Force-amplified, single-sided diffused-interface immersed boundary kernel for correct local velocity gradient computation and accurate no-slip boundary enforcement

Cheng Peng [*]

Department of Energy and Mineral Engineering and EMS Energy Institute, The Pennsylvania State University, University Park, Pennsylvania 16802, USA

Lian-Ping Wang [†]

Guangdong Provincial Key Laboratory of Turbulence Research and Applications, Center for Complex Flows and Soft Matter Research and Department of Mechanics and Aerospace Engineering, Southern University of Science and Technology, Shenzhen 518055, Guangdong, China and 126 Spencer Lab, Department of Mechanical Engineering, University of Delaware, Newark, Delaware, 19711, USA

The current diffused-interface immersed boundary method (IBM) with a two-sided force distribution kernel cannot be used to correctly calculate the velocity gradients within the diffused solid-fluid interfaces. This is because the nonzero boundary force distributed to the fluid nodes modifies the momentum equation solved at these locations from the Navier-Stokes equations (NSEs). In this paper, this problem is analytically identified in simple plane channel flow. A single-sided force distribution kernel is used to restrict the boundary force in the solid region and restore NSEs in the fluid region for correct velocity gradient computation. In order to improve the no-slip boundary enforcement in IBM, an extremely simple force amplification technique is proposed. This technique requires no additional computation cost and can significantly reduce the necessary iterations to achieve accurate no-slip boundary enforcement. The single-sided kernel and the force amplification technique are examined in both laminar and turbulent flows. Compared to the standard IBM, the proposed methods not only produce correct velocity gradient results near a solid surface but also reduce numerical errors in the flow velocity and hydrodynamic force and torque results.

I. INTRODUCTION

The immersed boundary method (IBM) is a powerful technique of treating boundary conditions on complex geometries in many applications of the modern computational fluid dynamics (CFD) [1]. These boundary conditions can be Dirichlet boundaries [2–4], Neumann boundaries [5], and combined types such as the slip boundaries for inviscid compressible flows [6,7]. The essential idea of IBM is to convert the physical boundary conditions on a solid surface to the Dirichlet constraints that should be enforced on the adjacent Cartesian grid points so body-fit meshes are not necessary for boundary treatment on the complex geometries. There are mainly two categories of IBM, sometimes named the sharp-interface (SI) IBM and the diffused-interface (DI) IBM. SI-IBM often enforces certain constraints on a layer of grid points inside the solid phase but close to the boundary, so that locations on the solid surface satisfy the desired boundary conditions. The biggest advantage of this category is that a "sharp" representation of the interface is obtained and the boundary treatment can be designed with a high accuracy [5]. DI-IBM, however, diffuses the interface as a thin but finite-thickness layer around the solid surface, which brings it relative easiness to implement and a smoothing technique

to suppress the numerical noise in simulations. Due to these advantages, this category of IBM is more popular than SI-IBM in applications involving a large amount of moving objects, such as particle-laden flows [3,8–10].

DI-IBM has many advantages, such as easiness in tracking the fluid-solid interfaces, reasonable computational costs, simplicity in implementation, and robustness in handling the hydrodynamic interactions, but it also has some disadvantages. One well-known disadvantage is its first-order accuracy for flow velocity. This disadvantage comes from the use of δ function, which is usually employed by DI-IBM to interpolate velocity field from the Eulerian grid for fluid flow simulation to the Lagrangian grid attached to the immersed objects, and the boundary force the other way around [2,11]. It is worth mentioning that there are also studies replacing δ functions with moving-least-square (MLS) reconstructions for information exchange between the Lagrangian and Eulerian grids and claiming second-order accuracy [12–14]. These alternatives are less seen in the applications of particle-laden flows, whose potential in the field may worth a further exploration. In practice, the first-order accuracy of DI-IBM does not seem to affect the quality of the simulation results, as validated in many flows, both laminar and turbulent [3,9,15–17]. Another disadvantage of DI-IBM is the lack of correctness in computing the local velocity gradients inside the diffused boundary [16–19]. This is because the nonzero boundary force applied to the fluid nodes modifies the momentum equation solved at these locations from the physical one, which is

---
[*]czp341@psu.edu
[†]wanglp@sustech.edu.cn

the Navier-Stokes equations (NSEs). Although this problem remains local in a thin layer around the solid surface, it still challenges the reliability of DI-IBM in certain applications. For example, DI-IBM has been widely used to study the flow modulation by finite-size solid particles in a variety of turbulent flows [20–24], where the enhanced dissipation rate of the turbulent kinetic energy around the particle surfaces has been identified as one of the most important modulation mechanisms [25]. The capability of correctly predicting the velocity gradient near the particle surfaces plays a crucial role to quantify turbulent modulation, especially the local modulation in the vicinity of particle surfaces. A third disadvantage of DI-IBM is it often requires iterations of boundary force to achieve an accurate no-slip boundary enforcement on solid surfaces [9,10,26]. In numerical simulations, iterations should be avoided or reduced whenever it is possible to save computational costs.

In the past, the issue of incorrect velocity gradient computation in DI-IBM has not been paid enough attention. Only a few attempts were made to restore correct velocity gradient results within the diffused layer of the solid boundary [18,19,27]. These attempts proposed to first calculate the velocity gradients at some ghost points in the fluid phase that are outside the diffused layer but still close the solid boundary, then use those results to approximate or extrapolate the velocity gradient inside the diffused layer. As will be analyzed in detail in Sec. II C 1, those attempts would only work well in cases where the velocity gradient is a constant or linearly distributed near solid boundaries. In more general cases such as particle-laden flows, where the velocity gradient and dissipation rate change dramatically near the particle surfaces, those attempts could easily fail.

In this study, we propose to use a single-sided force distribution kernel to replace the standard two-sided force distribution kernel in DI-IBM to achieve correct velocity gradient computation near solid surfaces. The essential idea is to restrict the distribution of boundary force only to the solid phase so the physical momentum equation is restored in the whole fluid phase, including the regions attached to the solid boundaries. Furthermore, in order to improve the accuracy of no-slip boundary enforcement with DI-IBM, a simple force amplification technique is introduced that introduces no additional computational cost but can significantly reduce the necessary number of boundary force iterations.

The remaining parts of the paper will be organized as follows. In Sec. II we will first develop a mathematical analysis using a simple plane channel flow to quantify the level of inaccuracy of velocity gradient computation inside the diffused layer of a solid boundary with the standard DI-IBM, and show that the error can be rather significant. The failure of those attempts made in the literature to solve this problem will also be discussed in detail. With the understanding gained from Sec. II and an analysis on the issue of inaccurate no-slip boundary enforcement in DI-IBM, a single-sided force distribution kernel to restore correct velocity gradient computation and a force amplification technique to improve the accuracy of no-slip boundary enforcement will be presented in Sec. III. In Sec. IV those proposed methods will be validated in both laminar and turbulent flows, and systematically compared against DI-IBM with the two-sided force distribution kernel. Finally,

a conclusion and remarks of this study will be recapitulated in Sec. V.

II. ISSUE OF INCORRECT VELOCITY GRADIENT COMPUTATION WITHIN THE DIFFUSED LAYER AND CURRENT SOLUTIONS

A. A standard DI-IBM with two-sided force distribution kernel

One of the most commonly used DI-IBM was refined by Uhlmann [3]. The same as the other DI-IBM algorithms, it requires two sets of grids, a fixed Eulerian grid on which the NSEs are solved, and a Lagrangian grid attached to every immersed solid body on which the no-slip boundary is enforced. Regardless of the flow solver used, this standard DI-IBM contains three key steps:

Step 1: the unforced preliminary velocity field on the Eulerian grid is used to interpolate the flow velocities at the Lagrangian grid via a regularized δ function

$$\vec{U}(\vec{X}_L) = \sum_{\vec{x}_E} \vec{u}(\vec{x}_E)\delta_h(\vec{x}_E - \vec{X}_L)h^3, \tag{1}$$

where $\vec{U}(\vec{X}_L)$ and $\vec{u}(\vec{x}_E)$ are the flow velocity at the Lagrangian grid $\vec{X}_L$ and Eulerian grid $\vec{x}_E$, respectively. By default, the uppercase letters indicate the properties on the Lagrangian grid points, and the lowercase letters indicate the properties on the Eulerian grid points. h^3 is the cell volume of an Eulerian grid cell in three dimensions, h is the grid spacing. δ_h is the regularized δ function that determines the weighting factors in the interpolation and ensures the interpolation is confined locally. One of the most commonly used regularized δ functions was given by Peskin [2]

$$\delta_h = \frac{1}{h^3}\phi\left(\frac{x_1}{h}\right)\phi\left(\frac{x_2}{h}\right)\phi\left(\frac{x_3}{h}\right),$$

$$\phi(r) = \begin{cases} 0, & |r| \geqslant 2, \\ \frac{1}{8}\left(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}\right), & 1 \leqslant |r| < 2, \\ \frac{1}{8}\left(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}\right), & 0 \leqslant |r| < 1, \end{cases}$$
$$\tag{2}$$

This regularized δ function conserves the zeroth- and first-order moment of any continuous function during the interpolation between the Lagrangian grid and the Eulerian grid.

Step 2: Once the interpolation in Step 1 is done, i.e., the unforced velocity at the Lagrangian grid is determined, a boundary force $\vec{F}$ that forces the fluid velocity at the Lagrangian grid to obey the no-slip condition is calculated as

$$\vec{F}(\vec{X}_L) = \frac{\vec{U}^d(\vec{X}_L) - \vec{U}(\vec{X}_L)}{\delta t}. \tag{3}$$

Here $\vec{U}^d$ is the target velocity at the Lagrangian grid point that satisfies the no-slip condition, δt is the time step size.

Step 3: At last, the boundary force at the Lagrangian grid is interpolated back to the Eulerian grid

$$\vec{f}(\vec{x}_E) = \sum_{\vec{X}_L} \vec{F}(\vec{X}_L)\delta_h(\vec{x}_E - \vec{X}_L)\Delta V_L, \tag{4}$$
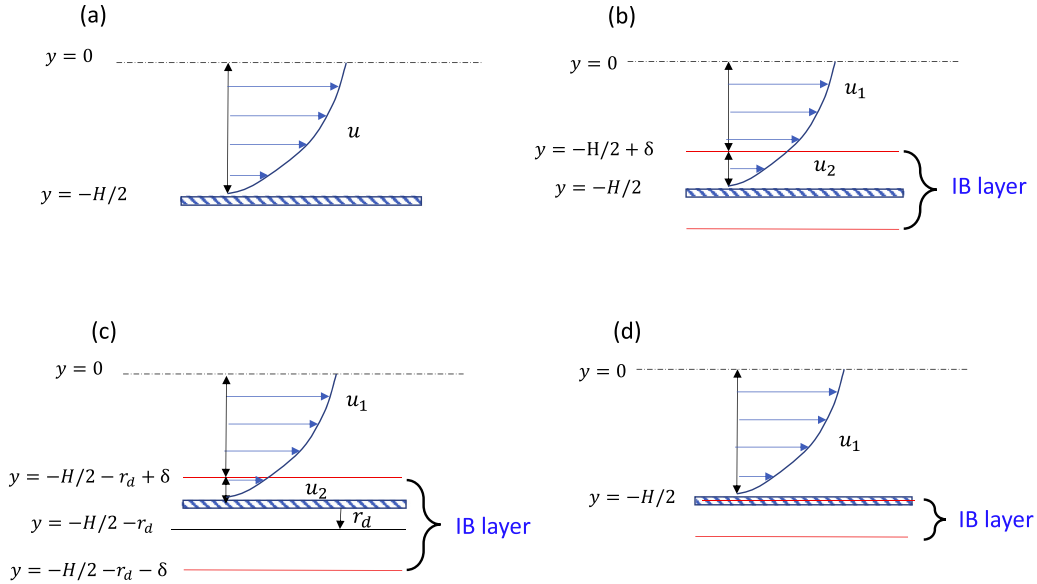
FIG. 1. The physical plane channel flow (a) and its numerical realizations with different IBM algorithms: (b) standard IBM [3], (c) IBM with Lagrangian grid retraction [9], (d) IBM with single-sided force distribution strategy.

ΔV_L is the cell volume represented by each Lagrangian grid point, which is usually chosen to be $\Delta V_L \approx h^3$ [3].

The above three key steps of this DI-IBM can be easily incorporated with any flow solver of NSEs. The implementation of this algorithm may also vary. To make the discussion general, our analysis below is not associated to any specific flow solver and implementation strategy.

B. Analytic demonstration of the incorrect velocity gradient computation in a plane channel flow

The essential reason that the above DI-IBM cannot correctly predict the velocity gradients within the diffused layer of the solid boundary is that the physical momentum equation solved at the Eulerian grid has been modified locally by the present of a nonzero boundary force. To illustrate, let us consider a steady-state plane channel flow driven by a constant body force [Fig. 1(a)], whose physical momentum equation is simplified from the incompressible N-S equation as

$$\mu \frac{d^2 u}{dy^2} + \rho g = 0, \quad \text{for} \quad -\frac{H}{2} \leqslant y \leqslant 0, \tag{5}$$

where u is the flow velocity in the streamwise direction, y is the spatial coordinate normal to the channel wall, μ is the fluid dynamic viscosity, ρ is the fluid density, and g is the driving force per unit fluid mass. The analytic solutions of velocity gradient and velocity in this plane channel flow at the steady state can be easily obtained as

$$\frac{du}{dy} = -\frac{\rho g y}{\mu}, \quad u = -\frac{\rho g}{2\mu}\left[y^2 - \left(\frac{H}{2}\right)^2\right],$$

$$\text{for} -\frac{H}{2} \leqslant y \leqslant 0 \tag{6}$$

On the other hand, the actual momentum equation solved by the DI-IBM [Fig. 1(b)] is

$$\mu \frac{d^2 u_1}{dy^2} + \rho g = 0, \quad -\frac{H}{2} + \delta \leqslant y \leqslant 0, \tag{7a}$$

$$\mu \frac{d^2 u_2}{dy^2} + \rho g - \frac{\rho g H}{4\delta}\left[1 + \cos\frac{\pi(y + H/2)}{\delta}\right] = 0,$$

$$-\frac{H}{2} \leqslant y \leqslant -\frac{H}{2} + \delta, \tag{7b}$$

where δ is the half span of the regularized δ function. Note that in Eqs. (7) we use u_1 and u_2 to distinguish the velocity solved by DI-IBM and its physical counterpart u. The last term in Eq. (7b) is the boundary force distributed by a cosine shape regularized δ function

$$\phi(r) = \frac{1}{2\delta}\left[1 + \cos\left(\frac{\pi r}{\delta}\right)\right]. \tag{8}$$

We adopt this δ function mainly for mathematical simplicity, which would not affect the general observation made below.

Equations (7) can be also solved analytically. Assuming $y = 0$ is still a symmetric plane, i.e., $du_1/dy = 0$ at $y = 0$, the no-slip condition is still satisfied at the channel wall, i.e., $u_2 = 0$ at $y = -H/2$, and the velocity is continuously differentiable everywhere in the fluid region, i.e., $du_1/dy = du_2/dy$ and $u_1 = u_2$ at $y = -H/2 + \delta$, the solutions of velocity gradient and velocity of the IBM are

$$\frac{du_1}{dy} = \underbrace{-\frac{\rho g y}{\mu}}_{\text{exact}}, \tag{9a}$$

$$\frac{du_2}{dy} = \underbrace{-\frac{\rho g y}{\mu}}_{\text{exact}} + \frac{\rho g H}{4\mu} \underbrace{\left\{ \frac{1}{\delta}\left(y + \frac{H}{2}\right) - 1 + \frac{1}{\pi}\sin\left[\frac{\pi}{\delta}\left(y + \frac{H}{2}\right)\right] \right\}}_{O(1)}, \tag{9b}$$

$$u_1 = \underbrace{-\frac{\rho g}{2\mu}\left[y^2 - \left(\frac{H}{2}\right)^2\right]}_{\text{exact}} - \frac{\rho g H^2}{2\mu} \underbrace{\left(\frac{1}{4} - \frac{1}{\pi^2}\right)\frac{\delta}{H}}_{O(\delta/H)}, \tag{10a}$$

$$u_2 = \underbrace{-\frac{\rho g}{2\mu}\left[y^2 - \left(\frac{H}{2}\right)^2\right]}_{\text{exact}} + \frac{\rho g H^2}{8\mu} \underbrace{\left(\left(y + \frac{H}{2}\right)\left(y + \frac{H}{2} - 2\delta\right)\frac{1}{\delta H} + \frac{2\delta}{\pi^2 H}\left\{1 - \cos\left[\frac{\pi}{\delta}\left(y + \frac{H}{2}\right)\right]\right\}\right)}_{O(\delta/H)}. \tag{10b}$$

Within the diffused layer of the channel wall $-H/2 \leqslant y \leqslant -H/2 + \delta$, the full term inside { } of Eq. (9b) varies from -1 to 0, thus the error term of the velocity gradient computation within the diffused layer is of $O(1)$. At the wall location $y = -H/2$, this relative error reaches its maximum

$$\left\| \frac{\frac{du_2}{dy}\big|_{y=-H/2} - \frac{du}{dy}\big|_{y=-H/2}}{\frac{du}{dy}\big|_{y=-H/2}} \right\| = 50\%, \tag{11}$$

which is significant.

Equations (10) indicate that DI-IBM also creates a relative error in the velocity field. This error grows from zero and finally reaches a constant outside the diffused layer. The magnitude of the error is on the order of $O(\delta/H)$. This also confirms that DI-IBM is first-order accuracy only in terms of flow velocity field, which is a well-known property of diffused interface IBM [2].

C. A review of current solutions for the problem of velocity gradient computation

The identified problem of incorrect velocity gradient computation in DI-IBM has not been widely recognized by the IBM community. In the literature, only a few attempts were made to restore correct velocity gradient within the diffused layer of solid boundaries. These attempts are reviewed in this section.

1. Extrapolation approaches

To correctly compute the velocity gradient at **X** locating on the solid surface, Williams *et al.* [18] proposed to first compute velocity gradients at an alternative location in the fluid phase $\mathbf{X} - d\mathbf{n}$, where **n** is the normal vector pointing into the fluid phase and d is made large enough so the location is just outside the diffused layer of the solid boundary; then use the computed results at $\mathbf{X} - d\mathbf{n}$ to approximate the velocity gradients at **X**. This may be viewed as a first-order approximation with $\delta \sim h$. A similar "ghost cell method" was introduced by Santarelli *et al.* [19], where the velocity gradients were computed at two locations $\mathbf{X} - d\mathbf{n}$, and $\mathbf{X} - 2d\mathbf{n}$, both inside the fluid phase, then linearly extrapolated to **X** on the solid surface, as such yielding a second-order accuracy. In order to ensure that the velocity gradients can be accurately computed at $\mathbf{X} - d\mathbf{n}$, and $\mathbf{X} - 2d\mathbf{n}$, d needs to be large enough so

the Eulerian grid points within the diffused layer could be excluded in the computation. Huang and Tian concluded that $d > 2h$ is typically required [27].

Unfortunately, the above attempts function well only in a few specific flows, where the velocity gradient is either constant or linearly distributed near a solid boundary, such as plane Couette flow and plane channel flow. In more general flows, such as particle-laden flows, where the velocity gradient changes dramatically near the solid boundary, those attempts can easily fail. To illustrate this point, let us consider one of the simplest particle-laden flow cases, a fixed sphere confronting a uniform stream in the Stokes limit, where the analytic solution of this flow is widely known as

$$u_r = V\left(1 - \frac{3a}{2r} + \frac{a^3}{2r^3}\right)\cos\theta,$$

$$u_\theta = -V\left(1 - \frac{3a}{4r} + \frac{a^3}{4r^3}\right)\sin\theta, \quad u_\varphi = 0, \tag{12}$$

where u_r, u_θ, and u_φ are the velocity components in the radial, polar, and azimuthal directions in a spherical coordinate, respectively. V is the velocity of the incoming uniform stream, θ is the polar angle, r is the radial location, and $r = a$ represents the location of the particle surface.

With this analytic velocity field, the nonzero velocity gradient components are

$$\frac{\partial u_r}{\partial r} = V\left(\frac{3a}{2r^2} - \frac{3a^3}{2r^4}\right)\cos\theta,$$

$$\frac{1}{r}\frac{\partial u_r}{\partial \theta} = -V\left(\frac{1}{r} - \frac{3a}{2r^2} + \frac{a^3}{2r^4}\right)\sin\theta,$$

$$\frac{\partial u_\theta}{\partial r} = -V\left(\frac{3a}{4r^2} - \frac{3a^3}{4r^4}\right)\sin\theta,$$

$$\frac{1}{r}\frac{\partial u_\theta}{\partial \theta} = -V\left(\frac{1}{r} - \frac{3a}{4r^2} + \frac{a^3}{4r^4}\right)\cos\theta. \tag{13}$$

TABLE I. Results of velocity gradients and dissipation rate on the surface of a fixed particle confronting a uniform flow in the Stokes limit computed from the approach of Williams *et al.* and that of Santarelli *et al.*

	$\left[\frac{a}{V\cos\theta}\frac{\partial u_r}{\partial r}\right]^W$	$\left[\frac{-a}{V\sin\theta}\frac{1}{r}\frac{\partial u_r}{\partial\theta}\right]^W$	$\left[\frac{-a}{V\sin\theta}\frac{\partial u_\theta}{\partial r}\right]^W$	$\left[\frac{-a}{V\cos\theta}\frac{1}{r}\frac{\partial u_r}{\partial\theta}\right]^W$	$\left[\frac{2a^2}{3\nu V^2}\langle\varepsilon\rangle^{\theta,\varphi}\right]^W$
$a=10h$	0.3183	0.0328	0.1591	0.4331 (13.38%)	0.3001 (69.99%)
$a=20h$	0.2151	0.0109	0.1076	0.4600 (8.00%)	0.4974 (50.26%)
$a=40h$	0.1265	0.0032	0.0632	0.4778 (4.44%)	0.6875 (31.25%)
Theory	0	0	0	0.5000	1.0000
	$\left[\frac{a}{V\cos\theta}\frac{\partial u_r}{\partial r}\right]^S$	$\left[\frac{-a}{V\sin\theta}\frac{1}{r}\frac{\partial u_r}{\partial\theta}\right]^S$	$\left[\frac{-a}{V\sin\theta}\frac{\partial u_\theta}{\partial r}\right]^S$	$\left[\frac{-a}{V\cos\theta}\frac{1}{r}\frac{\partial u_r}{\partial\theta}\right]^S$	$\left[\frac{2a^2}{3\nu V^2}\langle\varepsilon\rangle^{\theta,\varphi}\right]^S$
$a=10h$	0.2618	−0.0135	0.1309	0.4694 (6.12%)	0.4388 (56.12%)
$a=20h$	0.1119	−0.0109	0.0560	0.4870 (2.60%)	0.6946 (30.54%)
$a=40h$	0.0379	−0.0046	0.0189	0.4956 (0.88%)	0.8776 (12.24%)
Theory	0	0	0	0.5000	1.0000

The dissipation rate of the kinetic energy averaged over θ and φ directions also can be obtained:

$$
\begin{aligned}
\langle\varepsilon\rangle^{\theta,\varphi} &= \frac{1}{4\pi r^2}\int_0^\pi (2\nu s_{ij}s_{ij})2\pi r^2\sin\theta\,d\theta \\
&= \frac{3}{2}\frac{\nu V^2}{r^2}\left(\frac{3a^2}{2r^2}-\frac{3a^4}{r^4}+\frac{5a^6}{2r^6}\right).
\end{aligned} \tag{14}
$$

In Table I we show the results of velocity gradients and averaged dissipation rate computed with the approach of Williams *et al.* [18] (with a superscript W) and that of Santarelli *et al.* [19] (with a superscript S) at the particle surface $r=a$ with different grid resolutions a/h. Compared with the analytic solutions, even with a high grid resolution $a=40h$, the computed dissipation rates still contain relative errors over 10%. Further increasing the grid resolution should result in more accurate results, but is barely affordable, especially for studies involving large amounts of particles. In Table II we listed the grid resolutions adopted in some state-of-the-art turbulent particle-laden flow studies conducted with DI-IBM. The grid resolutions in those studies barely passed $a/h=20$. It should be emphasized that the relative errors listed in Table I come from a simple case of a uniform Stokes flow passing a sphere. In turbulent particle-laden flows where not only the flows become more complex but also the particle Reynolds numbers are usually higher, it is reasonable to expect these approaches would perform even worse. Using these approaches to calculate the local velocity gradients and

dissipation rate inside the diffused layers of a solid boundary is thus unreliable.

2. Approaches of retracting the Lagrangian grid

Retracting the Lagrangian grid from the actual solid surface to an alternative surface inside the solid domain is a common strategy to compensate the boundary diffusion in DI-IBM [9,32]. Its effects on improving the accuracy of fluid velocity and hydrodynamic force and torque predictions have been discussed thoroughly in literature, e.g., by Breugem in particle-laden flows [9]. Unfortunately, this retraction approach still cannot fully solve the problem of the velocity gradient computation inside the diffused layer. An analysis is provided here.

For the plane channel flow case discussed in Sec. II B, assuming the Lagrangian grid is retracted from the channel wall location $y=-H/2$ to $y=-H/2-r_d$ by a distance r_d, as shown in Fig. 1(c), the momentum equation actually solved by DI-IBM becomes

$$
\mu\frac{d^2u_1}{dy^2}+\rho g=0,\quad -\frac{H}{2}+\delta-r_d\leqslant y\leqslant 0, \tag{15a}
$$

$$
\mu\frac{d^2u_2}{dy^2}+\rho g-\frac{\rho gH}{4\delta}\left[1+\cos\frac{\pi(y+H/2+r_d)}{\delta}\right]=0,
$$
$$
-\frac{H}{2}\leqslant y\leqslant-\frac{H}{2}+\delta-r_d. \tag{15b}
$$

TABLE II. Grid resolutions in some recent particle-laden turbulent flow simulations with spherical particles using IBM. Abbreviations: HIT (homogeneous isotropic turbulence), FV (finite-volume), LBM (lattice Boltzmann method).

	Flow type	Numerical method	No. of particles	a/h
Uhlmann [21]	Vertical channel	FV-IBM	512–4096	6.4
Lucci *et al.* [22]	HIT	FV-IBM	640–6400	4.0–8.5
Cisse *et al.* [28]	HIT	FV-IBM	1	8.3–32.8
Picano *et al.* [23]	Horizontal channel	FV-IBM	2500–10 000	8.0
Rubinstein *et al.* [29]	Fluidized bed	LBM-IBM	⩽1600	4.0–9.0
Costa *et al.* [30]	Horizontal channel	FV-IBM	80 000–640 000	8.0
Ardekani *et al.* [31]	Horizontal channel	FV-IBM	5000	16.0
Eshghinejadfard *et al.* [24]	Horizontal channel	LBM-IBM	66–305	7.8–13
Brändle de Motta *et al.* [15]	HIT	FV-IBM	4450–35 602	6.0

The velocity and velocity gradient are once again solved analytically as

$$
u_1 = \underbrace{-\frac{\rho g}{2\mu}\left[y^2 - \left(\frac{H}{2}\right)^2\right]}_{\text{exact}} \underbrace{-\frac{\rho g H}{2\mu}\left(\frac{1}{4} - \frac{1}{\pi^2}\right)\delta}_{\text{original error term}} + \underbrace{\frac{\rho g H}{2\mu}\left(r_d + \frac{r_d^2}{H}\right)}_{\text{correction term}},
\tag{16a}
$$

$$
u_2 = \underbrace{-\frac{\rho g}{2\mu}\left[y^2 - \left(\frac{H}{2}\right)^2\right]}_{\text{exact}} + \frac{\rho g H}{2\mu}\left(r_d + \frac{r_d^2}{H}\right) + \frac{\rho g H}{8\mu\delta}\left\{\left(y + \frac{H}{2} + r_d\right)\left(y + \frac{H}{2} + r_d - 2\delta\right)\right.
$$

$$
\left. + \frac{2\delta^2}{\pi^2}\left[1 - \cos\frac{\pi(y + H/2 + r_d)}{\delta}\right]\right\},
\tag{16b}
$$

$$
\frac{du_1}{dy} = \underbrace{-\frac{\rho g y}{\mu}}_{\text{exact}},
\tag{17a}
$$

$$
\frac{du_2}{dy} = \underbrace{-\frac{\rho g y}{\mu}}_{\text{exact}} + \underbrace{\frac{\rho g H}{4\mu\delta}\left\{\left(y + \frac{H}{2}\right) + (r_d - \delta) + \frac{\delta}{\pi}\sin\left[\frac{\pi}{\delta}\left(y + \frac{H}{2} + r_d\right)\right]\right\}}_{O(1)}.
\tag{17b}
$$

Compared to the solutions of DI-IBM without retraction, i.e., Eq. (10), a positive correction term [the third term on the right-hand side of Eq. (16)] is generated by the retraction that can compensate the original error term and result in more accurate velocity field. The optimal retraction distance r_d can be determined by canceling the second and third terms in Eq. (16), which gives

$$
r_d|_{\text{optimal}} = \frac{\sqrt{H^2 + 4\left(\frac{1}{4} - \frac{1}{\pi^2}\right)\delta H} - H}{2} \approx \left(\frac{1}{4} - \frac{1}{\pi^2}\right)\delta.
\tag{18}
$$

Assuming $\delta = 2h$, the optimal r_d calculated from Eq. (18) with $H = 20h$, $H = 40$, $H = 80$, and $H = 160$ are $0.2931h$, $0.2952h$, $0.2963h$, and $0.2968h$, respectively. The optimal retraction distance is not sensitive to the grid resolution, and it is roughly $0.3h$. This retraction distance was also recommended by Breugem because it always yielded the most accurate fluid velocity and hydrodynamic interaction results in a variety of tested flows [9]. Our analysis provides a theoretical backing for the retraction distance recommended by Breugem.

Unfortunately, the optimal retraction distance is not able to fully restore the velocity gradient within the diffused layer. According to Eq. (17b), assuming $\delta = 2h$, the velocity gradient at wall location $y = -H/2$ would still be under predicted by 35.37% with the optimal retraction distance of $r_d = 0.3h$. On the other hand, a retraction distance $r_d = \delta$ does fully restore correct velocity gradient computation in the whole diffused layer, but such a large retraction distance could ruin the accuracy of velocity field and hydrodynamic interaction results. Due to such conflict, the retraction approach is not a good candidate to solve the problem of velocity gradient computation in DI-IBM.

III. A FORCE-AMPLIFIED, SINGLE-SIDED DI-IBM FOR CORRECT VELOCITY GRADIENT COMPUTATION AND ACCURATE NO-SLIP BOUNDARY ENFORCEMENT

A. A singled-sided force distribution kernel

In this section, we introduce a simple solution to the issue of incorrect velocity gradient computation within the diffused layer of solid boundaries. Following the analysis in Sec. II, it should be quite clear that the nonzero boundary force at the Eulerian grid points in the fluid phase is the origin of the incorrect velocity gradient computation inside the diffused layer. Therefore, if the boundary force is restricted only to the solid phase, a correct velocity gradient should be restored for the whole fluid region of interest. With this understanding, we propose to use a single-sided force distribution kernel to replace the two-sided kernel of the standard DI-IBM when distributing the boundary force from the Lagrangian grid to Eulerian grid. In the third step of DI-IBM, Eq. (4) is modified as

$$
\vec{f}(\vec{x}_E \in \text{solid}) = \sum_{\vec{X}_L} \frac{\vec{F}(\vec{X}_L)\delta_h(\vec{x}_E - \vec{X}_L)\Delta V_L}{W(\vec{X}_L)},
\tag{19}
$$

where $W(\vec{X}_L)$ is the weighting factor calculated in terms of each individual Lagrangian grid point $\vec{X}_L$ as

$$
W(\vec{X}_L) = \sum_{\vec{x}_E \in \text{solid}} \delta_h(\vec{x}_E - \vec{X}_L)h^3.
\tag{20}
$$

Obviously, this weighting factor is divided to ensure the integral boundary forces applied to the Eulerian and Lagrangian grids is conserved, i.e.,

$$
\sum_{\vec{X}_L} \vec{F}(\vec{X}_L) = \sum_{\vec{x}_E} \vec{f}(\vec{x}_E).
\tag{21}
$$

It should be noted that the same single-sided boundary force distribution kernel was introduced by Ji *et al.* [33] for the

purpose of reducing the boundary diffusion. Compared to the regular two-sided force distribution kernel, the single-sided kernel was reported to result in more accurate drag and lift coefficients and local velocity fields near the solid boundaries in cases of a uniform flow passing a fixed cylinder or ellipse. However, no effort was made by Ji *et al.* to investigate how this kernel could be used to restore correct velocity gradient close to the boundary. In addition, due to its one-sided boundary force implementation, the single-sided kernel does not conserves torque during the interpolation of boundary force from the Lagrangian grid to the Eulerian grid. How this lack of torque conservation affects its actual performances was also not investigated by Ji *et al.*, as their validation tests involved no particle rotations [33]. These aspects will be covered by the present study. Furthermore, Ji *et al.* also reported that the single-sided kernel required a large number of iterations to enforce accurate no-slip conditions on the solid surface [33]. We would further refine the IBM algorithm by introducing a simple force amplification technique to improve the no-slip boundary enforcement. This aspect will be discussed shortly in Sec. III B.

The differences between the single-sided force distribution kernel and the approaches of retracting the Lagrangian grid reviewed in Sec. II C 2 should be also emphasized. The essential idea of the single-sided kernel is to restore the physical momentum equations of the flow by removing the boundary force from the whole fluid region. On the other hand, retracting the Lagrangian grid is to offset the enlarged effective radius of the solid object due to boundary diffusion. In terms of the implementation, the single-sided kernel reduces the thickness of the diffused layer by a half, i.e., from 2δ to δ, as shown in Fig. 1(d). On the contrary, retracting the Lagrangian grid only shifts the location of the diffused layer by a distance r_d. The thickness of the diffused layer is unchanged, as shown in Fig. 1(c). Finally, as being analyzed in Sec. II C 2 and will be confirmed by numerical simulations in Sec. IV, retracting the Lagrangian grid is generally incapable of restoring the correct velocity gradient near the solid boundaries. The single-sided kernel, on the other hand, ensures correct velocity computation in all circumstances.

B. A simple force amplification technique for accurate no-slip boundary enforcement

It is well-known that a single execution of the DI-IBM kernel, i.e., Eq. (1) to Eq. (4) cannot enforce accurate no-slip condition at the Lagrangian grid [9,10,26]. One method to address this issue is to execute the DI-IBM kernel and updating the velocity field on the Eulerian grid multiple times while pausing the flow solver until the interpolated velocity at the Lagrangian grid converges to its targeted value with sufficiently small deviations [9,10,26]. In numerical simulations, any type of iteration would require additional computational costs and should be avoided if possible. In this section, an extremely simple force amplification technique is introduced that could significantly improve the accuracy of no-slip boundary enforcement with a negligible computational cost.

Before bringing up this technique, let us first understand where the error of no-slip boundary enforcement comes from and why the iterative-direct forcing could eventually eliminate
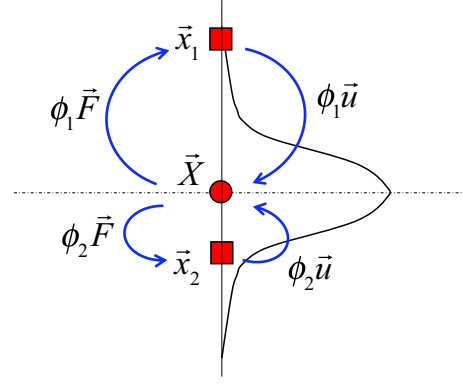

FIG. 2. An ideal alignment of Lagrangian and Eulerian grid points around a solid boundary.

this error. To help demonstration, let us consider an ideal boundary configuration in one dimension shown in Fig. 2 that consists of only one Lagrangian grid point and two Eulerian grid points, one on each side of the Lagrangian grid. For this ideal configuration, $h = \Delta V_L = 1$, the three key steps in the DI-IBM reduce to

$$\vec{U}(\vec{X}) = \phi_1 \vec{u}(\vec{x}_1) + \phi_2 \vec{u}(\vec{x}_2), \tag{22a}$$

$$\vec{F}(\vec{X}) = \frac{\vec{U}^d(\vec{X}) - \vec{U}(\vec{X})}{\delta t}, \tag{22b}$$

$$\vec{f}(\vec{x}_1) = \phi_1 \vec{F}(\vec{X}), \quad \vec{f}(\vec{x}_2) = \phi_2 \vec{F}(\vec{X}). \tag{22c}$$

With the boundary force $\vec{f}(\vec{x}_1)$ and $\vec{f}(\vec{x}_2)$, the velocity at the Eulerian grid will be updated as

$$\tilde{\vec{u}}(\vec{x}_1) = \vec{u}(\vec{x}_1) + \vec{f}(\vec{x}_1)\delta t, \quad \tilde{\vec{u}}(\vec{x}_2) = \vec{u}(\vec{x}_2) + \vec{f}(\vec{x}_2)\delta t, \tag{23}$$

where the notations $\tilde{\vec{u}}(\vec{x}_1)$ and $\tilde{\vec{u}}(\vec{x}_2)$ are used to indicate the velocity field after the boundary forcing. When this updated velocity field is interpolated back to the Lagrangian grid point, the updated velocity at the Lagrangian grid becomes

$$\tilde{\vec{U}}(\vec{X}) = \vec{U}(\vec{X}) + (\phi_1^2 + \phi_2^2)[\vec{U}^d(\vec{X}) - \vec{U}(\vec{X})]. \tag{24}$$

Obviously, unless $(\phi_1^2 + \phi_2^2) = 1$, the precise no-slip condition would not be satisfied at the Lagrangian point. However, as the weighting factors generated from regularized δ functions satisfy $\phi_1 + \phi_2 = 1$, with $0 \leqslant \phi_1 \leqslant 1$, $0 \leqslant \phi_2 \leqslant 1$, $(\phi_1^2 + \phi_2^2) = 1$ can be barely fulfilled. With multidirect forcing, Eq. (24) can be generalized to arbitrary two consecutive iterations as

$$\tilde{\vec{U}}^{(n+1)}(\vec{X}) = (\phi_1^2 + \phi_2^2)\vec{U}^d(\vec{X}) + (1 - \phi_1^2 - \phi_2^2)\tilde{\vec{U}}^{(n)}(\vec{X}), \tag{25}$$

which recursively results in

$$\tilde{\vec{U}}^{(n+1)}(\vec{X}) = \left[(\phi_1^2 + \phi_2^2) \sum_{m=0}^{n} (1 - \phi_1^2 - \phi_2^2)^m \right] \vec{U}^d(\vec{X}) + (1 - \phi_1^2 - \phi_2^2)^{(n+1)} \vec{U}(\vec{X}). \tag{26}$$

Since $(\phi_1^2 + \phi_2^2) \in (0, 1]$, when $n \to \infty$, the coefficient of the second term in Eq. (26) vanishes and the coefficient of the first term converges to 1. Precise no-slip boundary enforcement is then achieved. In fact, the multidirect forcing essentially

generates a boundary forces series, whose summation results in precise no-slip condition when the number of iterations goes to infinity.

With the above understanding, the technique to avoid iteration of the boundary force emerges. If the boundary force is amplified by a factor $\alpha = 1/(\phi_1^2 + \phi_2^2)$ in Eq. (22b), the updated velocity interpolated back to the Lagrangian grid would satisfy the no-slip condition precisely. It should be kept in mind that the above analysis is based on an ideal configuration with only one Lagrangian grid point and two Eulerian grid points. In practice, each Lagrangian grid point will interact with multiple Eulerian grid points and each Eulerian grid point will also interact with multiple Lagrangian grid points, amplifying the boundary force by a factor no longer completely eliminates the error of no-slip boundary enforcement. However, as will be shown shortly, this force amplification technique would actually work quite well in simulations. It typically achieves an accuracy of no-slip boundary enforcement that previously needs 8 to 10 iterations to realize. This is probably because the regularized δ functions restrict the interpolations to happen only locally among only a few Lagrangian and Eulerian grid points, which is not far from the ideal scenario discussed above.

For standard DI-IBM with the two-sided force distribution kernel, the amplification coefficient α^T should be defined in terms of individual Lagrangian grid point $\vec{X}_L$ as

$$\alpha^T(\vec{X}_L) = \frac{1}{\sum_{\vec{x}_E}[\delta_h(\vec{x}_E - \vec{X}_L)]^2}. \tag{27}$$

For many commonly used regularized δ functions, this amplification coefficient α^T is actually constant that needs no additional computation. For the three-point δ function introduced by Roma [34], $\alpha^T = 2$. For Peskin's four-point δ function [Eq. (2)] and the four-point cosine δ function, $\alpha^T = 8/3$.

For the single-sided kernel, the force amplification coefficient α^S should be calculated,

$$\alpha^S(\vec{X}_L) = \frac{1}{\sum_{\vec{x}_E \in \text{solid}}[\delta_h(\vec{x}_E - \vec{X}_L)]}, \tag{28}$$

in order to accommodate the single-sided nature of the boundary force distribution. Note that this force amplification coefficient in Eq. (28) is essentially the weighting factor $W(\vec{X}_L)$ that has already been computed in Eq. (30). As a result, the force-amplification technique requires no additional computation cost for both the standard DI-IBM with two-sided force distribution kernel and the single-sided kernel.

It should be noted that there is also an alternative way that can directly obtain boundary forces that fulfill no-slip condition on the Lagrangian grid by solving matrix systems generated from the back-and-forth interpolation processes between the Lagrangian and Eulerian grids [35]. While it is accurate, the implementation of this method could be cumbersome as it requires the computation of multiple inverse matrices [35]. Simplified versions of this method based on approximately solving the matrix and vector systems [36,37] were very recently proposed, which no longer precisely enforces the no-slip boundary condition on the Lagrangian grid, but being much easier to implement. Nevertheless, those simplified versions still require extra computational costs to formulate matrices and vectors systems based on the specific alignment between the Lagrangian and Eulerian grids. In this regard, our force amplification technique that does not introduce any additional computations is still much simpler.

C. A force-amplified, single-sided DI-IBM algorithm

Combining the single-sided force distribution kernel and the force amplification technique, we propose a new DI-IBM that not only restores correct velocity gradient computation in the vicinity of a solid boundary but also improves the accuracy of no-slip boundary enforcement. This DI-IBM also contains three key steps.

Step 1: Interpolate the unforced velocity at the Lagrangian grid point $\vec{X}_L$ from neighboring Eulerian grid points $\vec{x}_E$ following precisely the standard DI-IBM:

$$\vec{U}(\vec{X}_L) = \sum_{\vec{x}_E} \vec{u}(\vec{x}_E)\delta_h(\vec{x}_E - \vec{X}_L)h^3. \tag{29}$$

In addition, sum up the total weight contributed by the Eulerian grid points locating in the solid phase in the interpolation, use this total weight as the force amplification coefficient i.e.,

$$W(\vec{X}_L) = \sum_{\vec{x}_E \in \text{solid}} \delta_h(\vec{x}_E - \vec{X}_L)h^3, \quad \alpha^S(\vec{X}_L) = W(\vec{X}_L). \tag{30}$$

Step 2: Calculate the boundary force at the Lagrangian grid, the force is amplified by dividing the force amplification coefficient. This is the essential difference between the present IBM algorithm and the algorithm of Ji *et al.* in Ref. [33]:

$$\vec{F}(\vec{X}_L) = \frac{1}{\alpha^S(\vec{X}_L)} \frac{\vec{U}^d(\vec{X}_L) - \vec{U}(\vec{X}_L)}{\delta t}. \tag{31}$$

Step 3: Distribute the boundary force calculated in Step 2 back to *only* the neighboring Eulerian grid points in the solid phase:

$$\vec{f}(\vec{x}_E \in \text{solid}) = \sum_{\vec{X}_L} \frac{\vec{F}(\vec{X}_L)\delta_h(\vec{x}_E - \vec{X}_L)\Delta V_L}{W(\vec{X}_L)}. \tag{32}$$

IV. NUMERICAL VALIDATIONS

In this section, the proposed DI-IBM will be examined in four purposely selected flows, including three laminar flows, a plane channel flow that provides numerical support for the analysis presented in Sec. II, a Taylor-Couette flow that is purely torque driven to assess the impact of lacking the torque conservation in the single-sided force distribution kernel, the sedimentation of a cylinder in quiescent fluid to benchmark the performance of the proposed DI-IBM for tracking particle motion, and one turbulent flow, a decaying homogeneous isotropic turbulent flow laden with a few thousand of resolved particles to examine the accuracy and numerical stability for high Reynolds number flows.

A. A steady-state plane channel flow

The first case tested is the steady-state plane channel flow discussed in Sec. II. The flow solver used here is the lattice Boltzmann method (LBM) on a D2Q9 lattice grid with a single relaxation time [38]. The no-slip condition on the two
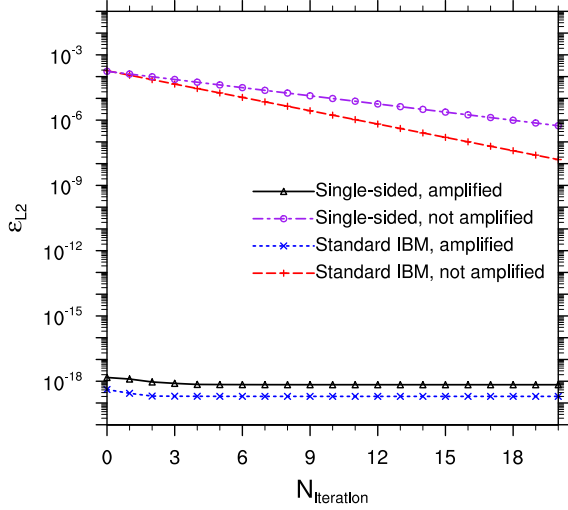
FIG. 3. Comparisons of L2 errors of no-slip boundary enforcement on plane channel walls with and without the force amplification technique.

channel walls are handled by DF-IBM. For demonstration purposes, we set the flow Reynolds number $\mathrm{Re} = Hu_c/\nu = 100$, where $H = 80h$ is the channel width, $u_c = 0.05$ is the steady-state flow velocity at the centerline of the channel, and ν is the kinematic viscosity. The simulation was conducted in a computational domain of $nx \times ny = 4 \times 89$. In the wall-normal direction (y), the fifth and 85th grid points are the first and last grid points in the fluid region, which are located on each channel wall. Other grid points are in the solid region. The driving force ρg is only applied to the grid points located in the fluid region. The four-point cosine-shaped δ function, i.e., Eq. (8) with $\delta = 2$, is used for interpolations between the Eulerian and Lagrangian grids to be consistent with the analysis in Sec. II.

To begin, let us validate the force amplification technique introduced in Sec. III B for more accurate no-slip boundary enforcement. The L2 errors of the no-slip condition, defined

as

$$\varepsilon_{\mathrm{L2}} = \sqrt{\sum_{\vec{X}_L}[\vec{U}(\vec{X}_L) - \vec{U}^d(\vec{X}_L)]^2} \Big/ \sqrt{\sum_{\vec{X}_L}[\vec{U}^d(\vec{X}_L)]^2}, \quad (33)$$

as functions of the iteration times are presented in Fig. 3 for both the standard DI-IBM with two-sided force distribution kernel and the proposed DI-IBM with single-sided kernel, with and without applying the force amplification. We would like to emphasize that here and after, $N_{\mathrm{iteration}}$ means the number of iterations for the boundary force of DI-IBM to enforce the no-slip condition, which should be distinguished from the possible iteration in flow solvers. For both kernels, when the boundary force is amplified by the proper coefficient, the L2 errors directly converge to the machine round-off error without the need of iteration. On the contrary, without the force amplification, the L2 errors of no-slip condition gradually reduce as the number of iterations increases. Compared to the standard two-sided force distribution kernel, the L2 error of the single-sided kernel has a slower convergence rate. This is because when the boundary force is applied single-sidedly, only the fluid velocity in the solid region is being updated during the iteration. It should be noted that the force amplification technique works extremely well in this plane channel flow case because the simple geometry and the periodicity in the streamwise direction together create an ideal alignment between the Lagrangian and Eulerian grids. In general cases with curved geometries (will be seen in other cases), the force amplification technique can still result in more accurate no-slip boundary enforcement, but the error wound not vanish.

The profiles of the velocity and velocity gradient obtained with different DI-IBM algorithms are shown in Figs. 4 and 5, respectively. Here we compared the results from four DI-IBMs, the standard DI-IBM, the proposed DI-IBM, and two DI-IBMs with nonzero retraction distances $r_d = 0.3h$ and $r_d = 2h$. The first retraction distance is the optimal retraction distance that should result in accurate velocity field, while the second retraction distance is expected to fully restore the
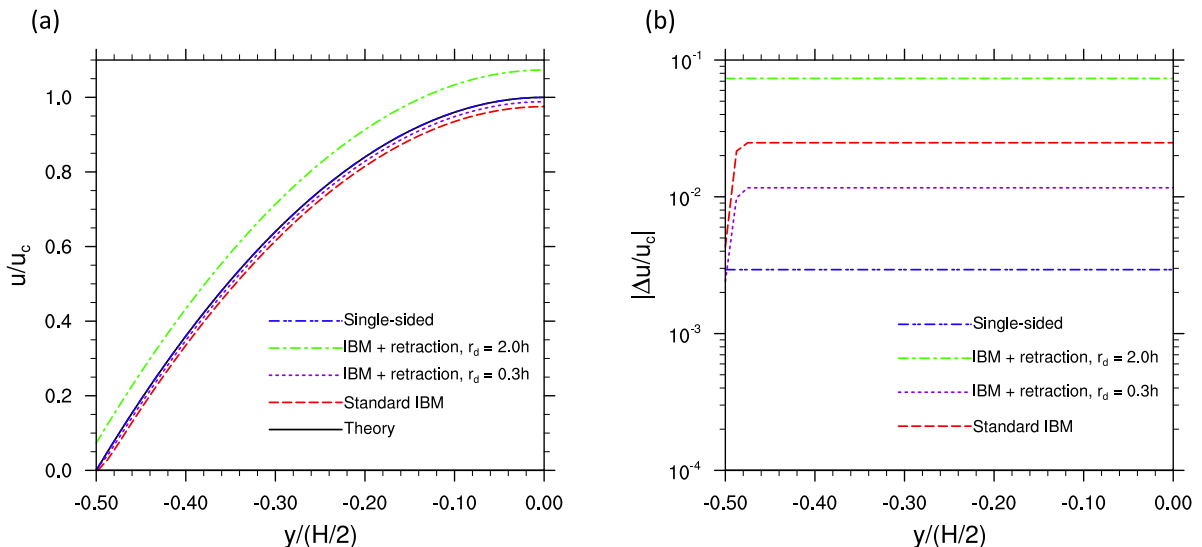

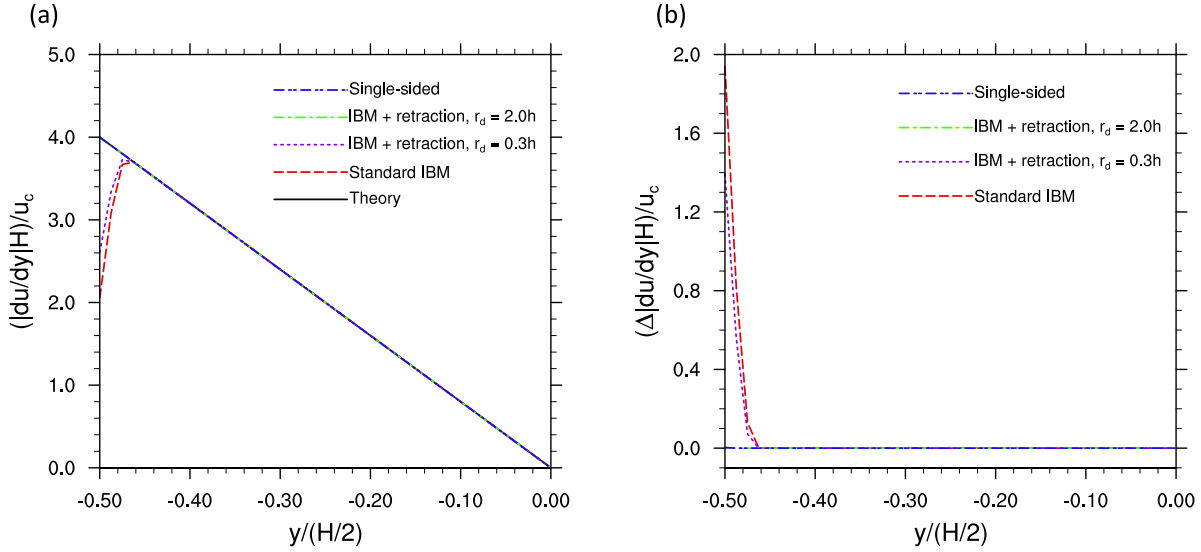FIG. 4. Profiles of velocity in a steady-state plane channel flow.

(a)

(b)

FIG. 5. Profiles of velocity gradient in a steady-state plane channel flow.

correct velocity gradient results. The velocity gradient results du/dy in all cases are computed as

$$\frac{du}{dy}(x) = \frac{4u(x+h) - 3u(x) - u(x+2h)}{2h} \quad (34)$$

near the channel wall and

$$\frac{du}{dy}(x) = \frac{u(x+h) - u(x-h)}{2h} \quad (35)$$

away from the channel wall, in order to avoid involving the grid points inside the solid phase.

As shown in Fig. 4, a visible relative error about 2.5% is observed in the velocity profile of the standard DF-IBM. This error, as analyzed in Sec. II B, is caused by the nonzero boundary force inside the fluid region and being a constant outside the diffused layer. However, the magnitude of this relative error is larger than the corresponding error predicted by Eq. (10), which is about 1.5%. The deviation is probably because the analysis in Sec. II B is based on a continuous boundary force distribution, while in the actual simulations, the boundary force is only distributed on a few Eulerian grid points. With the retraction distance $r_d = 0.3h$, the numerical error in the velocity result does decrease to 1.1%, but it is not fully eliminated following the prediction, probably due to the same reason that the boundary force is distributed discretely in the simulation. These two DF-IBMs both lead to significant errors (48.6% for the standard DF-IBM and 34.8% with $r_d = 0.3h$) in the velocity gradient computation near the channel wall, but the results matches perfectly with the theory outside the diffused layer, as shown in Fig. 5. With a large retraction distance $r_d = 2h$, correct velocity gradient computation can be fully restored, but the numerical error in the velocity result becomes significant. Therefore, employing a large retraction distance should still be avoided. Finally, with the single-sided force distribution kernel, not only correct velocity gradient computation is achieved in the whole fluid region, but also the numerical error in the velocity result is further reduced to 0.3%. This simple test provides solid evidence that there is an essential difference between the proposed DI-IBM and

the approach of retracting the Lagrangian grid from the solid surface. The former can ensure accurate velocity and velocity gradient computation at the same time, which is not achieved by the latter.

B. A steady-state Taylor-Couette flow

The second validation case is a steady-state two-dimensional Taylor-Couette flow. This case is chosen because (1) it has curved solid boundaries and (2) it is a pure torque driven rotational flow that can be used to assess the impact of losing torque conservation when interpolating the boundary force from Lagrangian to Eulerian grid. As sketched in Fig. 6, this flow is bounded and driven by two rotating concentric cylinders with radii R_1 and R_2 and angular velocity Ω_1 and
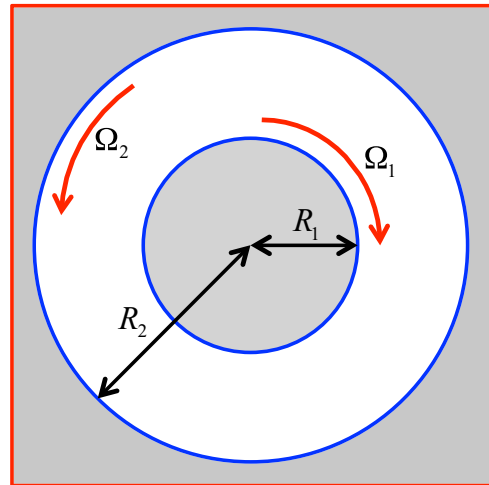

FIG. 6. Sketch of circular Taylor-Couette flow. The red frame indicates the edge of the computational domain. The two blue circles are the physical boundaries, which confine the fluid region in white color. The gray region is the virtual fluid region, which is physically irrelevant to the problem, but also filled with fluid with IBM.
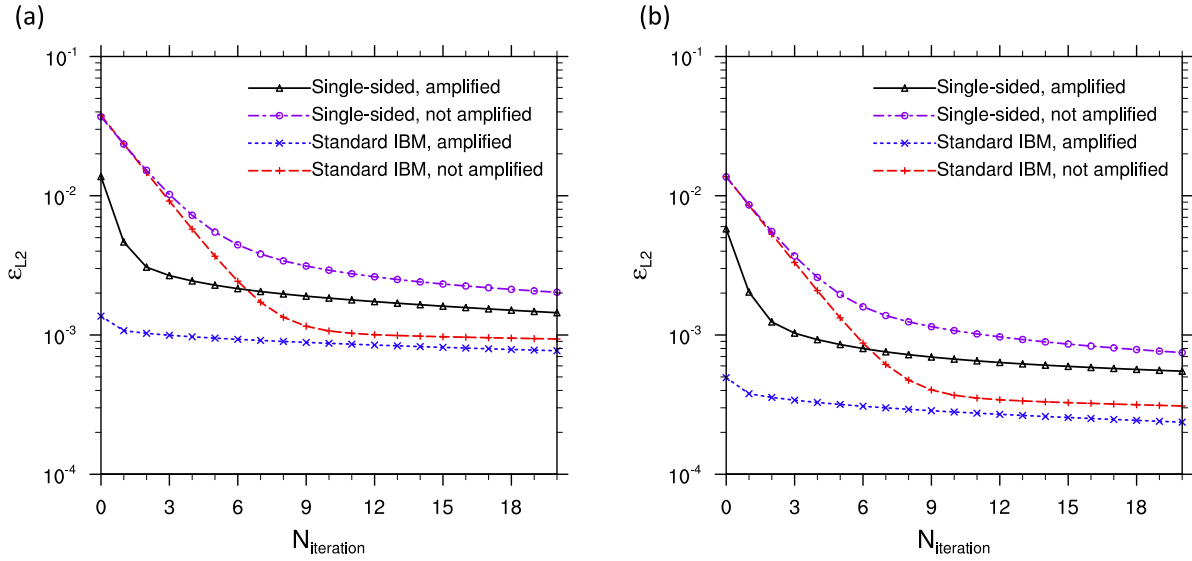
(a)

(b)

FIG. 7. Comparisons of L2 errors of no-slip boundary enforcement on (a) inner cylinder and (b) outer cylinder with and without the force amplification technique.

Ω_2, respectively. This flow also has analytic solution

$$u_\theta = \frac{1}{r}\frac{\Omega_1 - \Omega_2}{R_1^{-2} - R_2^{-2}} + \frac{\Omega_2 R_2^2 - \Omega_1 R_1^2}{R_2^2 - R_1^2}r, \quad u_r = 0. \quad (36)$$

For demonstration purposes, in the numerical test, we set the outer cylinder static and the inner cylinder rotating. The Reynolds number of the flow $\mathrm{Re} = (R_2 - R_1)\Omega_1 R_1/\nu$ is set to 15, with $R_1 = 50h$, $R_2 = 100h$, and $\Omega_1 R_1 = 0.05$. The flow is again solved by LBM on a D2Q9 lattice grid. The size of the computational domain is $nx \times ny = 211 \times 211$ with the centers of two cylinders located at $(x_c, y_c) = (105.5h, 105.5h)$. The no-slip boundary treatment on the two cylinder surfaces is handled with tested IBM algorithms. From now on, the four-point δ function by Peskin [2], i.e., Eq. (2), will be used to execute the interpolations in all DI-IBMs. For this specific flow, the treatment on the edges of the computation domain (the red square in Fig. 6) could also significantly affect the accuracy of the results [16]. In the numerical tests, the stress-free condition, i.e., $\partial u_t/\partial \mathbf{n} = 0$, $u_n = 0$, where u_t and $u_n = 0$ are the tangential and normal velocity component and $\mathbf{n}$ is the normal direction on each edge, is applied to the edges of the computational domain.

We again begin by examining the force amplification technique for enforcing the no-slip boundary condition. The L2 error of the no-slip boundary enforcement on the inner and outer cylinders with and without applying the force amplification technique are shown in Fig. 7, for both the two-sided and single-sided force distribution kernels. A few observations can be made from Fig. 7. First, for general curved boundaries, the force amplification no longer immediately achieves precise no-slip condition. This is because the force amplification coefficients were obtained from an ideal case, which do not fit perfectly for general cases. Second, the standard DI-IBM kernel provides smaller errors compared to the single-sided kernel. This is also understandable because with the single-sided kernel, only the velocity field on the solid side of the boundary is updated, which affects the convergence of the error. A same observation was also made by Ji *et al.* [33]. Third, for both kernels, applying the force amplification technique could still significantly reduce the necessity for boundary force iterations to achieve an accurate no-slip boundary enforcement. This technique functions particularly well with the standard DI-IBM kernel. Without any iteration, an L2 error of no-slip boundary enforcement has been achieved that previously would require eight steps of iterations. For the single-sided force distribution kernel, even with the force amplification technique, a few iteration steps are still necessary in order to enforce a sufficiently accurate no-slip boundary. Based on the results in Fig. 7, when the force amplification technique is used, we recommend zero iteration and three iterations for the standard DI-IBM kernel and the single-sided kernel, respectively, to balance the computation cost and the accuracy of no-slip boundary enforcement.

Next, the profiles of velocity u_θ and velocity gradient du_θ/dr with different DF-IBM algorithms are compared with the analytic solutions in Figs. 8 and 9, respectively. The compared IBM algorithms are the same four algorithms as in the plane channel flow case. For each algorithm, the force amplification technique is applied and the boundary force is iterated three times in order to achieve more accurate no-slip condition on the two cylinders. Although such iteration does not significantly improve the no-slip boundary enforcement when the two-sided force distribution kernel is adopted, the same amount of iterations are kept in all cases just to reduce discrepancies among different cases.

As shown in Fig. 8, the standard DF-IBM again results in certain visible error in the velocity profile, which can be eliminated when a proper retraction distance $r_d = 0.3h$ of the Lagrangian grid is applied. The retraction of the Lagrangian grid also leads to more accurate velocity gradient computation in the region away from the cylinder surfaces, as shown in Fig. 9, but significant errors in velocity gradient computation near the cylinder surfaces still remain. On the contrary, the single-sided force distribution kernel not only improves the accuracy of velocity field, but also restores correct velocity gradient
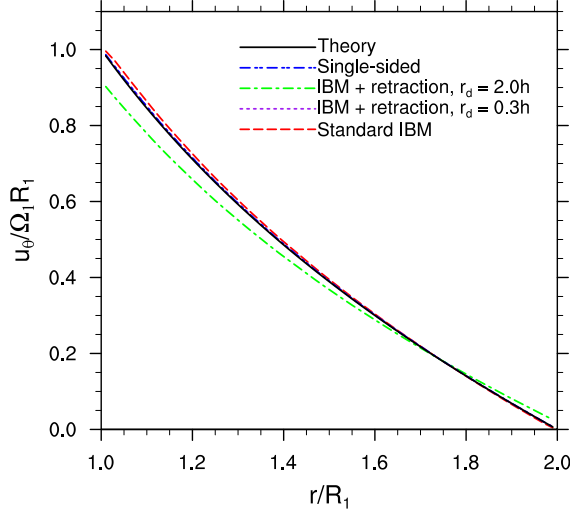
FIG. 8. Comparison of velocity profiles with different DI-IBM algorithms in a steady-state Taylor-Couette flow.

computation in the whole fluid domain. We emphasize that an upwind scheme like finite difference approximation, i.e., Eq. (34) should be used, in order to avoid the contamination from the solid region in the computation of velocity gradient in all cases. Finally, with a large retraction distance $r_d = 2h$, both the velocity and velocity gradient results contain significant errors through the whole fluid domain. This is consistent with the observations reported in the literature, e.g., Ref. [9]. From now on, the inaccurate results associated with a large retraction distance will no longer be included in our discussions. It should also be quite clear at this point that retracting the Lagrangian grid is *not* capable of restoring the correct velocity gradient results in the vicinity of a solid surface.

As mentioned in Sec. III A, the single-sided strategy of distributing the boundary force no longer conserves torque during the interpolation of boundary force from Lagrangian to
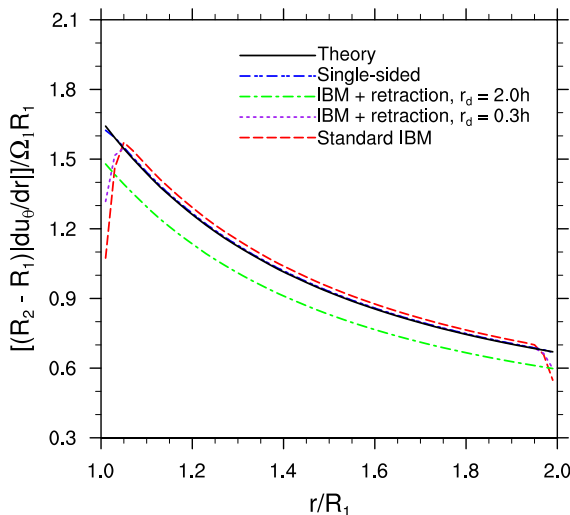

FIG. 9. Comparison of velocity gradient profiles with different DI-IBM algorithms in a steady-state Taylor-Couette flow.
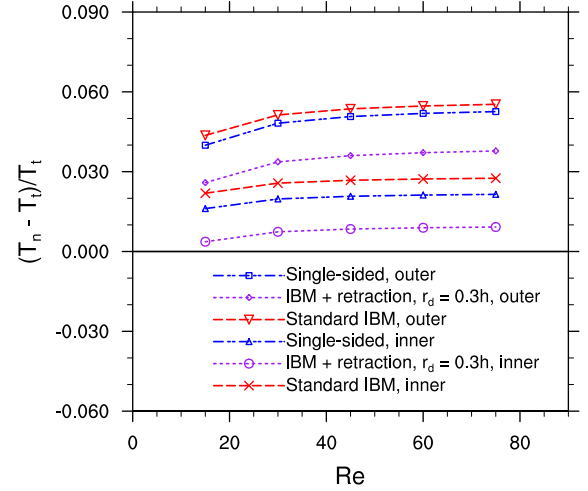

FIG. 10. The relative error of hydrodynamic torque on the inner and outer cylinders computed with different DI-IBM algorithms.

Eulerian grid. This issue and its potential impact on the simulation results need a further investigation, especially when the rotational motion is playing an important role. For this pure torque-driven Taylor-Couette flow, the relative errors of the hydrodynamic torque on the inner and outer cylinder, i.e., $(T_n - T_t)/T_t$, where T_n and T_t are the numerical and theoretic torque, respectively, are computed and compared in Fig. 10 for different Reynolds numbers. All those hydrodynamic torque results are computed based on the Lagrangian grid and with a grid resolution of $R_1 = 50$. As shown in Fig. 10, all three examined IBM algorithms overpredict the hydrodynamic torque by 1% to 5%. The errors of hydrodynamic torque on the outer cylinder are always larger than their counterparts on the inner cylinder, as the former is more influenced by the boundary treatment on the edges of the computation domain. DI-IBM with a retraction distance of $0.3h$ always gives the most accurate hydrodynamic torque predictions, followed by the proposed DI-IBM with single-sided force distribution kernel, then the standard DI-IBM. Indeed the proposed DI-IBM with single-sided force distribution kernel does not conserve torque nor it is the optimal method lead to the most accurate hydrodynamic torque results, it still offers slightly improved hydrodynamic torques compared to the standard DI-IBM.

At last, at Re = 15, we vary the grid resolutions to simulate the same flow and measure the the orders of accuracy of velocity, velocity gradient, and hydrodynamic torque computations with different IBM algorithms. The errors of the torque are simply measured as $|(T_n - T_t)/T_t|$, where T_n and T_t are the numerical and theoretical torque, respectively. Here we show only the torque on the inner cylinder because the outer cylinder is very close to the edge of the computational domain (see Fig. 6), which make its torque significantly affected by the boundary treatment on the edge of the computational domain, rather than the boundary treatment on the physical boundary itself [16]. The errors of velocity and velocity gradient are evaluated over the whole fluid domain as

$$\varepsilon = \frac{\sqrt{\sum_{(x,y)}[q_n(x, y) - q_t(x, y)]^2}}{\sqrt{\sum_{(x,y)}[q_t(x, y)]^2}},\qquad(37)$$
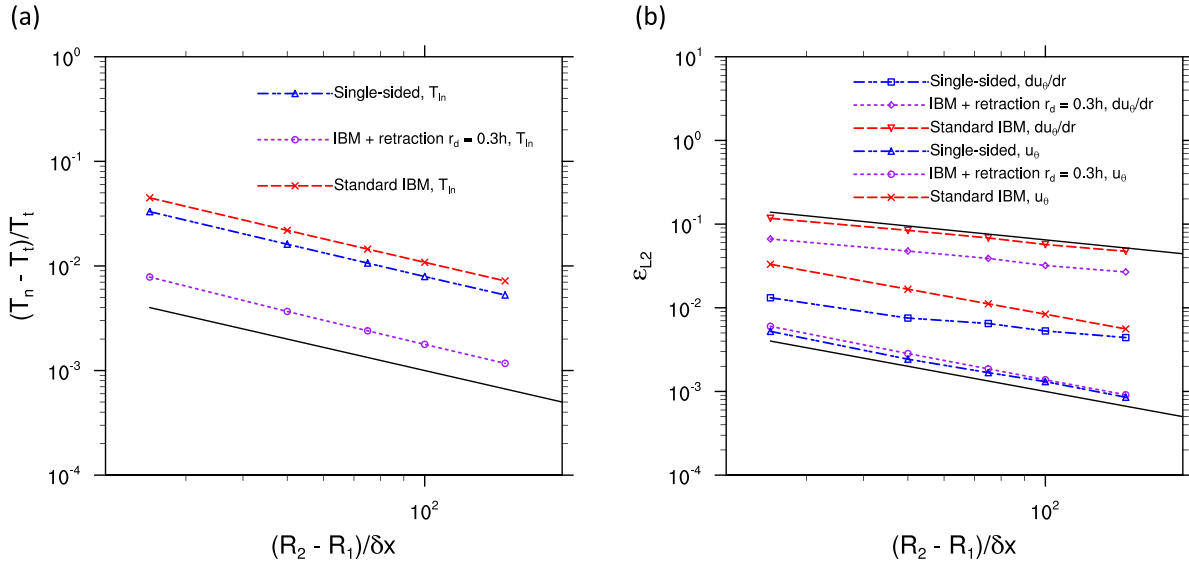
(a)

(b)

FIG. 11. Orders of accuracy for velocity, velocity gradient, and hydrodynamic torque computation with different DI-IBM algorithms. The solid black line in the left plot shows the reference of the slope -1.0. The top and bottom black solid lines in the right plot are reference lines for the slope -0.55 and -1.0, respectively.

where q_n and q_t should be replaced with the numerical and theoretical values of the velocity and the velocity gradient. As compared in Fig. 11, compared to the results of the standard IBM, both retracting the Lagrangian grid by a distance of $0.3h$ and the single-sided force distribution kernel could reduce the magnitudes of numerical errors in the fluid velocity field and hydrodynamic torque computations. The retracting approach gives the smallest numerical errors in the hydrodynamic torque computations, while the single-sided kernel leads to the most accurate fluid field among three tested DI-IBMs. On the other hand, the orders of accuracy of the results are not improved. The fluid velocity and hydrodynamic torque still have a first-order accuracy, and the velocity gradient computation has an accuracy about 0.55 order. The orders of accuracy in the diffused-interface based IBM are constrained by the use of regularized δ function [2,16]. Since the three IBM algorithms still rely on the same regularized δ functions to interpolate the velocity and force back and forth between the Lagrangian and Eulerian grids, the orders of accuracy are not improved.

C. Sedimentation of a cylindrical particle in quiescent fluid

We now move to the validation cases with freely moving curved boundaries. The third validation case is the sedimentation of asymmetrically placed cylindrical particle in an infinite long channel filled with quiescent fluid, as depicted in Fig. 12. For test purposes, we set $D = 0.1$ cm, $L = 4$ cm, $H = 0.4$ cm, $a = 0.324$ cm, $\nu = 0.01$ cm^2/s, $\rho_f = 1$ g/cm^3, $g = 980$ cm^2/s^2, and the density ratio $\rho_p/\rho_f = 1.03$. This is to match the arbitrary Lagrangian Eulerian (ALE) simulation performed by Hu *et al.* [39], whose results will be used as benchmark results for our simulations. Under this setting, the terminal particle Reynolds number is Re$_p = \frac{u_T D}{\nu} = 8.441$,

where u_T is the terminal settling velocity of the particle. Due to its initial asymmetric location, the particle would experience both transverse immigration and rotational motion during its sedimentation. In this case, we mainly examine how the proposed single-sided force distribution strategy would
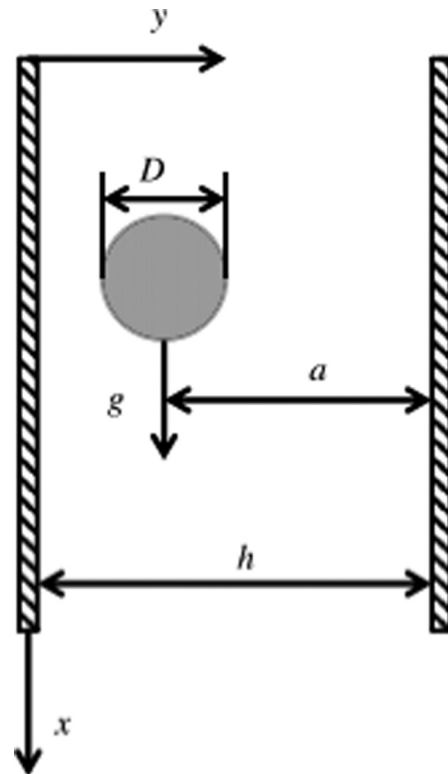

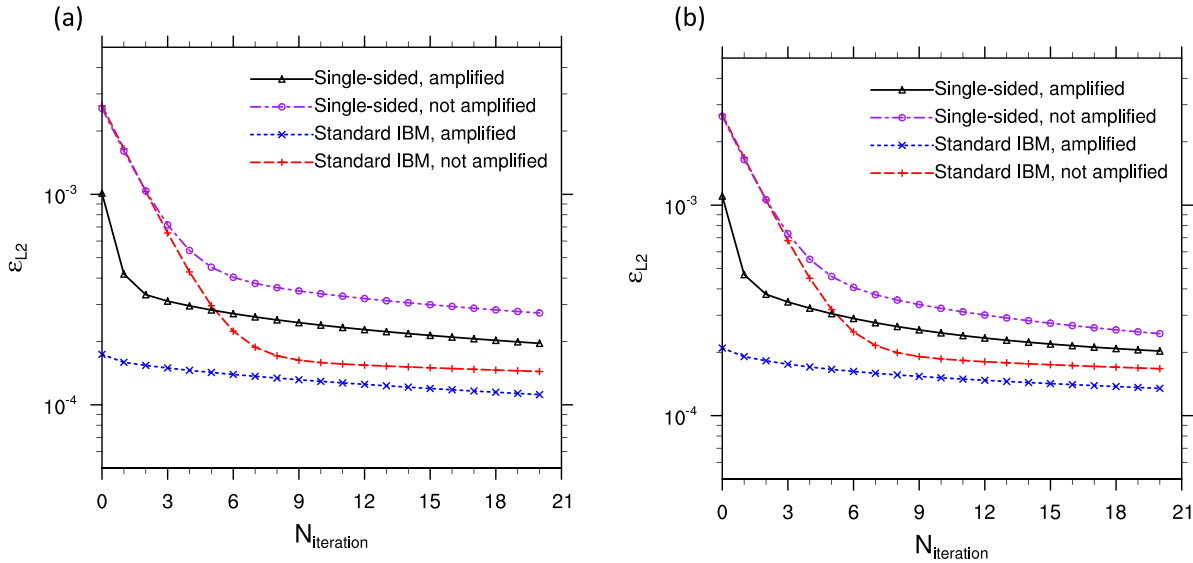FIG. 12. Sketch of a cylindrical particle settling under gravity in a vertical channel.

FIG. 13. The L2 errors of no-slip enforcement with and without the force amplification technique: (a) $t = 0.370$ s, (b) $t = 2.963$ s.

perform compared to those DI-IBMs based on two-sided force distribution. The flow is solved by LBM with a D2Q9 lattice grid in a $nx \times ny = 1200 \times 120$ domain. For demonstration purpose, the grid resolution of all simulations is set to $D = 30h$. When the particle to fluid density ratio is close to unity (in this case $\rho_p/\rho_f = 1.03$), the inertia of fluid inside the particle have to be computed explicitly to avoid numerical instability [10,40]. Unless otherwise specified, the scheme proposed by Kempe *et al.* [10] is employed to compute the inertia of fluid inside the particle.

First, the L2 errors of no-slip boundary enforcement on the particle surface with and without the force amplification technique are shown for the standard DI-IBM and the proposed DI-IBM with single-sided force distribution in Fig. 13. Due to the unsteadiness of the particle motion, we calculate these L2 errors at two specific time frames, one at the early stage, $t = 0.370$ s, and the other one at the later stage, $t = 2.963$ s. At both time frames, we confirm the force amplification technique significantly reduces the necessary iteration steps to achieve the same accuracy of no-slip enforcement. The information extracted from Fig. 13 further strengthens the observations made in Sec. IV B for the Taylor-Couette flow, which are not repeated.

The results of particle motion, i.e., particle trajectory, particle angular velocity, particle velocity in both directions are presented in Fig. 14. In all these simulations, the force amplification technique has been used with three iterations. As shown in Fig. 14, compared to the standard DF-IBM, both retracting the Lagrangian grid by a distance of $r_d = 0.3h$ and the single-sided force distribution strategy lead to results of particle motion that match better with the benchmark results from the ALE simulation. Particularly, the results of the particle angular velocity have also been improved with the single-sided force distribution strategy. This again validates that, although the local torque is not conserved by the single-sided kernel during the interpolation of boundary force, its

global impact on the simulation results of particle velocity and trajectory is negligible.

However, there is one trade-off in using the single-sided force distribution strategy for flows involving freely moving solid objects. When a numerical method that solves the N-S equations on a structured Eulerian grid is used to resolve the interactions between a solid object and its surrounding fluid, the strengths of the interactions would have a slight dependence on the specific alignment between the fixed Eulerian mesh lines and the solid object. This slight dependence is known as the "grid-locking" effect [9]. When the solid object moves continuously with respect to the mesh lines, the results of instantaneous force and torque on this solid object contain high-frequency fluctuations. The use of regularized δ functions to spread the boundary force into a finite-width band in DF-IBM serves as a smoothing technique for these high-frequency fluctuations [3,9]. Since the single-sided force distribution kernel reduces the width of forced band by a half, the grid-locking effect becomes more significant, and more severe force fluctuations would be observed. In Fig. 15 the instantaneous drag and lift forces acting on the settling particle with different DI-IBMs are compared. The single-sided force distribution kernel does result in more severe force fluctuations compared to its two-sided counterparts.

When the motion of the particle reaches a steady state, the force fluctuations become periodic with a period h/u, as the repeated alignment will restore once the particle moves over exactly one grid space. This is confirmed by the zoom-in plots in Fig. 15. At the steady state, the particle velocity in the vertical direction is about $u = 0.84$ cm/s, and $h = 1/300$ cm, therefore the period of the force fluctuations is about $0.004s$, being consistent with the observation in Fig. 15. It is worth mentioning that while being undesired, these high-frequency fluctuations in the instantaneous force will be automatically smoothed out and do not affect the results of particle velocity and trajectory.
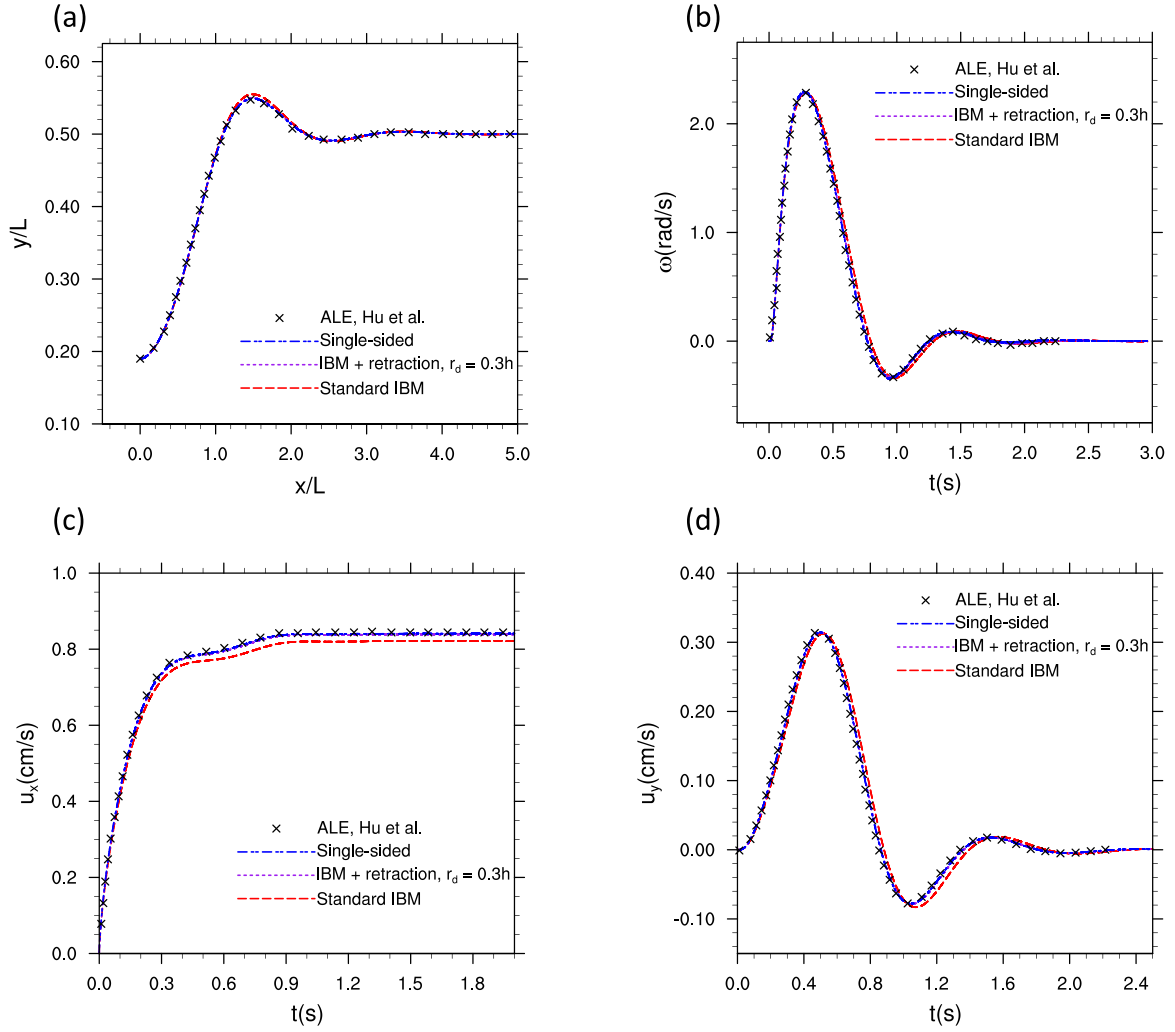
FIG. 14. Comparisons of the motion of a settling particle in a vertical channel: (a) particle trajectory, (b) particle angular velocity, (c) particle velocity in vertical direction, (d) particle velocity in horizontal direction.

To illustrate this point, let us assume the high-frequency force fluctuation can be expressed as a single sine function

$$f'(\tau) = \alpha \sin(2\pi\omega\tau)F, \qquad (38)$$

where F is the characteristic force, α and ω are the amplification and frequency of the force fluctuation. By definition, this force fluctuation results in a velocity difference between two arbitrary time frames t_1 and t_2 as

$$\Delta u = \frac{1}{M_p} \int_{t_1}^{t_2} \alpha \sin(2\pi\omega\tau) F \, d\tau \leqslant \frac{\alpha F}{M_p \pi \omega}, \qquad (39)$$

where M_p is the mass of the particle. Since the force fluctuation is caused by the grid locking, the period of the force fluctuations is h/u. We can therefore replace ω in Eq. (39) by u/h and have

$$\frac{\Delta u}{u} \leqslant \frac{\alpha F h}{M_p \pi u^2}. \qquad (40)$$

We can further use $F = \frac{1}{2} C_d \rho_f u^2 A$, where C_d is the drag coefficient, A is the reference area of the particle, to replace

the characteristic force in Eq. (40). The magnitude of the velocity fluctuation will then becomes

$$\frac{\Delta u}{u} \leqslant \frac{\alpha \left(\frac{1}{2} C_d \rho_f u^2 A\right) h}{M_p \pi u^2} = \frac{2 C_d}{\pi^2} \frac{h}{D} \frac{\rho_f}{\rho_p} \alpha. \qquad (41)$$

Evaluating this magnitude with the parameters in the simulation, i.e., $D = 30h$, $C_d \approx 2$, $\pi^2 \approx 10$, $\rho_f/\rho_p \approx 1$, we have

$$\frac{\Delta u}{u} \leqslant 0.0133\alpha. \qquad (42)$$

Even with a significant force fluctuation as large as $\alpha = 50\%$, the resulting velocity fluctuation is less than 1%, which is hardly visible in the plots. Without introducing any additional smoothing techniques, the results of the particle velocity, angular velocity, and particle trajectory shown in Fig. 14 are already sufficiently smooth.

We also noted that the magnitude of force fluctuation is strongly affected by how accurate the inertia of the fluid inside the particle is computed. This computation is need since
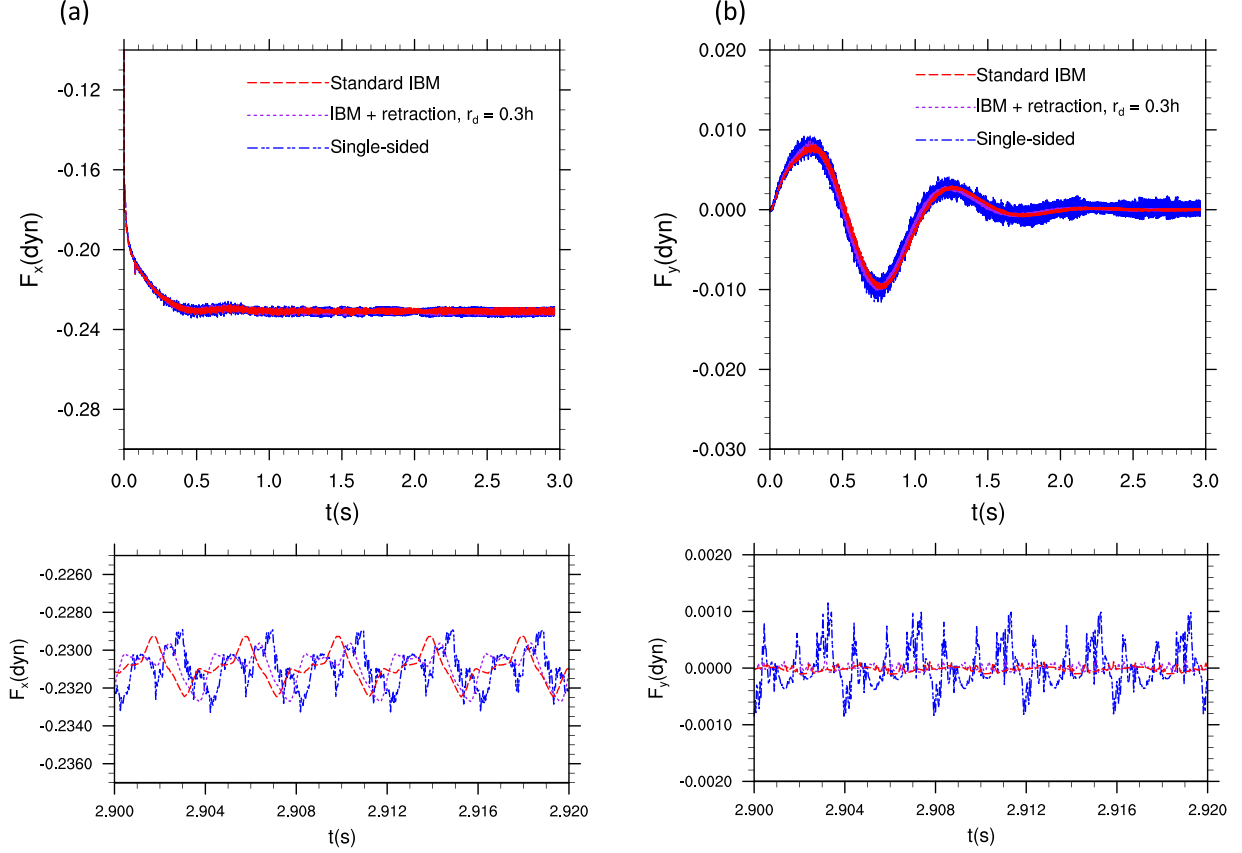
(a)

(b)

FIG. 15. Results of instantaneous hydrodynamic force in (a) the vertical direction and (b) the horizontal direction with different DI-IBM algorithms.

the kinematic equations of particle motion with DI-IBM are written as [3,9]

$$M_p \frac{d\vec{v}}{dt} \approx -\sum_{\vec{X}_L} \vec{F}(\vec{X}_L)\Delta V_m + \rho_f \frac{d}{dt} \int_{V_p} \vec{u} \, dV + (\rho_p - \rho_f)V_p \vec{g},$$

(43a)

$$I_p \frac{d\vec{\omega}}{dt} \approx -\sum_{\vec{X}_L} \vec{r}_L \times \vec{F}(\vec{X}_L)\Delta V_m + \rho_f \frac{d}{dt} \int_{V_p} (\vec{r} \times \vec{u}) \, dV,$$

(43b)

where V_p, M_p, and I_p are the volume, mass, and moment of inertia of the particle, respectively. $\vec{r}_L = \vec{X}_L - \vec{X}_c$ is the position vector of Lagrangian grid point $\vec{X}_L$ relative to the particle centroid $\vec{X}_c$. The second term on the right-hand side of each equation in Eqs. (43) is the momentum and angular momentum change contributed by the inertia of the virtual fluid inside the particle, which is nonzero when the particle moves. There are different ways in the literature to measure the magnitude of those terms. For example, Kempe *et al.* [10] explicitly computed the two integrals in Eqs. (43) as

$$\int_{V_p} \vec{u} \, dV = \sum_{1}^{n_x} \sum_{1}^{n_y} \sum_{1}^{n_z} \vec{u}_{i,j,k} h^3 \alpha_{i,j,k},$$

$$\int_{V_p} \vec{r} \times \vec{u} \, dV = \sum_{1}^{n_x} \sum_{1}^{n_y} \sum_{1}^{n_z} \vec{r}_{i,j,k} \times \vec{u}_{i,j,k} h^3 \alpha_{i,j,k},$$

(44)

where the volume fraction $\alpha_{i,j,k}$ of grid cell (i, j, k) in the solid region is computed as

$$\alpha_{i,j,k} = \frac{\sum_{l=1}^{8} -\phi_l H(-\phi_l)}{\sum_{l=1}^{8} \|\phi_l\|},$$

(45)

where the sum is over all eight corners of the grid cell, ϕ_l is a signed distance function, H is the Heaviside function.

Alternatively, Feng and Michaelides [40] approximated the inertia of the fluid inside the particle as

$$\rho_f \frac{d}{dt} \int_{V_p} \vec{u} \, dV = \rho_f V_p \frac{\vec{v}^n - \vec{v}^{n-1}}{\delta t},$$

$$\rho_f \frac{d}{dt} \int_{V_p} (\vec{r} \times \vec{u}) \, dV = I_p \frac{\rho_f}{\rho_p} \frac{\vec{\omega}^n - \vec{\omega}^{n-1}}{\delta t},$$

(46)

where $\vec{v}^n$ and $\vec{v}^{n-1}$, $\vec{\omega}^n$ and $\vec{\omega}^{n-1}$, are particle translational and angular velocities at the current and last time steps, respectively. This approximation circumvents the explicit computation these terms but can still maintain numerical stability when the particle to fluid density ratio is close to unity.

The above two methods were found to result in similar levels of grid-locking effects with DF-IBM with two-sided force distributions in our previous study [16]. However, when the single-sided force distribution strategy is used, the method of Kempe *et al.* [10] suppresses the grid-locking effects much better compared to the alternative method by Feng and Michaelides [40]. In Fig. 16 we compare the instantaneous
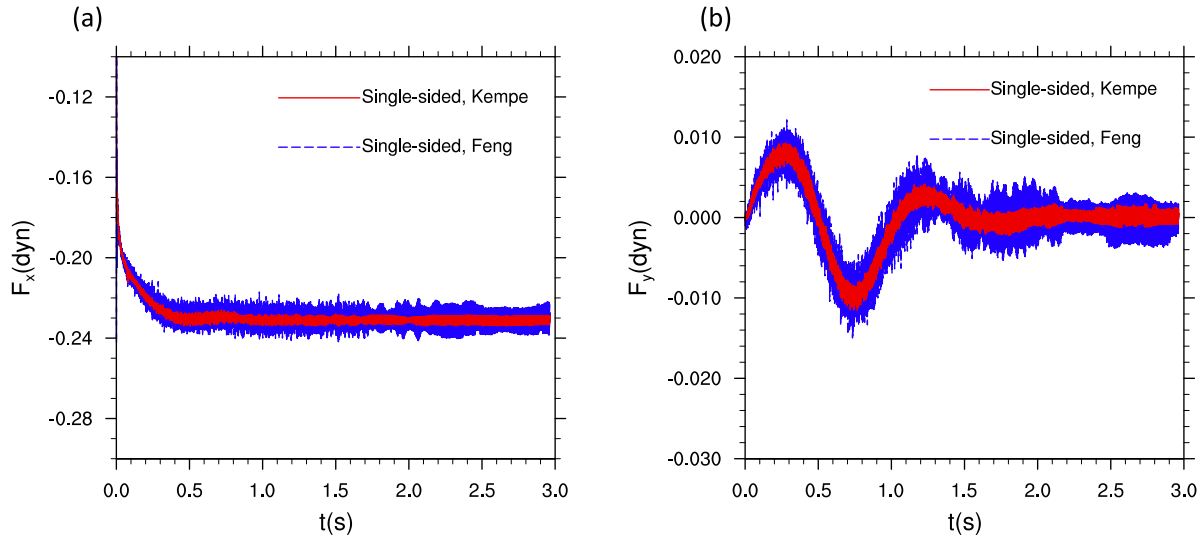
FIG. 16. Results of instantaneous hydrodynamic force in (a) the vertical direction and (b) the horizontal direction with different ways of computing the inertia of virtual fluid inside the settling particle.

drag and lift forces resulting from these two methods with the single-sided force distribution strategy. The explicit method of Kempe *et al.* gives much smaller force fluctuations. Since the boundary force is only applied to grid points on one side of the solid boundary, the flow inside the particle is more strongly perturbed. Under this condition, an explicit computation of the inertia of the flow inside the particle should be more accurate, which helps to suppress the grid-locking effects.

D. A particle-laden decaying homogeneous isotropic turbulence

While we have demonstrated in the last validation test that the more severe force fluctuations generated by the single-sided force distribution do not decrease the accuracy of particle motion, when the flow Reynolds number increases, such high-frequency fluctuations may not be fully dissipated and it might trigger some numerical instabilities in the simulation. To explore the potential of the single-sided kernel in high Reynolds number flow simulations, we directly examine the kernel in a more complicated decaying homogeneous isotropic turbulent flow laden with a few thousand of resolved spherical particles.

The properties of the carrier flow can be found in Table 1 of Ref. [15]. For the properties of the dispersed particle phase, we follow the parameter settings of "case 512" in Table 2 of Ref. [15], and these parameters are no longer repeated here. The particle Reynolds number $Re_p = u^0_{r.m.s.} D / \nu$, where $u^0_{r.m.s.}$ is the initial r.m.s. velocity of the background turbulence, D is the particle diameter, is 94.3. The simulation is conducted with two different grid resolutions, 512^3 and 1024^3. The flow solver is LBM based on a D3Q19 lattice grid [41], and the no-slip boundaries on particle surfaces are treated with DI-IBMs. Here we compared only two DI-IBM algorithms to save computational resources, the standard DI-IBM with two-sided force distribution, and the proposed DI-IBM with single-sided force distribution. In each simulation, the force-amplification technique has been applied with two iterations. Other information on the implementation details of the sim-

ulations has been reported in our recent publication [17] and thus is not repeated here for conciseness.

Rather than contrasting specific flow fields, it is more meaningful to examine the flow and particle statistics in turbulent particle-laden flows. In this validation case, we focus on comparing the flow and particle statistics result from DF-IBMs with the benchmark results obtained from the identical LBM code, but using the interpolated bounce-back (IBB) schemes [42–44] for no-slip boundary condition treatment on particle surfaces. IBB schemes are kinetic no-slip boundary treatments in LBM that can ensure second-order accuracy in fluid flow simulations. In our previous study [16,17], it has been systematically demonstrated that results from IBB schemes are generally more accurate than those from DF-IBMs, which justifies their use as benchmark results. Simply served for validation purposes, the physical interpretations of the simulation results will not be detailed, we focus only on contrasting the numerical performances.

The time-dependent turbulent kinetic energy (TKE) and dissipation rate are shown in Fig. 17. While results of TKE from different simulations match each other well, the dissipation rates with the standard DI-IBM have been underestimated by 5% at the 512^3 resolution, and 2%–3% at the 1024^3 resolution, compared to the IBB benchmark result, especially at the early time $t/T^0_e \leqslant 1$. The dissipation rate in a turbulent flow is usually defined as $\epsilon = 2\nu \langle s'_{ij} s'_{ij} \rangle$, where s'_{ij} is the strain rate of the turbulent velocity fluctuation, $\langle \cdots \rangle$ represents statistical averaging. A high level of errors in the dissipation rate indicates inaccurate computation of the velocity gradient. Considering that the total volume of the diffused layers in the fluid phase is only up to $V_{DI} = 4\pi N_p [(r + 2h)^3 - r^3]/3$, which equals about 4.2% and 1.8% of the total volume of the fluid phase with 512^3 and 1024^3 grid resolutions, these deviations are actually not negligible.

It is worth emphasizing that the results of TKE and dissipation rate shown in Fig. 17 are computed in a way that velocities in the regions occupied by particles are masked with velocity fields generated from particle rigid-body
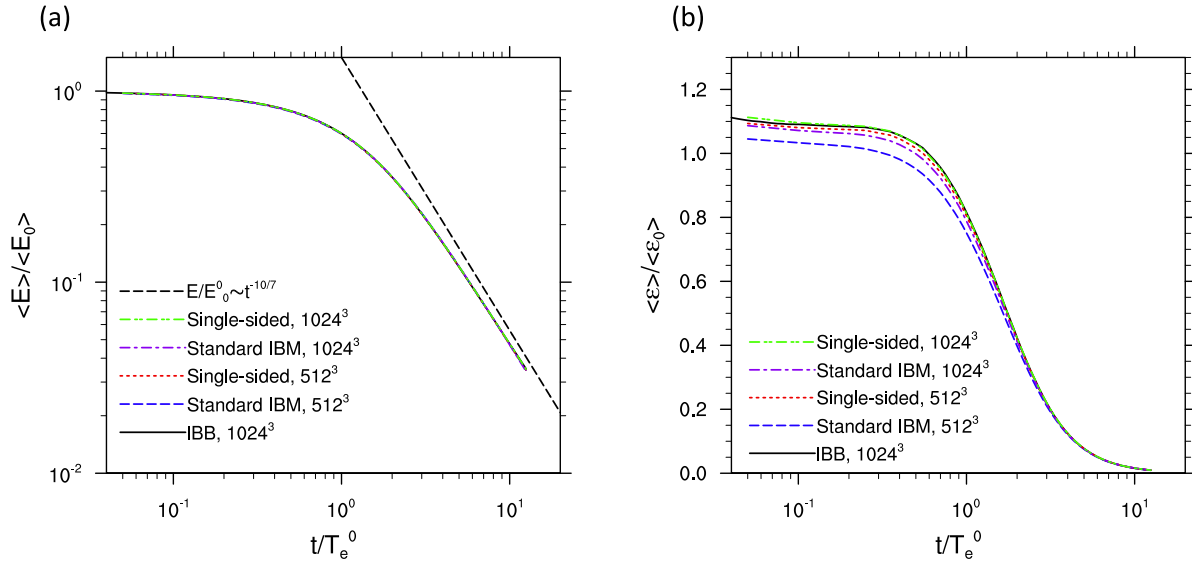
FIG. 17. Decaying TKE (a) and dissipation rate (b) in a particle-laden decaying homogeneous isotropic turbulence.

rotation. This masking eliminates the contribution of the virtual flows in particles to the TKE and dissipation rate results. For the proposed DI-IBM with single-sided force distribution kernel, the virtual flows in particles are more intensively perturbed since the boundary force is distributed single-sidedly. The masking turns out to be a necessity for the proposed DI-IBM, otherwise the virtual flow in the solid phase would contaminate the accuracy of the dissipation rate computation.

Next, the conditionally averaged TKE and dissipation rate as functions of the distance from the particle surface are shown in Fig. 18. The results obtained from the proposed DI-IBM with single-sided force distribution kernel always match better with the benchmark results compared to these from the standard DI-IBM. The most obvious improvement brought by the single-sided kernel is the correct calculation of dissipation rate close to the particle surface. The standard DI-IBM with

two-sided force distribution yields a decreased dissipation rate towards the particle surface, which is physically unlikely to happen. On the contrary, the proposed DI-IBM successfully captures the monotonically increasing dissipation rate towards the particle surface. This further confirms that the proposed single-sided kernel is able to restore correct velocity gradient computation in the vicinity of solid boundaries. This property has not been achieved by other DI-IBM algorithms.

Finally, we show the statistics of decaying particle kinetic energy in Fig. 19. Similar to what we have observed before, although the single-sided force distribution strategy could cause more severe fluctuations in the instantaneous hydrodynamic force and torque of individual particles, the statistical average of particle kinetic energy is not affected. The result of particle kinetic energy with the proposed DI-IBM match better with the IBB results than its counterpart with ths standard DI-IBM. The successful application of the proposed DI-IBM in
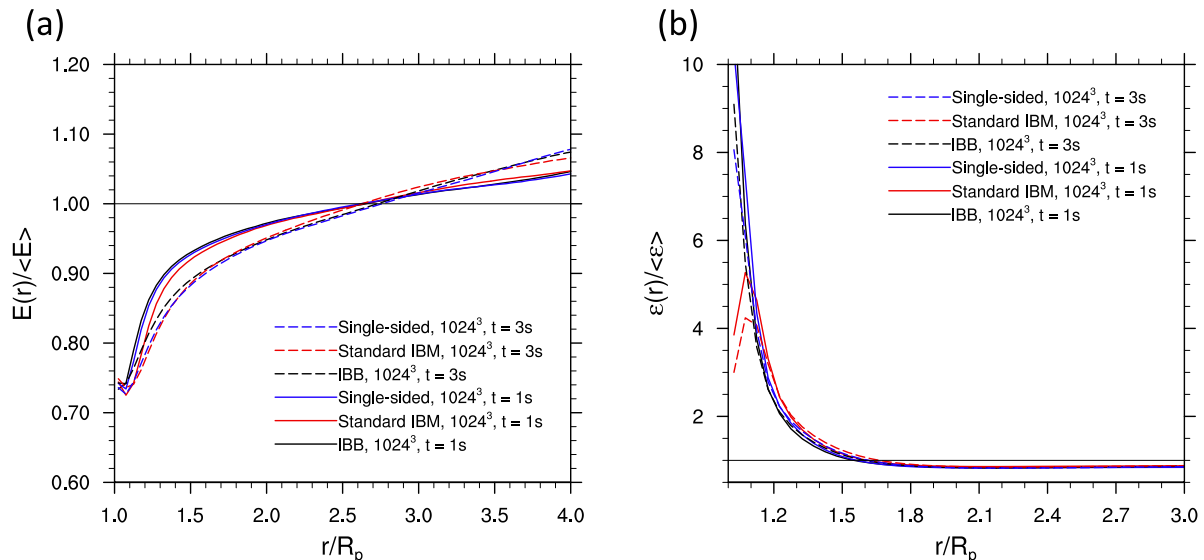

FIG. 18. Conditionally averaged TKE (a) and dissipation rate (b) as functions of distance from the particle surface.
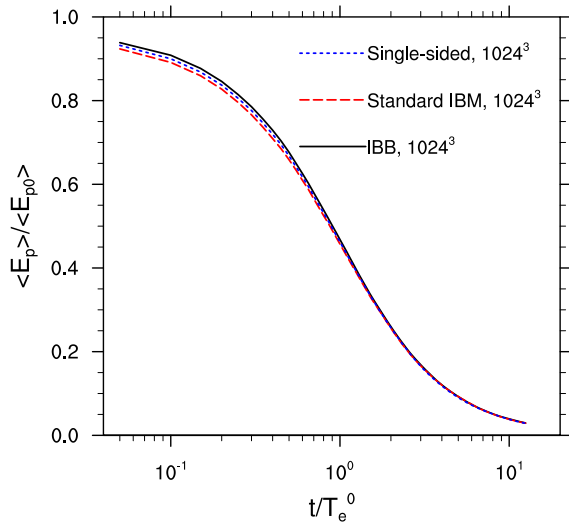
FIG. 19. Decaying particle kinetic energy.

a turbulent particle-laden flow also proves that the proposed DI-IBM still possesses sufficient numerical stability for high Reynolds number flows.

V. CONCLUSION AND REMARKS

In this paper, we analytically demonstrated that the standard DI-IBM with two-sided force distribution kernel could not ensure correct velocity gradient computation within the diffused layers of solid boundaries. A few attempts in the literature to address this issue were also shown to be ineffective, at least for particle-laden flow applications. To solve the problem, a single-sided force distribution kernel is proposed, where the boundary force is applied only to the Eulerian grid points locating within the solid region. In order to improve the accuracy of no-slip boundary enforcement in DF-IBM, an extremely simple force amplification technique was also

introduced, which can significantly reduces the number of required iterations for a given accuracy of no-slip boundary enforcement. This proposed force-amplified, single-sided DI-IBM was then validated in a series of laminar and turbulent flows.

Compared to standard DI-IBM, the proposed DI-IBM not only restores the correct velocity gradient in the vicinity of solid boundaries, but also improves the accuracy of other simulation results, such as the fluid velocity and hydrodynamic force and torque. The potential issues brought by the proposed DI-IBM, such as the lack of local torque conservation during the interpolation of boundary force from the Lagrangian to Eulerian grid, and the more severe fluctuation in the results of instantaneous hydrodynamic force and torque on a moving solid object, have also been carefully assessed. We did not find any evidence that those issues would damage the overall accuracy of the simulation results, thus the use of the proposed DI-IBM in practice should not be affected.

Finally, while the flow solver used in this study is the lattice Boltzmann method, the above conclusions should also hold for other flow solvers. We encourage readers familiar with other flow solvers to test the single-sided force distribution kernel and the force amplification technique.

[1] R. Mittal and G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. **37**, 239 (2005).

[2] C. S. Peskin, The immersed boundary method, Acta Numerica **11**, 479 (2002).

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

[4] W. Ren, C. Shu, J. Wu, and W. Yang, Boundary condition-enforced immersed boundary method for thermal flow problems with Dirichlet temperature condition and its applications, Comput. Fluids **57**, 40 (2012).

[5] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, and A. Von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comput. Phys. **227**, 4825 (2008).

[6] Y. Gorsse, A. Iollo, H. Telib, and L. Weynans, A simple second order cartesian scheme for compressible Euler flows, J. Comput. Phys. **231**, 7780 (2012).

[7] C. Liu and C. Hu, An immersed boundary solver for inviscid compressible flows, Int. J. Numer. Methods Fluids **85**, 619 (2017).

[8] M. R. Maxey, Simulation methods for particulate flows and concentrated suspensions, Annu. Rev. Fluid Mech. **49**, 171 (2017).

[9] W.-P. Breugem, A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows, J. Comput. Phys. **231**, 4469 (2012).

[10] T. Kempe and J. Fröhlich, An improved immersed boundary method with direct forcing for the simulation of particle laden flows, J. Comput. Phys. **231**, 3663 (2012).

[11] M.-C. Lai and Z. Li, A remark on jump conditions for the three-dimensional Navier-Stokes equations involving an immersed moving membrane, Appl. Math. Lett. **14**, 149 (2001).

[12] M. Vanella and E. Balaras, A moving-least-squares reconstruction for embedded-boundary formulations, J. Comput. Phys. **228**, 6617 (2009).

[13] A. Coclite, M. D. de Tullio, G. Pascazio, and P. Decuzzi, A combined lattice Boltzmann and immersed boundary approach for

predicting the vascular transport of differently shaped particles, Comput. Fluids **136**, 260 (2016).

[14] M. D. de Tullio and G. Pascazio, A moving-least-squares immersed boundary method for simulating the fluid–structure interaction of elastic bodies with arbitrary thickness, J. Comput. Phys. **325**, 201 (2016).

[15] J. C. Brändle de Motta, P. Costa, J. J. Derksen, C. Peng, L.-P. Wang, W.-P. Breugem, J. L. Estivalezes, S. Vincent, E. Climent, P. Fede *et al.*, Assessment of numerical methods for fully resolved simulations of particle-laden turbulent flows, Comput. Fluids **179**, 1 (2019).

[16] C. Peng, O. Ayala, and L.-P. Wang, A comparative study of immersed boundary method and interpolated bounce-back scheme for no-slip boundary treatment in the lattice Boltzmann method: Part I, laminar flows, Comput. Fluids **192**, 104233 (2019).

[17] C. Peng, O. Ayala, J. C. Brändle de Motta, and L.-P. Wang, A comparative study of immersed boundary method and interpolated bounce-back scheme for no-slip boundary treatment in the lattice Boltzmann method: Part II, turbulent flows, Comput. Fluids **192**, 104251 (2019).

[18] H. A. Williams, L. J. Fauci, and D. P. Gaver III, Evaluation of interfacial fluid dynamical stresses using the immersed boundary method, Discrete Contin. Dyn. Syst. Series B **11**, 519 (2009).

[19] C. Santarelli, J. Roussel, and J. Fröhlich, Budget analysis of the turbulent kinetic energy for bubbly flow in a vertical channel, Chem. Eng. Sci. **141**, 46 (2016).

[20] A. Ten Cate, J. J. Derksen, L. M. Portela, and H. E. A. Van Den Akker, Fully resolved simulations of colliding monodisperse spheres in forced isotropic turbulence, J. Fluid Mech. **519**, 233 (2004).

[21] M. Uhlmann, Interface-resolved direct numerical simulation of vertical particulate channel flow in the turbulent regime, Phys. Fluids **20**, 053305 (2008).

[22] F. Lucci, A. Ferrante, and S. Elghobashi, Modulation of isotropic turbulence by particles of taylor length-scale size, J. Fluid Mech. **650**, 5 (2010).

[23] F. Picano, W.-P. Breugem, and L. Brandt, Turbulent channel flow of dense suspensions of neutrally buoyant spheres, J. Fluid Mech. **764**, 463 (2015).

[24] A. Eshghinejadfard, A. Abdelsamie, S. A. Hosseini, and D. Thevenin, Immersed boundary lattice Boltzmann simulation of turbulent channel flows in the presence of spherical particles, Int. J. Multiphase Flow **96**, 161 (2017).

[25] S. Balachandar and J. Eaton, Turbulent dispersed multiphase flow, Annu. Rev. Fluid Mech. **42**, 111 (2010).

[26] K. Luo, Z. Wang, J. Fan, and K. Cen, Full-scale solutions to particle-laden flows: Multidirect forcing and immersed boundary method, Phys. Rev. E **76**, 066709 (2007).

[27] W.-X. Huang and F.-B. Tian, Recent trends and progress in the immersed boundary method, Proc. Inst. Mech. Eng., Part C **233**, 7617 (2019).

[28] M. Cisse, H. Homann, and J. Bec, Slipping motion of large neutrally buoyant particles in turbulence, J. Fluid Mech. **735**, R1 (2013).

[29] G. J. Rubinstein, J. Derksen, and S. Sundaresan, Lattice Boltzmann simulations of low-Reynolds-number flow past fluidized spheres: Effect of stokes number on drag force, J. Fluid Mech. **788**, 576 (2016).

[30] P. Costa, F. Picano, L. Brandt, and W.-P. Breugem, Universal scaling laws for dense particle suspensions in turbulent wall-bounded flows, Phys. Rev. Lett. **117**, 134501 (2016).

[31] M. N. Ardekani, P. Costa, W.-P. Breugem, F. Picano, and L. Brandt, Drag reduction in turbulent channel flow laden with finite-size oblate spheroids, J. Fluid Mech. **816**, 43 (2017).

[32] T. T. Bringley and C. S. Peskin, Validation of a simple method for representing spheres and slender bodies in an immersed boundary method for Stokes flow on an unbounded domain, J. Comput. Phys. **227**, 5397 (2008).

[33] C. Ji, A. Munjiza, and J. Williams, A novel iterative direct-forcing immersed boundary method and its finite volume applications, J. Comput. Phys. **231**, 1797 (2012).

[34] A. M. Roma, C. S. Peskin, and M. J. Berger, An adaptive version of the immersed boundary method, J. Comput. Phys. **153**, 509 (1999).

[35] J. Wu and C. Shu, Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications, J. Comput. Phys. **228**, 1963 (2009).

[36] Y. Zhang, G. Pan, Y. Zhang, and S. Haeri, A relaxed multi-direct-forcing immersed boundary-cascaded lattice Boltzmann method accelerated on GPU, Comput. Phys. Commun. **248**, 106980 (2020).

[37] S. Tao, Q. He, L. Wang, S. Huang, and B. Chen, A non-iterative direct-forcing immersed boundary method for thermal discrete unified gas kinetic scheme with Dirichlet boundary conditions, Int. J. Heat Mass Transf. **137**, 476 (2019).

[38] S. Chen and G. D. Doolen, Lattice Boltzmann method for fluid flows, Annu. Rev. Fluid Mech. **30**, 329 (1998).

[39] H. H. Hu, N. A. Patankar, and M. Zhu, Direct numerical simulations of fluid–solid systems using the arbitrary Lagrangian–Eulerian technique, J. Comput. Phys. **169**, 427 (2001).

[40] Z.-G. Feng and E. E. Michaelides, Robust treatment of no-slip boundary condition and velocity updating for the lattice-Boltzmann simulation of particulate flows, Comput. Fluids **38**, 370 (2009).

[41] D. d'Humières, Multiple–relaxation–time lattice Boltzmann models in three dimensions, Philos. Trans. R. Soc. London A **360**, 437 (2002).

[42] M. Bouzidi, M. Firdaouss, and P. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, Phys. Fluids **13**, 3452 (2001).

[43] D. Yu, R. Mei, L.-S. Luo, and W. Shyy, Viscous flow computations with the method of lattice Boltzmann equation, Prog. Aerospace Sci. **39**, 329 (2003).

[44] W. Zhao and W.-A. Yong, Single-node second-order boundary schemes for the lattice Boltzmann method, J. Comput. Phys. **329**, 1 (2017).