2. When actually solving the fluid flow, the momentum equation is advanced in time via the Crank-Nicholson scheme, except for the pressure gradient term, which is computed using the values from the previous time step. The discretized momentum equation is solved using a matrix-free Newton-Krylov method [22, 41].

2. Solve the following Poisson equation for satisfying the continuity equation.

$$\text{Lap}(\phi)=\frac{2\rho_\text{f}}{\Delta t}\text{Grad}(u_i^*), \tag{2.5}$$

where ϕ is the pressure increment. In this step, the boundary condition for the pressure is then enforced on cells with neighbors of IB nodes for the CURVIB method. In the present work, the above Poisson equation is solved using the Generalized Minimal Residual (GMRES) method with an algebraic multi-grid acceleration.

3. The velocity and pressure at time step $n+1$ are finally obtained as follows

$$u_i^{n+1}=u_i^*+\frac{\Delta t}{\rho_\text{f}}\text{Grad}(\phi), \tag{2.6}$$

$$p^{n+1}=p^n+\phi. \tag{2.7}$$

In the CURVIB method, the background grid nodes are classified as fluid nodes, solid nodes and IB nodes, in which the IB node is identified as the fluid node with neighbors in the solid. The ray tracing method [44] is employed for the classification of background grid nodes. The velocity at the IB node is reconstructed in the wall normal direction using the velocity at the boundary and the velocity at the reconstruction point [41]. For instance, the velocity at the reconstruction point R can be interpolated using the velocities at points P and Q as shown in Fig. 2. The boundary condition for the gradient of the pressure is applied at the face center between the IB node and the fluid node.

2.2 The diffuse interface direct forcing IB method with subiterations

In the diffuse interface direct forcing IB method, the forces on the Lagrangian points, which represent the geometry of the particle, are computed by directly satisfying the boundary conditions. When the particle collides with the complex geometry, certain Lagrangian points enter/move close to the region of the static structure. In these scenario, we use the 'dry' contact approach of Biegert et al. [39] so that these Lagrangian points are free of calculating the Lagrangian forces and distributing the force to their surrounding Eulerian points. As shown in Fig. 2, the Lagrangian point is set as inactive when its distance to wall is smaller than the band width of the employed discrete Dirac delta function (i.e., $2.5\Delta x$ for the simulated cases)

$$\frac{u_i^{n+1}(\boldsymbol{x})-u_i^n(\boldsymbol{x})}{\Delta t}=\text{rhs}_i^n+\sum_{\boldsymbol{X}\in g_\text{L}}F_i^{n+1/2}(\boldsymbol{X})\delta_h(\boldsymbol{x}-\boldsymbol{X})V_L(\boldsymbol{X}), \tag{2.8}$$

where g_L denotes the collection of the Lagrangian points, $\boldsymbol{X}$ denotes the coordinates of the Lagrangian points, δ_h is the discrete delta function, $V_\text{L}(\boldsymbol{X})$ is the volume of the Lagrangian point. The upper script $n+1/2$ for F indicates the fact that the forces are obtained by advancing the momentum equation explicitly for one step (Eq. (2.11)). The rhs_i^n term collects the convection, viscous and pressure gradient terms at time step n. In the above equation, the force distribution process for computing f_i reads as follows

$$f_i^{n+1/2}(\boldsymbol{x})=\sum_{\boldsymbol{X}\in g_\text{L}}F_i^{n+1/2}\delta_h(\boldsymbol{x}-\boldsymbol{X})V_\text{L}(\boldsymbol{X}). \tag{2.9}$$

Eq. (2.8) is then re-organized as follows

$$\frac{u_i^{n+1}(\boldsymbol{x})-\tilde{u}_i(\boldsymbol{x})}{\Delta t}=\sum_{\boldsymbol{X}\in g_\text{L}}F_i^{n+1/2}(\boldsymbol{X})\delta_h(\boldsymbol{x}-\boldsymbol{X})V_\text{L}(\boldsymbol{X}), \tag{2.10}$$

where the estimated velocity is computed by

$$\tilde{u}_i=u_i^n+\text{rhs}_i^n\Delta t. \tag{2.11}$$

Interpolating the Eq. (2.10) to the Lagrangian points, the following equation is obtained,

$$\frac{U_i^{\mathrm{d}}(\boldsymbol{X})-\tilde{U}_i(\boldsymbol{X})}{\Delta t}=\sum_{\boldsymbol{x}\in g_{\mathrm{b}}}\sum_{\boldsymbol{X}\in g_{\mathrm{L}}}F_i^{n+1/2}(\boldsymbol{X})\delta_h(\boldsymbol{x}-\boldsymbol{X})\delta_h(\boldsymbol{x}-\boldsymbol{X})V_{\mathrm{L}}(\boldsymbol{X})V_{\mathrm{b}}(\boldsymbol{x}), \tag{2.12}$$

where U_i^{d} are the desired velocity on the boundary, $V_{\mathrm{b}}(\boldsymbol{x})$ is the volume of the background grid node, and $\tilde{U}$ is obtained via the velocity interpolation as follows

$$\tilde{U}_i(\boldsymbol{X})=\sum_{\boldsymbol{x}\in g_{\mathrm{b}}}\tilde{u}_i(\boldsymbol{x})\delta_h(\boldsymbol{x}-\boldsymbol{X})V_{\mathrm{b}}(\boldsymbol{x}). \tag{2.13}$$

Eq. (2.12) for F_i can be solved easily in a direct way when the two-point hat function is employed for force interpolation. However, banded matrix cannot be formed easily for discrete delta functions in three spatial dimensions. In this work, we employ the smoothed 4-point discrete delta function proposed by Yang et al. [15] for force distribution and velocity interpolation with a better conservation of the force and the torque and their derivatives, which are of vital importance for moving boundary simulations in order to reduce the non-physical force oscillations. We propose to solve Eq. (2.12) in an iterative way as follows

1. Give an initial guess ($k=0$) of the Lagrangian Force as

$$F_i^k=\frac{U_i^{\mathrm{d}}(\boldsymbol{X})-\tilde{U}_i(\boldsymbol{X})}{\Delta t}, \tag{2.14}$$

   in which the coupling between the force distribution (for computing velocity on the background grid nodes) and the velocity interpolation (for computing forces on the Lagrangian points) is neglected.

2. Distribute forces from the Lagrangian points to the background grid nodes via

$$f_i^k(\boldsymbol{x})=\sum_{\boldsymbol{X}\in g_{\mathrm{L}}}F_i^k\delta_h(\boldsymbol{x}-\boldsymbol{X})V_{\mathrm{L}}(\boldsymbol{X}). \tag{2.15}$$

3. Compute the estimated velocity on the background grid nodes via

$$\tilde{u}_i^{k+1}=u_i^n+\mathrm{rhs}_i^n\Delta t+f_i^k\Delta t. \tag{2.16}$$

4. Compute the estimated velocity on the Lagrangian points via

$$\tilde{U}_i^{k+1}(\boldsymbol{X})=\sum_{\boldsymbol{x}\in g_{\mathrm{b}}}\tilde{u}_i^{k+1}(\boldsymbol{x})\delta_h(\boldsymbol{x}-\boldsymbol{X})V_{\mathrm{b}}(\boldsymbol{x}). \tag{2.17}$$

5. Compute the error via

$$\mathrm{error}=\mathrm{Max}_{\boldsymbol{X}\in g_{\mathrm{L}}}|\tilde{U}_i^{\mathrm{d}}(\boldsymbol{X})-\tilde{U}_i^{k+1}(\boldsymbol{X})|. \tag{2.18}$$

6. If the error is less than a threshold or the allowed maximum number of iteration (which is 6 for the present cases) is achieved, stop the subiteration; otherwise, compute the Lagrangian forces via

$$F_i^{k+1} = \frac{U_i^{\mathrm{d}}(\boldsymbol{X}) - \tilde{U}_i^{k+1}(\boldsymbol{X})}{\Delta t}, \tag{2.19}$$

and go to step 2.

With the Lagrangian forces at each point, the total force and torque on the particle are computed as follows

$$\boldsymbol{F}^{\mathrm{IB}} = -\rho_f \sum_{x \in g_{\mathrm{L}}} \boldsymbol{F}(\boldsymbol{X}) V_{\mathrm{L}}(\boldsymbol{X}) + \rho_{\mathrm{f}} V_{\mathrm{s}} \frac{\boldsymbol{U}_n - \boldsymbol{U}_{n-1}}{\Delta t}, \tag{2.20}$$

and

$$\boldsymbol{T}^{\mathrm{IB}} = -\rho_{\mathrm{f}} \sum_{x \in g_{\mathrm{L}}} (\boldsymbol{X} - \boldsymbol{X}_{\mathrm{c}}) \times \boldsymbol{F}(\boldsymbol{X}) V_{\mathrm{L}}(\boldsymbol{X}) + \frac{\rho_{\mathrm{f}}}{\rho_{\mathrm{s}}} \mathbb{I}_{\mathrm{s}} \frac{\boldsymbol{\omega}_n - \boldsymbol{\omega}_{n-1}}{\Delta t}, \tag{2.21}$$

which will be employed in the discrete element simulations, where the second terms on the right-hand-side of the above equations are correction terms for minimizing the effect of non-physical motion inside the particle [45], which may appear for diffuse interface IB methods.

It is noticed that the explicit Euler scheme employed in the above formulation is not consistent with the temporal scheme for solving the flow in Eq. (2.3), with the objective to reduce the computational cost of solving the momentum equation in each subiteration when computing the Lagrangian force. As will be shown in the validation cases, the proposed approach can predict well the quantities of interest. This approach can also be modified to be consistent with the temporal scheme of the flow solver when explicit schemes are employed for different non-boundary-conforming methods [17,34,35]. Compared with the methods using matrix inversion [32,33], the proposed method has an advantage of no need to carefully distribute the boundary points.

The 'master-to-slave' approach proposed by Uhlmann et al. [46] is adopted for the parallelization of the particle-related calculations. The data of the spheres including the position, velocity of the particles are stored on all processors. The 'master' processor is the processor where the particle centered at gathers the Lagrangian velocity from all slave processors. Then the Lagrangian velocities are distributed to the 'slave' processors from the 'master' processor. We then calculate the Lagrangian forces in each 'slave' processor and use the Dirac delta function to distribute the Lagrangian forces to local background nodes. Note that the 'slave' processors should contain at least one Lagrangian node belonging to the specific particle. When the particle is only within one processor, there are no 'slave' processors. In the cases considered in this work, only Cartesian grids are considered to discretize the fluid domain and the whole domain is partitioned into multiple cuboids.

2.3 Discrete element method for particle motion and collision

The motion of the particle is governed by Newton's second law and the collision between particles is handled by using the discrete element method implemented in the open source code Yade [47].

2.3.1 Damping

During particle-wall collision (the gap distance between the particle and the wall is within $[0,\Delta x]$, in which Δx is the smallest space step of the fluid solver), numerical damping is added to dissipate the kinetic energy.

$$\tilde{\boldsymbol{F}}^{\text{all}} = \boldsymbol{F}^{\text{c}} + \boldsymbol{F}^{\text{l}} + \boldsymbol{F}^{\text{IB}}, \tag{2.22}$$

$$\boldsymbol{F}^{\text{all}} = \left(1 - \eta^{\text{d}} \text{sgn}\left(\tilde{\boldsymbol{F}}^{\text{all}}\left(\boldsymbol{U} + \frac{\boldsymbol{a}\Delta t}{2}\right)\right)\right)\tilde{\boldsymbol{F}}^{\text{all}}, \tag{2.23}$$

in which η^{d} is the coefficient of damping.

2.3.2 Solid motion

The governing equations for the motion of the particle are

$$\frac{\text{d}\boldsymbol{U}}{\text{d}t} = \boldsymbol{a}, \tag{2.24}$$

$$\frac{\text{d}\boldsymbol{X}}{\text{d}t} = \boldsymbol{U}, \tag{2.25}$$

$$\frac{\text{d}\boldsymbol{w}^{\circ}}{\text{d}t} = \boldsymbol{w}, \tag{2.26}$$

$$\boldsymbol{W} = \boldsymbol{R}(\boldsymbol{w}^{\circ})\boldsymbol{W}, \tag{2.27}$$

in which $\boldsymbol{R}$, $\boldsymbol{w}^{\circ}$, $\boldsymbol{w}$ and $\boldsymbol{W}$ are the rotation matrix, the angular velocity, the angular acceleration and the orientation angle, respectively. The translational acceleration can be obtained via

$$\boldsymbol{a} = \frac{\boldsymbol{F}^{\text{all}}}{m_{\text{s}}} + \frac{\rho_{\text{s}} - \rho_{\text{f}}}{\rho_{\text{f}}}\boldsymbol{g}, \tag{2.28}$$

where $\boldsymbol{g}$ is the gravity acceleration, m_s is the mass of the particle, and $\boldsymbol{F}^c$ is the force due to the contact of particles. For rotational motions of a particle,

$$\mathbb{I}\boldsymbol{w} = \boldsymbol{T}^{\text{c}} + \boldsymbol{T}^{\text{IB}}, \tag{2.29}$$

in which $\mathbb{I}$ is a diagonal moment of inertial tensor with $\mathbb{I}_{xx} = \mathbb{I}_{yy} = \mathbb{I}_{zz}$ and $\boldsymbol{T}$ is the torque.

The "leapfrog" method is employed for advancing the translational motion of the particle, such that the displacement and the translational velocity are updated via

$$\boldsymbol{U}_{\overleftarrow{n}+\frac{\overleftarrow{1}}{2}} = \boldsymbol{U}_{\overleftarrow{n}-\frac{\overleftarrow{1}}{2}} + \boldsymbol{a}_{\overleftarrow{n}}\overleftarrow{\Delta t}, \tag{2.30}$$

$$\boldsymbol{X}_{\overleftarrow{n}+\overleftarrow{1}} = \boldsymbol{X}_{\overleftarrow{n}} + \boldsymbol{U}_{\overleftarrow{n}+\frac{\overleftarrow{1}}{2}}\overleftarrow{\Delta t}, \tag{2.31}$$

where the $\leftarrow$ symbol over the variables represent the time scale used in DEM calculations. Typically, $\frac{\Delta t}{\overleftarrow{\Delta t}} \sim \mathcal{O}(100)$. Then the angular velocity is updated by

$$\boldsymbol{\omega}^{\circ} = \boldsymbol{\omega}^{\circ}_{\overleftarrow{n}+\frac{\overleftarrow{1}}{2}} = \boldsymbol{\omega}^{\circ}_{\overleftarrow{n}-\frac{\overleftarrow{1}}{2}} + \overleftarrow{\Delta t}\,\boldsymbol{w}_{\overleftarrow{n}}. \tag{2.32}$$

Therefore we can define a quaternion with four components

$$\Delta \boldsymbol{W} = \cos(\Delta W) + \frac{\sin(\Delta W)}{\sqrt{3}}\boldsymbol{i} + \frac{\sin(\Delta W)}{\sqrt{3}}\boldsymbol{j} + \frac{\sin(\Delta W)}{\sqrt{3}}\boldsymbol{k}, \tag{2.33}$$

in which $\Delta W = ||\frac{\boldsymbol{\omega}^{\circ}\Delta t}{2}||$ and $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ are unit vectors in three spatial axes. Finally the quaternion can be updated as

$$\boldsymbol{W}_{\overleftarrow{n}+\overleftarrow{1}} = \Delta \boldsymbol{W}_{\overleftarrow{n}+\overleftarrow{1}}\,\boldsymbol{W}_{\overleftarrow{n}}. \tag{2.34}$$

It should be noted that there is a half time step mismatch for the "leapfrog" algorithm described above. However, the time step for the particle motion is about two orders of magnitude smaller than the fluid. Therefore, the effect of the mismatch in time can be considered to be negligible. At the initial fluid time step, $\boldsymbol{U}_{\overleftarrow{n}-\frac{\overleftarrow{1}}{2}}$ is either prescribed or set to be zero for the particle at rest.

2.3.3 Contact model

The detection of the collision of spheres and bounding boxes employs the 'sweep and prune' algorithm of $\mathcal{O}(N\log N)$ complexity with the collision model proposed by Cundall and Strack [48]. The normal force is calculated by

$$\boldsymbol{F}^{\mathrm{c}}_{N} = K_N \boldsymbol{X}^{pq}_{N}, \tag{2.35}$$

where K_N is the normal stiffness, $\boldsymbol{X}^{pq}_{N} = -\delta^{pq}(\boldsymbol{X}^{q}-\boldsymbol{X}^{p})/||\boldsymbol{X}^{q}-\boldsymbol{X}^{p}||$ is the relative displacement between particle p and particle q in the normal direction of contact with $\delta^{pq} = ||\boldsymbol{X}^{q}-\boldsymbol{X}^{p}|| - D$. Here, D represents the diameter of the sphere.

To obtain the tangential force, we need to calculate the trial shear force first by

$$\boldsymbol{F}^{\mathrm{t}}_{T} = K_S \boldsymbol{X}^{pq}_{T}, \tag{2.36}$$

where K_S is the shear stiffness, which is related with the normal stiffness by $K_S = \frac{K_N}{2(1+\mu_{\mathrm{s}})}$, in which μ_{s} is Poisson's ratio and is chosen to be 0.45 in this study. Then the tangential force is calculated by

$$\boldsymbol{F}^{\mathrm{c}}_{T} = \begin{cases} \boldsymbol{F}^{\mathrm{t}}_{T}\frac{|\boldsymbol{F}_N|\tan\phi}{|\boldsymbol{F}^{t}_{T}|}, & \text{if } |\boldsymbol{F}^{\mathrm{t}}_{T}| > |\boldsymbol{F}_N|\tan\phi, \\ \boldsymbol{F}^{\mathrm{t}}_{T}, & \text{otherwise}, \end{cases} \tag{2.37}$$

in which ϕ is the friction angle. As the contact force ($\boldsymbol{F}^{\mathrm{c}} = \boldsymbol{F}^{\mathrm{c}}_{N} + \boldsymbol{F}^{\mathrm{c}}_{T}$) acts at the contact point, torques generated by the contact forces should be considered, leading to

$$\boldsymbol{T}^{\mathrm{c}} = d^{\mathrm{c}}(-\boldsymbol{n}) \times \boldsymbol{F}^{\mathrm{c}}, \tag{2.38}$$

in which d^{c} is the distance between the sphere center and the contact point.

3 Benchmark simulations

In this section, the accuracy and robustness of the numerical methods presented in Section 2 are validated by simulating several benchmark problems. We first verify the variant of the diffuse interface IB method in simulating problems involving static and moving boundaries in Sections 3.1-3.2. Then the capability of the coupling of the diffuse interface IB method and the DEM method in simulating particle to particle collision is shown in Section 3.3. In Section 3.4, the accuracy of the coupling between the CURVIB, the diffuse IB and the DEM methods are verified. As the forces are distributed to the Eulerian grid nodes close to the immersed boundaries in the diffuse interface IB method, the effective radius of the spherical particle is larger than the actual one. For the cases presented in Sections 3.2-3.4, the input diameter used for discretizing the surface of the particle is chosen as $D_{\text{input}} = D - \Delta x$ (which is similar to the treatment in the literature [35]), where Δx is the grid spacing of the Eulerian grid. Note that in Section 3.1 the actual diameter is employed, i.e., $D_{\text{input}} = D$, is used in order to examine the order of accuracy of the proposed diffuse interface IB method, where the number of grid nodes for discretizing the diameter is large for the fine grid case that the above effect is not significant.

3.1 Flow around a sphere

Flow around a sphere is a typical benchmark problem for testing numerical methods designed to simulate immersed boundaries in fluids. To validate the proposed diffuse interface IB method, we consider a sphere with diameter $D=1$ centered at $(15D,15D,10D)$. The physical domain is chosen as $[0,30D]\times[0,30D]\times[0,30D]$. A uniform inflow velocity $U=(0,0,1)$ is imposed on $z=0$, and the convective boundary condition is imposed at $z=15$ as an outflow condition. Along the other boundaries, the freeslip boundary condition is specified. We set $\rho_{\text{f}}=1$ and use the inflow velocity U as the characteristic velocity so that the Reynolds number is defined as $Re=\frac{\rho_{\text{f}}UD}{\mu_{\text{f}}}$. After discretization, the sphere diameter is $D=40\Delta x$ and the surface of the sphere is discretized via triangular mesh with $\Delta X\approx\Delta x$, where Δx is the smallest background mesh size. The computational domain is discretized by using a nonuniform orthogonal mesh of $281\times281\times421$. We use a fine region of $3D\times3D\times1.25D$ by uniformly distributed meshes of $\Delta x=\frac{D}{40}$. The CFL number is chosen as 0.8. The drag coefficient, lift coefficient and the Strouhal number are defined as $C_{\text{D}}=\frac{F_z}{0.5\rho U^2A}$, $C_{\text{L}}=\frac{F_y}{0.5\rho U^2A}$ and $St=\frac{fD}{U}$, respectively, where $A=\frac{\pi D^2}{4}$ and f is the frequency of the drag coefficient.

Two cases with Reynolds numbers of Re=100 and Re=300 are considered. At Re=100, the flow field ends in a steady state, a comparison of the drag coefficients obtained by the proposed variant of the direct forcing IB method and the reference values from the literature are shown in Table 1. It is seen that the present prediction agrees well with those published results. An instantaneous vorticity field and streamline for the case of unsteady flow at $Re=300$ is shown in Fig. 3, where the vorticity field is represented by

Table 1: Comparison of C_D, C_L and St with reference values at $Re=100$ and 300.

References		$Re=100$	
		C_D	
Johnson and Patel [49]		1.10	
Fadlun et al. [50]		1.08	
Wang and Zhang [33]		1.13	
Present		1.136	
		$Re=300$	
	$\overline{C_D}$	$\overline{C_L}$	St
Johnson and Patel [49]	0.656	0.069	0.137
Kim et al. [51]	0.657	0.067	0.134
Wang and Zhang [33]	0.680	0.071	0.135
Present	0.678	0.062	0.131

Table 2: List of the drag coefficient and the order of accuracy of the diffuse interface IB method for simulating flow around a circular cylinder.

D	C_D	Or_s
$20\Delta x$	1.163	
$30\Delta x$	1.140	4.35
$40\Delta x$	1.136	

the Q criterion. The Q value is defined as $Q=\frac{1}{2}(\boldsymbol{\Omega}:\boldsymbol{\Omega}-\boldsymbol{S}:\boldsymbol{S})$, where $\boldsymbol{\Omega}=\frac{1}{2}(\nabla\boldsymbol{u}+\nabla\boldsymbol{u}^{\mathrm{T}})$ and $\boldsymbol{S}=\frac{1}{2}(\nabla\boldsymbol{u}-\nabla\boldsymbol{u}^{\mathrm{T}})$ are the rate of strain and the rate of rotation tensors. This criterion implies that the vortex core is the region where the strength of rotation overcomes the strain. The evolution of C_D against time is shown in Fig. 4. We also present the average drag and lift coefficients and the Strouhal number (St) in Table 1. Again, the present results are in good agreement with the results reported in the literature. The order of accuracy is defined as [52]

$$\mathrm{Or_s}=\log_{D_c/D_f}\left(\frac{e_c}{e_f}\right), \tag{3.1}$$

in which D_c and D_f corresponds to the number of Eulerian grid points per diameter of the sphere for the coarse and the fine meshes, respectively, e_c and e_f are the errors from the coarse and the fine meshes. Here, the errors are calculated with reference to the solution from the finest mesh case using the Richardson extrapolation method. From Table 2, it is seen that the order of accuracy for the drag coefficient is 4.35. Because C_D is quadratic to the velocity, the error of the velocity should be approximately 2.09. Therefore, the spatial order of accuracy of the proposed diffuse IB method is "formal" second-order.

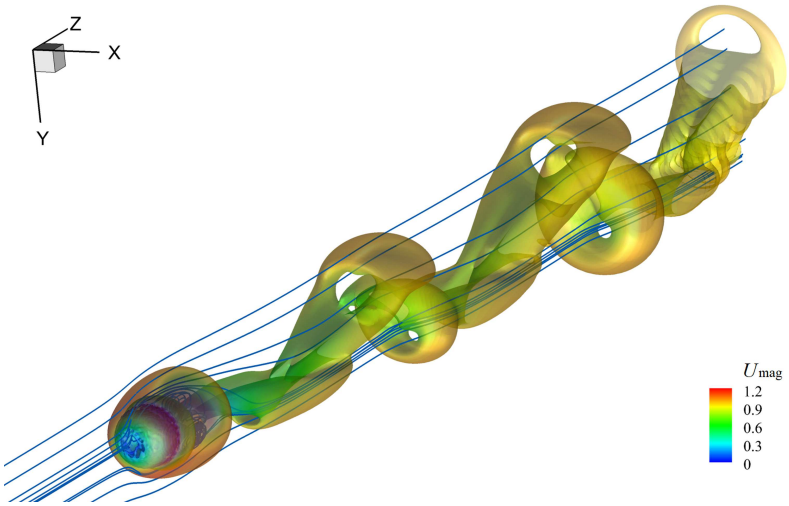


Figure 3: Instantaneous vorticity fields for flow around a sphere by means of isosurface of the Q-criterion at $Re=300$. The isosurface is colored by the magnitude of the velocity (U_{mag}).

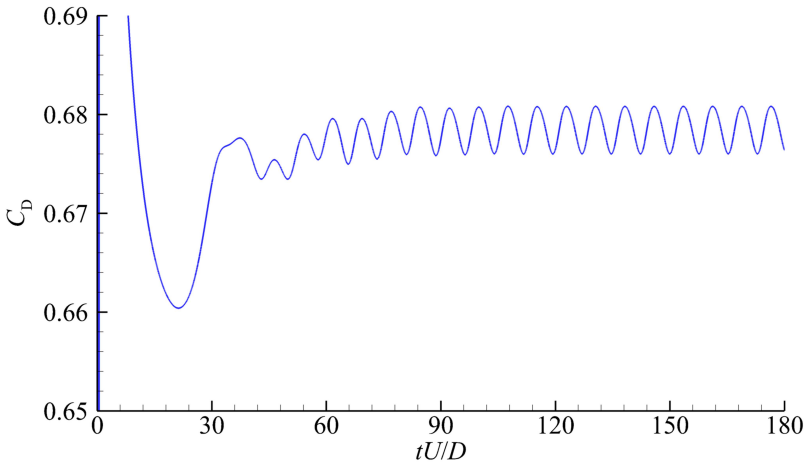


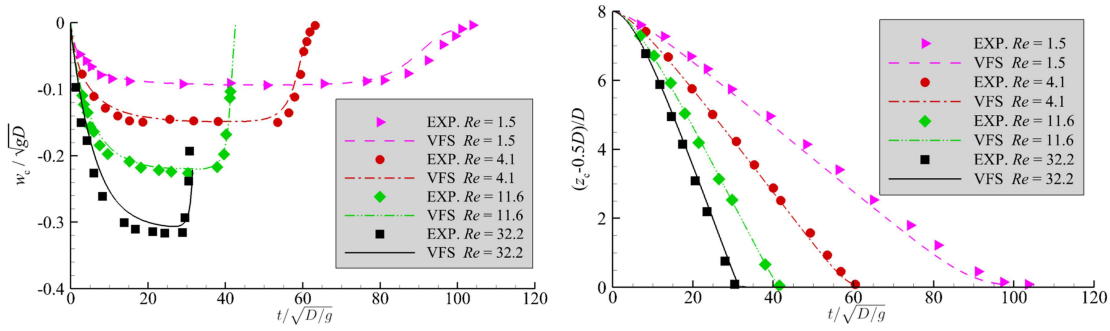
Figure 4: Evolution of the drag coefficient against time for flow around a sphere at $Re=300$.

3.2 Sedimentation of a sphere in a quiescent viscous fluid

The sedimentation of a sphere particle in an initially quiescent fluid region is simulated by using the proposed variant of the diffuse interface IB method. At $t=0$, the sphere is centered at [5D, 5D, 8D] with the fluid domain of $[0,10D]\times[0,10D]\times[-0.5D,11.5D]$, where $D=1$ is the diameter of the sphere. In the simulation, we use $\Delta X=\Delta x=D/16$. The parameters that govern the system are shown in Table 3. The CFL number is 0.6 for $Re=32.2$ and is decreased when the Reynolds number becomes smaller. As shown in Fig. 5(a) for the displacement of the center of the sphere, the time consumed from the

Table 3: Dimensionless parameters for the sedimenting sphere.

Re	ρ_s	μ_f	ρ_f	g
1.5	1.155	0.067	1	-1
4.1	1.161	0.038	1	-1
11.6	1.164	0.020	1	-1
32.2	1.167	0.011	1	-1



(a) (b)

Figure 5: Comparison of the velocity and displacement of the particle during sedimentation for various Reynolds numbers for our present simulation and the results of Cate et al. [53].

initial release position to the bottom of the domain becomes less as the Reynolds number increases. From Fig. 5(b), the sphere reaches the terminal velocity for $Re=1.5$, 4.1 and 11.6 while the magnitude of the velocity in the z direction still grows for $Re=32.2$. It can also be concluded from Fig. 5 that the present numerical results are in good agreement with the experimental results of Cate et al. [53].

3.3 Collision of two sedimenting spheres in a confined channel

When multiple particles move in the same fluid domain, collision between the particles are common. In this section, we study the collision of two sedimenting spheres in a confined channel.

The dimensionless variables are given as follows. The computational domain is a box of $[0,6D]\times[0,6D]\times[0,24D]$ with rigid walls on all the boundaries of the domain. The gravity is $g=-1$. The densities of the sphere and the fluid are 1.14 and 1, respectively. The dynamic viscosity of the fluid is $\nu=4.68\times10^{-3}$. In this simulation, the stiffness is chosen as $K_N=\frac{100m_s||g||}{D}$. The initial positions of the leading and trailing particles are located $(2.97D,2.97D,18.96D)$ and $(3.03D,3.03D,21D)$, respectively, with the initial vertical distance between the two particles being equal to $2.04D$, and the initial distances in the x and y directions $0.06D$ to trigger the drafting−kissing−tumbling (DKT) process. The

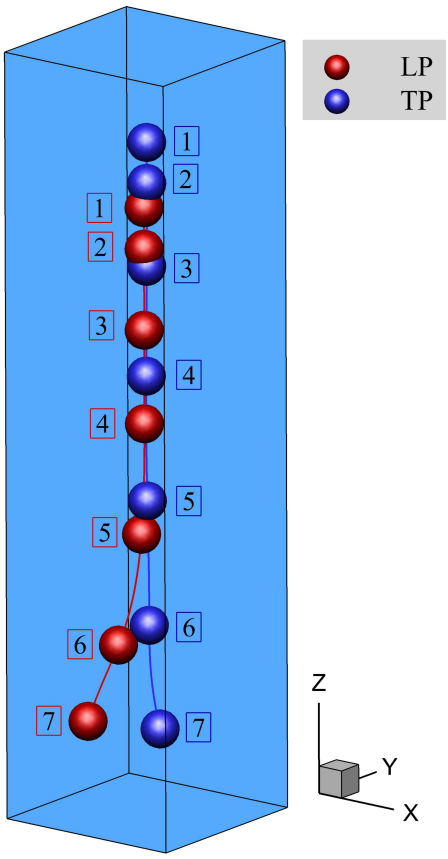
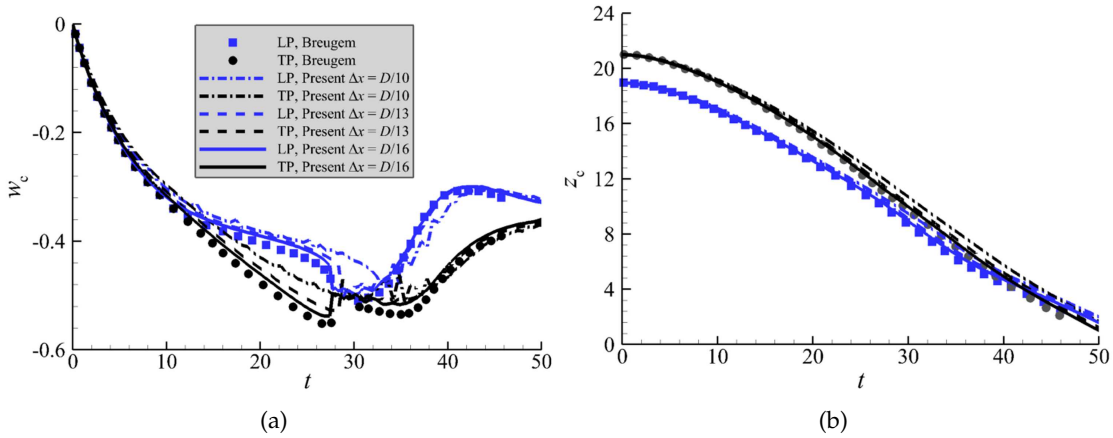


Figure 6: Trajectories of two sedimenting spheres undergoing the DKT process. The numbers in the square are added to identify the order of the sedimentation process.

CFL number is chosen as 0.5. The sedimentation process of the two spheres are shown in Fig. 6, where the leading and trailing particles are colored in red and blue, respectively. We compare the obtained results with Breugem [35] in Fig. 7. As seen, the curves of the velocity and the displacement for the three different meshes of $\Delta x = D/10$, $D/13$ and $D/16$ are similar to each other, and the results for the latter two meshes are very close. Here, Δx is the smallest background mesh size. For $\Delta x = D/16$, after an initial drafting period, the two particles come close to each other and "kiss" at around $t = 28$ when their velocities become almost identical. Then the two spheres separate and "tumble" in their own paths without further colliding.

3.4 Collision between a particle and a flat wall

In Section 3.3, the proposed diffuse IB method with DEM is verified for particle-particle collisions. In this part, the experiment by Gondret et al. [54] is chosen to examine the ac-



(a) (b)

Figure 7: Comparison of the velocity and displacement of the leading particle and the trailing particle during the DKT sedimentation process obtained by our present simulation and the numerical results of Breugem [35].

Table 4: Dimensionless parameters for the bouncing particle case.

Case	$St_c=27$	$St_c=152$
U_{in}	1	1
D	1	1
ρ_s/ρ_f	8.083	8.342
g	-0.219	-0.086
ρ_f	1	1
ν_f	0.033	0.0061
Domain size	$12D\times12D\times24D$	$6D\times6D\times48D$

curacy of the present method in predicting particle-wall collision, in which the flat wall and the particle are simulated using the CURVIB method and the diffuse IB method, respectively. In this simulation, the gravity is assumed to be in the z direction. The velocity of the particle is prescribed to be $(0,0,-U_{in}(e^{-2tU_{in}/D}-1))$ before the particle is released to move freely at $z_c=D$, in which D is the diameter of the particle and U_{in} is the impact velocity. The impact Stokes number is defined as $St_c=\frac{\rho_s U_{in} D}{9\mu_f f}$, in which f is the vortex shedding frequency. Here, the cases of $St_c=27$ and 152 are simulated. The normal stiffness K_N is chosen as $\frac{1000m_s||g||}{D}$ for these two cases. The other parameters are listed in Table 4. Fig. 8 presents both the trajectories of the particle obtained by Gondret et al. in laboratory experiments and our present numerical simulations with $\Delta x=D/10$, $D/13$ and $D/16$ for the above mentioned two Stokes numbers, in which the reference time scale is defined as $t_{ref}=\sqrt{D/||g||}$. For $St=27$ and $St=152$, the coefficient of damping is set to $\eta^d=0.2$ and 0, respectively. As the mesh is refined, the result obtained by using the present coupling framework becomes closer to the experimental result of Gondret et al. [54]. As expected,

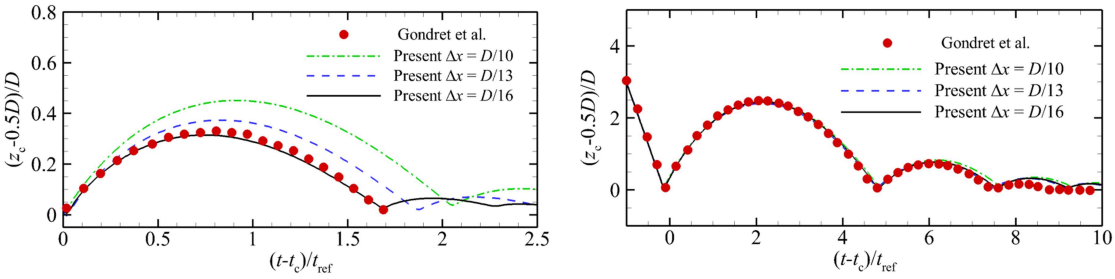
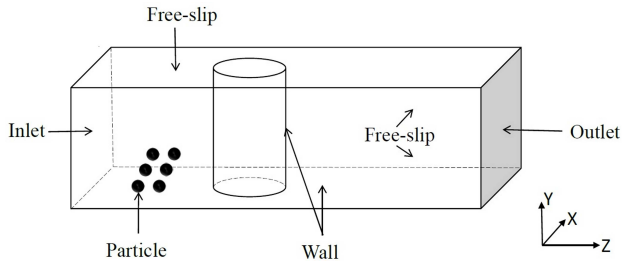


Figure 8: Comparison of the displacements obtained by using the present method with different meshes and the experimental results of Gondret et al. [54]. (a) $St_c=27$; (b) $St_c=152$.

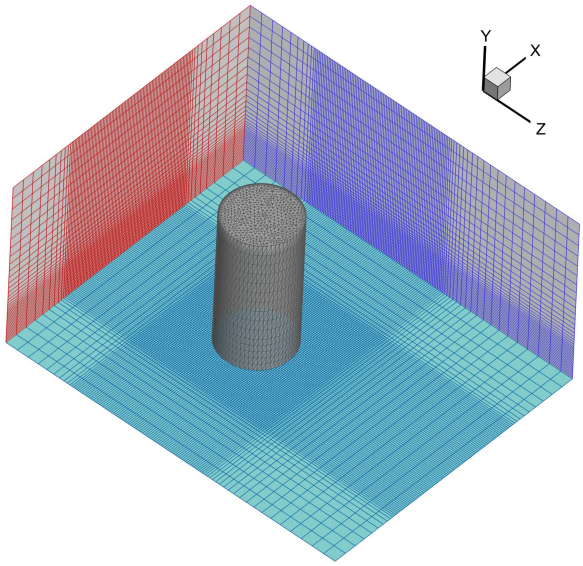
for the greater Stokes number case, the particle motion is less affected by the fluid, resulting in a relative longer bouncing back distance from the wall. In addition, the results shown in the figure indicate that the experimental and numerical results agree with each other.

4 Interaction between multiple particles and a wall-mounted cylinder

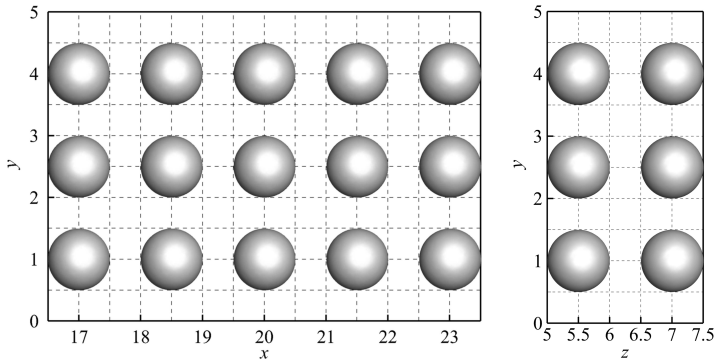
This section demonstrates that the methods used in this paper are suitable for simulating the collision between multiple particles and complex geometries. We apply the proposed method to simulate uniform flow around a vertical cylinder in the presence of multiple particles. The schematic of this case is shown in Fig. 9(a). The initial velocity of the flow field (U) is set to be 1. The diameter of the particle and the base of the vertical cylinder are defined as d and D, respectively. In this study, we choose $d=1$ and $D=10$. The density of the fluid and the particle are 1 and 2.3, respectively. The kinematic viscosity (ν_f) of the fluid is 0.05 so that the Reynolds number defined as $Re=\frac{UD}{\nu_f}$ is 200. In the simulation, we use $d=20\Delta x$, where Δx is the smallest background mesh size. The computational domain is a $[0,40d]\times[0,25d]\times[0,50d]$ box. On the left boundary, we apply an uniform flow as inlet, and on the right boundary, we impose an outflow boundary condition. At the front and back faces, the free-slip boundary condition is applied. The bottom boundary is set as the no-slip wall, and the top boundary is set as free-slip boundary. The vertical cylinder is centered at $(20d,12.5d,20d)$. The height of the cylinder is $25d$. The mesh used for this problem is displayed in Fig. 9(b) and the arrangement of the initial positions for the particles are shown in Fig. 9(c). We refine the mesh in a region of $[10d,30d]\times[0,5d]\times[10d,30d]$ where the interaction among particles, and the collision between the particle and the cylinder appear. Fig. 10 shows four typical snapshots of the simulated instantaneous flow structures, which are identified using the Q criterion with $Q=0.005$ iso-surface. As shown in Fig. 10(a) when the simulation starts for a short



(a)



(b)



(c)

Figure 9: (a) Schematic of the computational setup, (b) mesh of the cylinder and the fluid domain, and (c) initial position of the particles for the case of uniform flow around a vertical cylinder in the presence of multiple particles. The mesh in the figure shows every 4th grid lines.

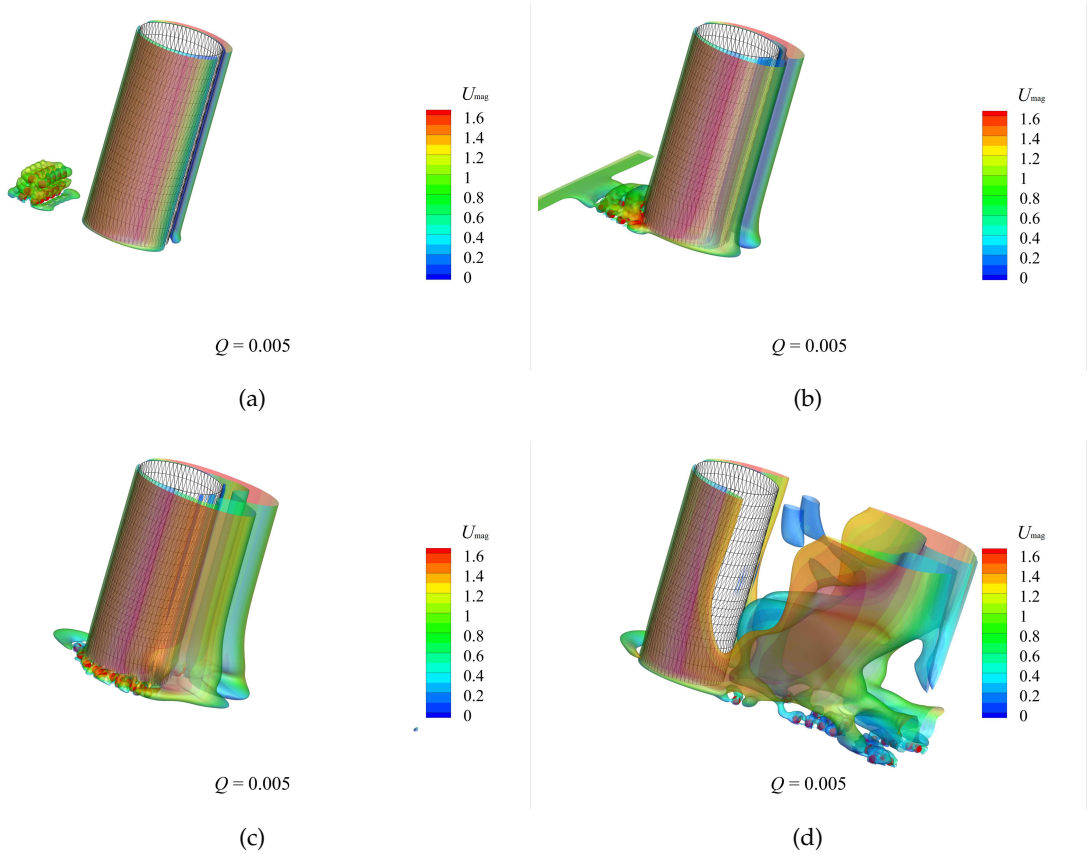


Figure 10: Flow structures identified using the Q criterion for the case of uniform flow around a vertical cylinder in the presence of multiple particles, where the particles are colored using the magnitude of the velocity. (a) $tU/d=2.5$; (b) $tU/d=7.5$; (c) $tU/d=17.5$; (d) $tU/d=65.0$.

period, a portion of the particles have not touched the bottom boundary. The particles are moving in the position z direction due to the convection of the surrounding fluid. Furthermore, the particles are sedimenting in the y direction due to the gravity force, which is stronger than the buoyancy force. At $tU/D=7.5$ in Fig. 10(b), some particles collide with the cylinder demonstrating the capability of the present code to simulate the interaction between the particle and the marine structure with complicated geometry. As can be seen in Fig. 10(c), the particles move along the cylinder and collide with each other. At the same time, the horseshoe vortex forms in the front of the cylinder. Fig. 10(d) shows the flow structure after particles left the cylinder. As seen, the wake and vorticity fields around the particles complicate the flow fields near the bottom wall.

Fig. 11 shows the trajectories of particles. When the particles move towards the cylinder, some particles collide with other particles instead of interacting with the cylinder. For the trajectories behind the cylinder, the particles move towards the center of the do-

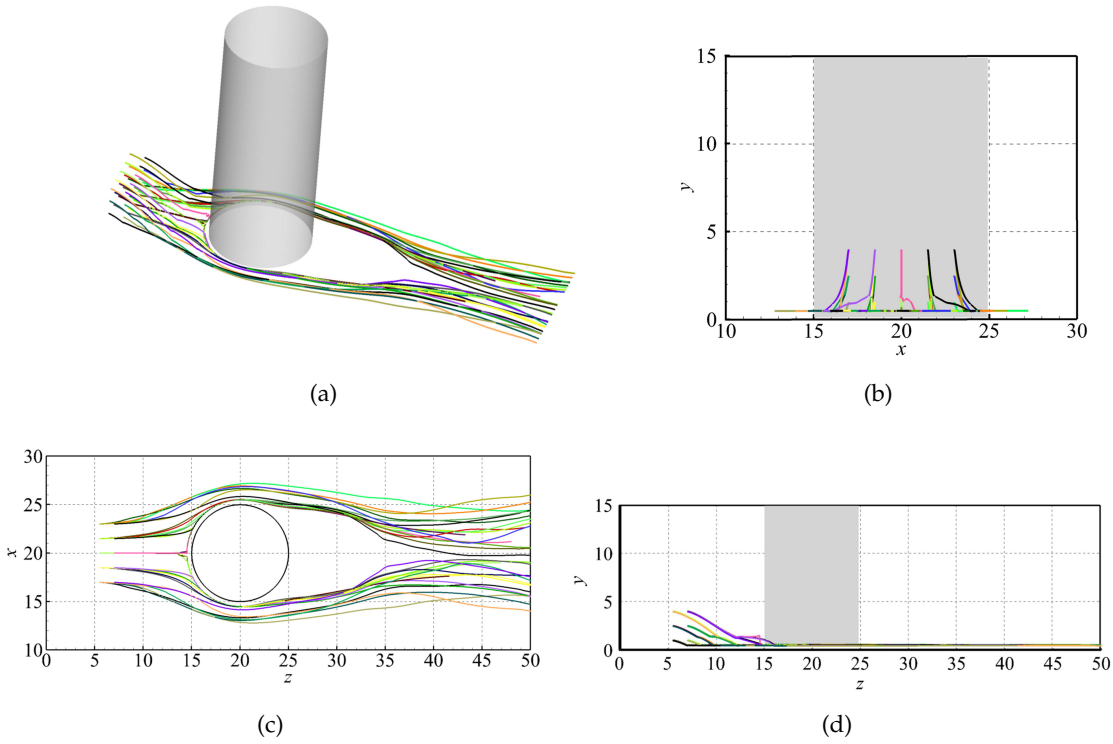


Figure 11: Trajectories of the particles for the case of uniform flow around a vertical cylinder in the presence of multiple particles. (a) Trajectories in three spatial dimensions; (b) trajectories in the x-y plane; (c) trajectories in the x-z plane; (d) trajectories in the y-z plane.

main in the x direction. This is because there exists a low pressure zone behind the cylinder. Due to the collision between the particles, the trajectories of the cylinder 'diverge' for $z>40$.

5 Particle-laden turbulent flows over periodic hills

Particle-laden turbulent flows involve rich mechanism to explore [55, 56] and are beneficial to obtain the drag law for turbulence models [9]. In this section, we apply the proposed method to simulate particle-laden turbulent flows over periodic hills.

A schematic of the simulated case is shown in Fig. 12, for which the setup is the same that in the literature [57,58]. As shown in the figure, two half parts of a hill of height h are placed at two ends in the x direction of the computational domain. The distance between the crests of two hills is $L_x=9h$, and a flat wall is placed at $2.036h$ above the crest of the hill so that the extent in the y direction is $L_y=3.036h$. In the spanwise direction, the extent of the computational domain is $L_z=4.5h$. Periodic boundary conditions are imposed along the streamwise and spanwise directions. The no-slip boundary conditions are applied

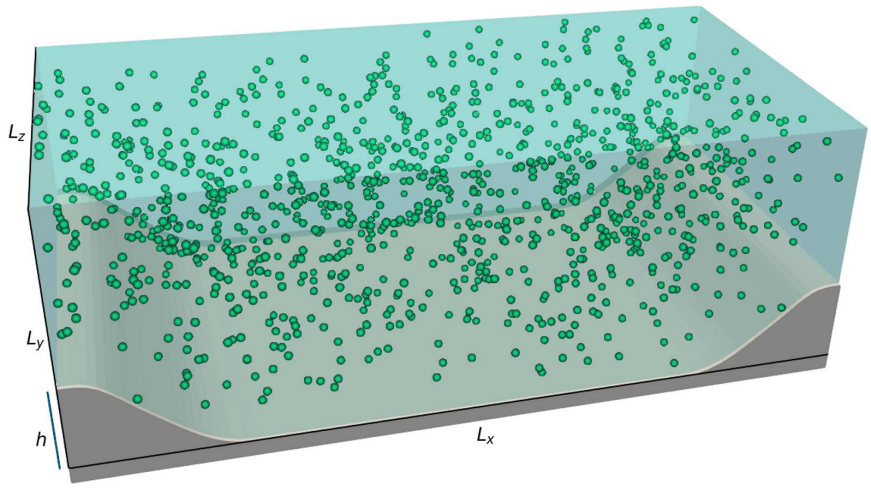


Figure 12: Schematic of particle-laden flow over periodic hills.

along the top wall and the surface of the hill. The Reynolds number defined using the height of the hill, the dynamic viscosity (ν) and the bulk velocity U_b is $Re_h = \frac{\rho_f U_b h}{\mu_f} = 700$, in which $U_b = \frac{Q_m}{\rho_f L_z (L_y - h)}$ with Q_m the mass flux. To maintain a constant mass flux, the flow is driven by a mean pressure gradient in the streamwise direction, which is uniformly applied in the whole domain. Neutrally buoyant particles with their densities the same as the fluid are considered. The diameter of the particle is set as $D = 0.1h$, such that the particle Reynolds number is $Re_p = \frac{\rho_f U_b D}{\mu_f} = 70$. The volume fraction of the particle to the fluid is $\phi_s = 0.5\%$, resulting in a total number of 1088 particles. The numbers of grid nodes are uniformly distributed with $N_x \times N_y \times N_z = 901 \times 302 \times 451$ in all three directions, which means that there are 10 grid cells resolving the diameter of the particle. Two cases with/without particles are carried out.

Comparisons between the results from the present simulation using the CURVIB method and the results of Breuer et al. [57] using a body-fitted mesh are presented in the Appendix A, which show the capability of the CURVIB method and the employed setup in predicting the key flow characteristics of the flow over periodic hills.

The instantaneous vorticity field showing the complicated flow structures and particle-vortex interaction is presented in Fig. 13. Fig. 14 gives the time-averaged streamwise velocity field with the streamline. Fig. 15 compares the vertical profiles of different flow statistics from the case with particles with that without particles at different streamwise locations, i.e., $x/h = 0.05, 0.5, 1, 2, 3, 4, 5, 6, 7$ and 8. It is seen from Fig. 15(a) that the streamwise velocities from the cases with/without particles are nearly the same at all considered streamwise locations except for $x/h = 4$. In the lower half region of this location, the magnitudes of the negative streamwise velocity from the case with particles are

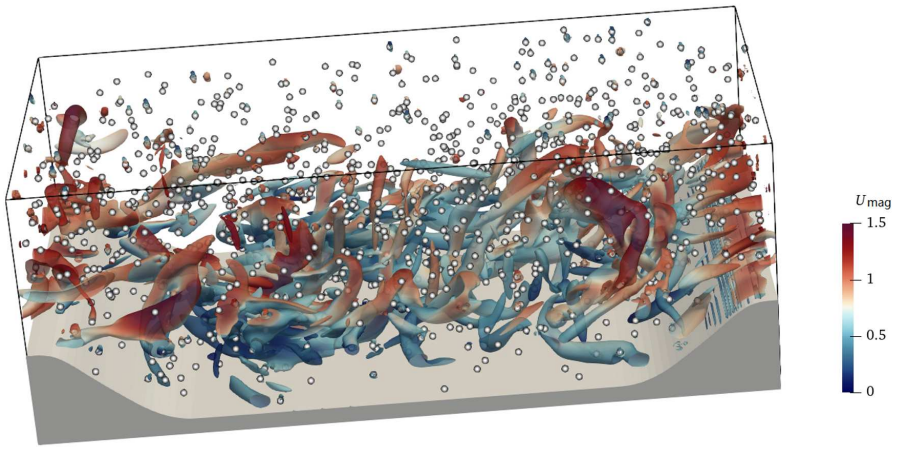


Figure 13: Instantaneous vorticity fields colored by the velocity magnitude (U_{mag}) for particle-laden flow over periodic hills.

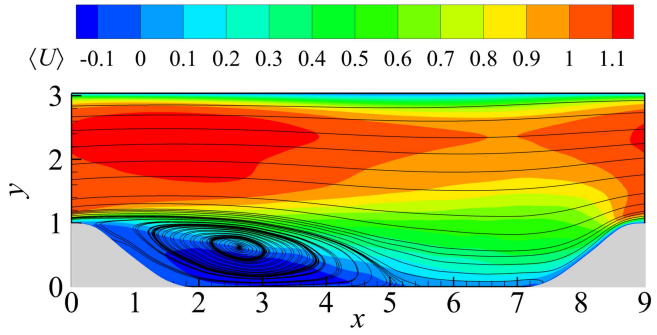


Figure 14: Time-averaged streamlines for flow over periodic hills with multiple particles.

smaller than those without particles. The above observation indicates that the neutrally buoyant particles decrease the mean convection in the streamwise direction in this critical region, which results a somewhat longer recirculation bubble for the case with particles. As shown in Fig. 15(b, c) for the vertical velocity and the primary Reynolds shear stress, the particle has minor effect on these two properties. Comparisons of the turbulence kinetic energy (TKE) are shown in Fig. 15(d). Differences for the TKE between cases with and without particles are observed especially at $x/h=3,4$. It is seen that the TKE is in general decreased in the presence of particles. Detailed analysis on the mechanism of particle-laden turbulence in flows over periodic hills will be carried out in our future study.

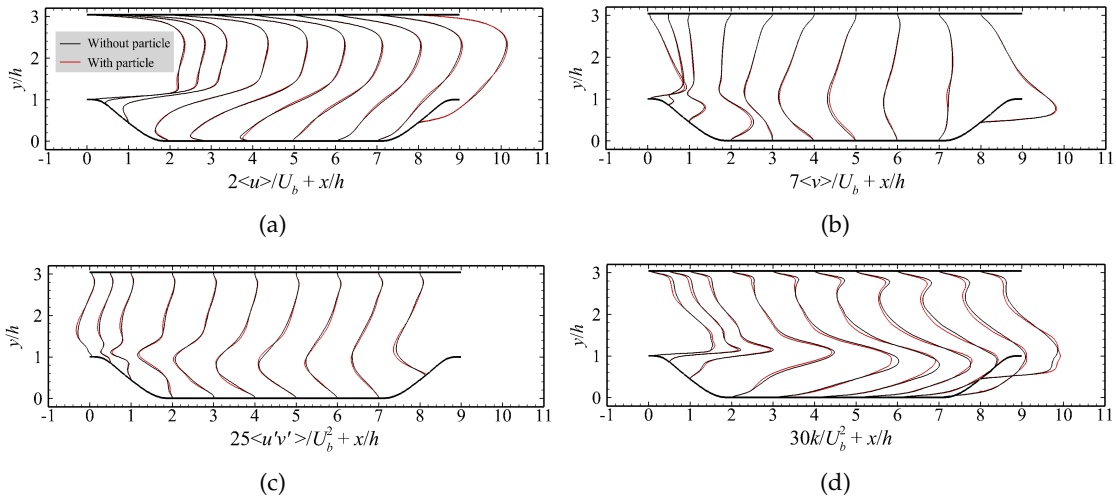


Figure 15: Comparison of vertical profiles from flow over periodic hills without and with particles. (a) Time-averaged streamwise velocity $\langle u\rangle$; (b) time-averaged vertical velocity $\langle v\rangle$; (c) primary Reynolds shear stress $\langle u'v'\rangle$; (d) turbulent kinetic energy $k=0.5\langle u'u'+v'v'+w'w'\rangle$.

6 Conclusions

In this work, a framework for simulating particle-resolved particle-laden flows considering particle-particle and particle-wall collisions is proposed, in which the curvilinear immersed boundary (IB) method is employed for the stationary complex boundaries, a variant of the direct forcing IB method is developed for simulating the boundary of the rigid particle, and the particle-particle collision and the particle-wall collision are handled with the discrete element method. The main advantage of the present framework is that it is capable of handling the collision between the particle and the complex boundaries of the environment conveniently without the need for the fluid mesh to conform to the complex geometry. Furthermore, the so called 'dry' contact approach is involved in the proposed framework to avoid the cohesion of the particle to the boundary. In the proposed diffuse interface direct forcing IB method, an interactive approach is employed to take into account the coupling between the velocity interpolation and the force distribution processes in order to satisfy the no-slip boundary conditions in a more accurate way.

We first validate the proposed direct forcing IB method with iteration by simulating the flow around a sphere at Reynolds number of 100 and 300 and sedimentation of a sphere in a quiescent fluid. We then evaluate the capability of the proposed method in predicting the particle-particle and particle-wall collisions using two benchmark problems including the drafting-kissing-tumbling of two particles during sedimentation, and the collision between a particle and a flat wall, with good accuracy for both cases. In addition, we demonstrate the capability of the proposed method in simulating the flow with

multiple particles and their interaction with a wall-mounted cylinder. At last, the proposed framework is used to simulate particle-laden turbulent flows over periodic hills. The simulation results show that the recirculation bubble, the time-averaged streamwise velocity and the turbulent kinetic energy are affected by neutrally buoyant particles with a volume fraction of 0.5%.

The present work was focused on relatively simple cases to demonstrate the capability of the proposed framework. It is still quite challenging to apply particle-resolved methods to real-life applications (e.g., bed formation under the influence of arrays of hydrokinetic turbines), where the scale of particles is orders of magnitude smaller than the scale of the environment. Cases representing the key mechanism in real-life applications will be explored in the future work.

Acknowledgments

The authors were supported by NSFC Basic Science Center Program for "Multiscale Problems in Nonlinear Mechanics" (Grant No. 11988102), National Natural Science Foundation of China (Grant No. 12172360), Institute of Mechanics CAS, and Chinese Academy of Sciences. J. Qin also acknowledges the support from China Postdoctoral Science Foundation (Grant No. 2021M693241).

A CURVIB simulation of turbulent flows over periodic hills

Turbulent flow over periodic hills is a classical benchmark problem for testing the flow solver for simulating complicated turbulent flows with separation and reattachment [57,58]. In this appendix, we compare the predictions from the CURVIB simulation with those in literature for the case without particles. Fig. 16 shows the time-averaged streamlines and the contours of the time-averaged streamwise velocity simulated using the CURVIB method. Fig. 17 presents the comparison between the present results and the results of Breuer et al. [57]. For all the flow statistics considered, the present predictions are in good agreement with those of Breuer et al.

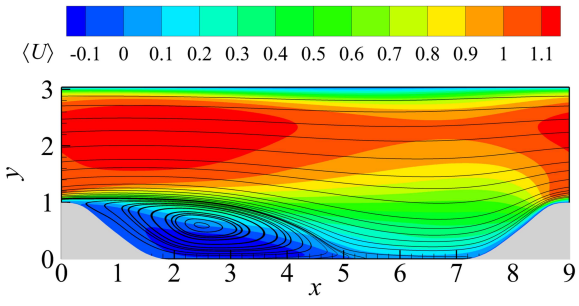


Figure 16: Time-averaged streamlines for the flow over periodic hills without particles.

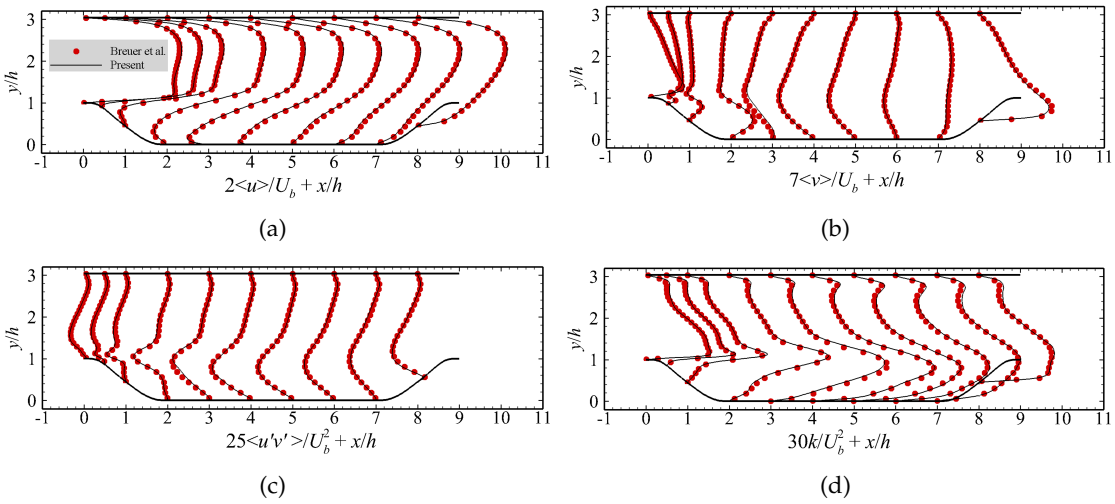


Figure 17: Comparison of vertical profiles for flow over periodic hills from Breuer et al. [57] and present CURVIB simulations for (a) Time-averaged streamwise velocity $\langle u\rangle$; (b) time-averaged vertical velocity $\langle v\rangle$; (c) primary Reynolds shear stress $\langle u'v'\rangle$; (d) turbulent kinetic energy $k=0.5\langle u'u'+v'v'+w'w'\rangle$.

References

[1] Rajat Mittal, Rui Ni, and Jung-Hee Seo. The flow physics of COVID-19. *Journal of Fluid Mechanics*, 894, 2020.

[2] Zhaobin Li, Xinlei Zhang, Ting Wu, Lixing Zhu, Jianhua Qin, and Xiaolei Yang. Effects of slope and speed of escalator on the dispersion of cough-generated droplets from a passenger. *Physics of Fluids*, 33(4):041701, 2021.

[3] Jianhua Qin, Guodan Dong, and Hui Zhang. Control of a sedimenting elliptical particle by electromagnetic forces. *Physics of Fluids*, 33(3):033305, 2021.

[4] Z Cui, A Dubey, L Zhao, and B Mehlig. Alignment statistics of rods with the Lagrangian stretching direction in a channel flow. *Journal of Fluid Mechanics*, 901, 2020.

[5] Xiaolei Yang, Ali Khosronejad, and Fotis Sotiropoulos. Large-eddy simulation of a hydrokinetic turbine mounted on an erodible bed. *Renewable Energy*, 113:1419–1433, 2017.

[6] Xiaolei Yang and Fotis Sotiropoulos. On the dispersion of contaminants released far upwind of a cubical building for different turbulent inflows. *Building and Environment*, 154:324–335, 2019.

[7] S. Balachandar and John K. Eaton. Turbulent dispersed multiphase flow. *Annual Review of Fluid Mechanics*, 42(1):111–133, 2010.

[8] Lixing Zhu and Arif Masud. Residual-based closure model for density-stratified incompressible turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 386:113931, 2021.

[9] Sudheer Tenneti and Shankar Subramaniam. Particle-resolved direct numerical simulation for gas-solid flow model development. *Annual Review of Fluid Mechanics*, 46:199–230, 2014.

[10] Charles S Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.

[11] Fotis Sotiropoulos and Xiaolei Yang. Immersed boundary methods for simulating fluid-structure interaction. *Progress in Aerospace Sciences*, 65:1–21, 2014.

[12] Boyce E Griffith and Neelesh A Patankar. Immersed methods for fluid-structure interaction. *Annual Review of Fluid Mechanics*, 52:421–448, 2020.
[13] Wei-Xi Huang and Fang-Bao Tian. Recent trends and progress in the immersed boundary method. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 233:7617–7636, 2019.
[14] Markus Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2):448–476, 2005.
[15] Xiaolei Yang, Xing Zhang, Zhilin Li, and Guo-Wei He. A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *Journal of Computational Physics*, 228(20):7821–7836, 2009.
[16] Shizhao Wang and Xing Zhang. An immersed boundary method based on discrete stream function formulation for two-and three-dimensional incompressible flows. *Journal of Computational Physics*, 230(9):3479–3499, 2011.
[17] Jianhua Qin, Yiannis Andreopoulos, Xiaohai Jiang, Guodan Dong, and Zhihua Chen. Efficient coupling of direct forcing immersed boundary-lattice Boltzmann method and finite element method to simulate fluid-structure interactions. *International Journal for Numerical Methods in Fluids*, 92(6):545–572, 2020.
[18] Amneet Pal Singh Bhalla, Rahul Bale, Boyce E Griffith, and Neelesh A Patankar. A unified mathematical framework and an adaptive numerical method for fluid-structure interaction with rigid, deforming, and elastic bodies. *Journal of Computational Physics*, 250:446–476, 2013.
[19] Lucy Zhang, Axel Gerstenberger, Xiaodong Wang, and Wing Kam Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193(21-22):2051–2067, 2004.
[20] Marcos Vanella and Elias Balaras. A moving-least-squares reconstruction for embedded-boundary formulations. *Journal of Computational Physics*, 228(18):6617–6628, 2009.
[21] Shizhao Wang, Marcos Vanella, and Elias Balaras. A hydrodynamic stress model for simulating turbulence/particle interactions with immersed boundary methods. *Journal of Computational Physics*, 382:240–263, 2019.
[22] Liang Ge and Fotis Sotiropoulos. A numerical method for solving the 3D unsteady incompressible Navier–Stokes equations in curvilinear domains with complex immersed boundaries. *Journal of Computational Physics*, 225(2):1782–1809, 2007.
[23] Long Lee and Randall J LeVeque. An immersed interface method for incompressible Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 25(3):832–856, 2003.
[24] Jianhua Qin, Ebrahim M Kolahdouz, and Boyce E Griffith. An immersed interface-lattice Boltzmann method for fluid-structure interaction. *Journal of Computational Physics*, 428:109807, 2021.
[25] Ebrahim M Kolahdouz, Amneet Pal Singh Bhalla, Brent A Craven, and Boyce E Griffith. An immersed interface method for discrete surfaces. *Journal of Computational Physics*, 400:108854, 2020.
[26] Ebrahim M Kolahdouz, Amneet PS Bhalla, Lawrence N Scotten, Brent A Craven, and Boyce E Griffith. A sharp interface Lagrangian-Eulerian method for rigid-body fluid-structure interaction. *Journal of Computational Physics*, 443:110442, 2021.
[27] Roland Glowinski, T-W Pan, Todd I Hesla, and Daniel D Joseph. A distributed Lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow*, 25(5):755–794, 1999.
[28] Lennart Schneiders, Claudia Günther, Matthias Meinke, and Wolfgang Schröder. An efficient conservative cut-cell method for rigid bodies interacting with viscous compressible

flows. *Journal of Computational Physics*, 311:62–86, 2016.
[29] Alexej Pogorelov, Lennart Schneiders, Matthias Meinke, and Wolfgang Schröder. An adaptive Cartesian mesh based method to simulate turbulent flows of multiple rotating surfaces. *Flow, Turbulence and Combustion*, 100(1):19–38, 2018.
[30] J Yang, S Preidikman, and E Balaras. A strongly coupled, embedded-boundary method for fluid-structure interactions of elastically mounted rigid bodies. *Journal of Fluids and Structures*, 24(2):167–182, 2008.
[31] Marcos Vanella, Patrick Rabenold, and Elias Balaras. A direct-forcing embedded-boundary method with adaptive mesh refinement for fluid-structure interaction problems. *Journal of Computational Physics*, 229(18):6427–6449, 2010.
[32] Shen-Wei Su, Ming-Chih Lai, and Chao-An Lin. An immersed boundary technique for simulating complex flows with rigid boundary. *Computers & Fluids*, 36(2):313–324, 2007.
[33] Shizhao Wang and Xing Zhang. An immersed boundary method based on discrete stream function formulation for two- and three-dimensional incompressible flows. *Journal of Computational Physics*, 230(9):3479 – 3499, 2011.
[34] Zeli Wang, Jianren Fan, and Kun Luo. Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles. *International Journal of Multiphase Flow*, 34(3):283–302, 2008.
[35] Wim-paul Breugem. A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows. *Journal of Computational Physics*, 231(13):4469–4498, 2012.
[36] Haoxiang Luo, Hu Dai, Paulo JSA Ferreira de Sousa, and Bo Yin. On the numerical oscillation of the direct-forcing immersed-boundary method for moving boundaries. *Computers & Fluids*, 56:61–76, 2012.
[37] Jung Hee Seo and Rajat Mittal. A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations. *Journal of Computational Physics*, 230(19):7347–7363, 2011.
[38] Thomas G. Fai and Chris H. Rycroft. Lubricated immersed boundary method in two dimensions. *Journal of Computational Physics*, 356:319–339, 2018.
[39] Edward Biegert, Bernhard Vowinckel, and Eckart Meiburg. A collision model for grain-resolving simulations of flows over dense, mobile, polydisperse granular sediment beds. *Journal of Computational Physics*, 340:105–127, 2017.
[40] Shilong Li, Xiaolei Yang, Guodong Jin, and Guowei He. Wall-resolved large-eddy simulation of turbulent channel flows with rough walls. *Theoretical and Applied Mechanics Letters*, 100228, 2021.
[41] Xiaolei Yang, Fotis Sotiropoulos, Robert J Conzemius, John N Wachtler, and Mike B Strong. Large-eddy simulation of turbulent flow past wind turbines/farms: The virtual wind simulator (VWiS). *Wind Energy*, 18(12):2025–2045, 2015.
[42] Guodan Dong, Zhaobin Li, Jianhua Qin, and Xiaolei Yang. How far the wake of a wind farm can persist for? *Theoretical and Applied Mechanics Letters*, 100314, 2021.
[43] Zhaobin Li and Xiaolei Yang. Large-eddy simulation on the similarity between wakes of wind turbines with different yaw angles. *Journal of Fluid Mechanics*, 921:A11, 2021.
[44] Iman Borazjani, Liang Ge, and Fotis Sotiropoulos. Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies. *Journal of Computational Physics*, 227(16):7587–7620, 2008.
[45] Zhi-Gang Feng and Efstathios E. Michaelides. Robust treatment of no-slip boundary condition and velocity updating for the lattice-Boltzmann simulation of particulate flows. *Computers & Fluids*, 38(2):370–381, 2009.

[46] Uhlmann Markus. Simulation of particulate flows on multi-processor machines with distributed memory. *CIEMAT Technical Report No. 1039, Madrid, Spain, ISSN 1135-9420*, 2003.
[47] Jan Kozicki and Frederic V Donze. YADE-OPEN DEM: An open-source software using a discrete element method to simulate granular material. *Engineering Computations*, 26(7):786–805, 2009.
[48] P.A. Cundall and O.D.L. Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 1:47–65, 1979.
[49] TA Johnson and VC Patel. Flow past a sphere up to a Reynolds number of 300. *Journal of Fluid Mechanics*, 378:19–70, 1999.
[50] E.A. Fadlun, R. Verzicco, and P. Orlandi. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161:35–60, 2000.
[51] Jungwoo Kim, Dongjoo Kim, and Haecheon Choi. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics*, 171(1):132–150, 2001.
[52] Boyce E. Griffith and Charles S. Peskin. On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems. *Journal of Computational Physics*, 208(1):75–105, 2005.
[53] A Ten Cate, CH Nieuwstad, JJ Derksen, and HEA Van den Akker. Particle imaging velocimetry experiments and lattice-Boltzmann simulations on a single sphere settling under gravity. *Physics of Fluids*, 14(11):4012–4025, 2002.
[54] P Gondret, M Lance, and L Petit. Bouncing motion of spherical particles in fluids. *Physics of Fluids*, 14(2):643–652, 2002.
[55] Jie Shen, Zhiming Lu, Lian-Ping Wang, and Cheng Peng. Influence of particle-fluid density ratio on the dynamics of finite-size particles in homogeneous isotropic turbulent flows. *Physical Review E*, 104(2):025109, 2021.
[56] Chenlin Zhu, Zhaosheng Yu, Dingyi Pan, and Xueming Shao. Interface-resolved direct numerical simulations of the interactions between spheroidal particles and upward vertical turbulent channel flows. *Journal of Fluid Mechanics*, 891:A6, 2020.
[57] Michael Breuer, Nikolaus Peller, Ch Rapp, and Michael Manhart. Flow over periodic hills–numerical and experimental study in a wide range of Reynolds numbers. *Computers & Fluids*, 38(2):433–457, 2009.
[58] Zhideng Zhou, Ting Wu, and Xiaolei Yang. Reynolds number effect on statistics of turbulent flows over periodic hills. *Physics of Fluids*, 33(10):105124, 2021.