Computers and Mathematics with Applications 72 (2016) 349–374

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Implementation issues and benchmarking of lattice Boltzmann method for moving rigid particle simulations in a viscous flow

Cheng Peng [a], Yihua Teng [b], Brian Hwang [a], Zhaoli Guo [c], Lian-Ping Wang [a,c,*]

[a] Department of Mechanical Engineering, 126 Spencer Laboratory, University of Delaware, Newark, DE 19716-3140, USA

[b] Department of Energy and Resource Engineering, Peking University, Beijing, PR China

[c] National Laboratory of Coal Combustion, Huazhong University of Science and Technology, Wuhan, PR China

ARTICLE INFO

Article history:
Available online 1 October 2015

Keywords:
Lattice Boltzmann method
Interpolated bounce back
Hydrodynamic force
Galilean invariance
Refilling

ABSTRACT

In this work, we revisit implementation issues in the lattice Boltzmann method (LBM) concerning moving rigid solid particles suspended a viscous fluid. Three aspects relevant to the interaction between flow of a viscous fluid and moving solid boundaries are considered. First, the popular interpolated bounce back scheme is examined both theoretically and numerically. It is important to recognize that even though significant efforts had previously been devoted to the performance, especially the accuracy, of different interpolated bounce back schemes for a fixed boundary, there were relatively few studies focusing on moving solid surfaces. In this study, different interpolated bounce back schemes are compared theoretically for a moving boundary. Then, several benchmark cases are presented to show their actual performance in numerical simulations. Second, we examine different implementations of the momentum exchange method to calculate hydrodynamic force and torque acting on a moving surface. The momentum exchange method is well established for fixed solid boundaries, however, for moving solid boundaries there are still open issues such as unphysical force fluctuations and Galilean invariance errors. Recent progress in this direction is discussed, along with our own interpretations and modifications. Several benchmark cases, including a particle-laden turbulent channel flow, are used to demonstrate the effects of different modifications on the accuracy and physical results under different physical configurations. The third aspect is the refilling scheme for constructing the unknown distribution functions for the new fluid nodes that emerge from the previous solid region as a particle moves relative to a fixed lattice grid. We examine and compare the performance of the refilling schemes introduced by Fang et al. (2002), Lallemand and Luo (2003), and Caiazzo (2008). We demonstrate that improvements can be made to suppress force fluctuations resulting from refilling.

1. Introduction

Turbulent flows laden with solid particles are ubiquitous in engineering, biological and environmental applications. Examples include fluidized bed reactors, spray atomization, bubble columns, plankton contact dynamics in ocean water,

* Corresponding author at: Department of Mechanical Engineering, 126 Spencer Laboratory, University of Delaware, Newark, DE 19716-3140, USA.
E-mail addresses: cpengxpp@udel.edu (C. Peng), yhteng@pku.edu.cn (Y. Teng), bhwang@udel.edu (B. Hwang), zlguo@hust.edu.cn (Z. Guo), lwang@udel.edu (L.-P. Wang).

http://dx.doi.org/10.1016/j.camwa.2015.08.027
0898-1221/

transport of blood corpuscles in the human body, sediment transport, warm rain process, volcanic ash eruptions, dust storms, and sea sprays. In these applications, particles are usually suspended in a turbulent carrier fluid. The interactions between the dispersed and the carrier fluid phases impact the dynamics of suspended particles (e.g., dispersion, deposition rate, collision rate, settling velocity) and the bulk properties of the multiphase flow (e.g., wall or surface drag, turbulence intensity). In some of these applications, the particle size is comparable to or larger than the flow Kolmogorov length [1], which introduces a finite-size effect greatly complicating the description of the flow system. Currently, the only rigorous method is to numerically resolve the disturbance flows around particles, known as the particle-resolved simulation (PRS). This requires an explicit implementation of the no-slip boundary condition on the surface of each moving particle.

PRS of turbulent particle-laden flows requires direct simulation of the turbulent carrier flow and explicit and accurate treatment of many moving fluid–solid interfaces, such that all scales from turbulence integral scale to dissipation scales and particle size are adequately resolved with realistic scale separations that depend on applications. In recent years, several PRS methods based on the Navier–Stokes (N–S) equation have been developed, with the particle–fluid interfaces treated by the immersed boundary method [2,3], direct-forcing [4], local analytical treatment [5], overset grid [1], force-coupling [6], or penalization method [7]. As reviewed in [8,9], these studies have contributed to the understanding of flow modulation by the inertial particles and the dynamic effects due to finite particle size.

As an alternative approach, lattice Boltzmann method (LBM) has also been applied as a PRS method for turbulent particle-laden flows [10,8,9]. The LBM approach features a high-level data locality essential to efficient implementation of PRS. Another advantage is that LBM has the flexibility and simplicity (i.e., via local bounce-back) for implementing interfacial boundary conditions. This offers the potential for the method to be applied to treat turbulent flows laden with non-spherical and deformable particles.

In LBM, a set of mesoscopic distribution functions are solved. The number of microscopic velocities at a give lattice node is usually several times larger than the number of macroscopic hydrodynamic variables in the continuum N–S equation. This feature provides LBM with a much simpler evolution equation and greater flexibilities, but leads to an implementation issue at boundaries or fluid–solid interfaces. Although the macroscopic boundary conditions (i.e., no penetration and no-slip) are clearly established, the method to construct the missing microscopic distribution functions is no unique since the number of unknowns is larger than the number of boundary conditions.

For particle-laden flow simulations, this results in three general issues. First, the unknown distribution functions must be carefully constructed to satisfy the no-penetration and no-slip boundary conditions at the moving fluid–solid interfaces and other considerations of physical consistency and numerical accuracy. Second, to simulate the motion of moving solids in the fluid, hydrodynamic forces need to be accurately calculated from the microscopic distribution functions. Third, for flows with moving fluid–solid interfaces, every time when a previous solid node becomes a fluid node, the information at such a new fluid node needs to be filled.

Over the past 25 years, many efforts have been devoted to these three topics for moving fluid–particle problems. While many different implementations of the velocity boundary condition have been developed for a fixed straight boundary (see the review by Latt et al. [11]), for a moving curved boundary the early efficient implementation may be traced to studies of Ladd and co-workers [12,13]. In those early studies, the standard (or mid-link) bounce back scheme was used, causing a curved boundary to be effectively approximated by a zigzag staircase. Later on, different interpolated (and sometime extrapolated) bounce back schemes are proposed to capture more precisely the real fluid–solid interface [14–21]. Even though each boundary treatment scheme has been separately tested and applied to different physical problems, to our knowledge, they are yet to be systematically compared and benchmarked under same conditions. For the users of LBM, it is still not clear which boundary-condition implementation scheme to choose, especially when the solid boundaries are moving in a nonuniform flow.

Regarding hydrodynamic force evaluation in LBM, the most popular and efficient approach is the momentum exchange method (MEM), whose general concept has been introduced in the early studies [12,18,22]. Recently, the Galilean variance property of MEM has been questioned and remedies are proposed in several studies [23–26]. This problem was largely ignored in early works because of three reasons. First, benchmark cases used to test a force evaluation model were mostly limited to fixed solid boundaries where the Galilean invariance is not an issue. Second, while the Galilean invariance errors resulting from MEM may be present at every boundary node, for simple cases with symmetric solid particle shapes, it is usually thought that the local errors can cancel with each other such that the calculated net hydrodynamic force acting on a moving particle remains accurate. Third, because the local Galilean invariance error typically has a magnitude of $\mathcal{O}\left(Ma^2\right)$, where Ma is the Mach number, it is normally thought to be negligible. In light of the very recent developments [24,25], we will point out some incorrect suggestions made in the literature in attempt to restore Galilean invariance, as well as the subtle difference in the recent correct treatments by Wen et al. [25] and Chen et al. [26]. We will demonstrate that a lack of Galilean invariance could lead to unphysical results in more general situations (*e.g.*, particle-laden turbulent flows). Therefore, careful and thorough tests are still required to re-examine previous MEM models when applied to complex flow configurations. The recent studies [25,26] have demonstrated the deficiencies of previous MEM implementations, using simple to moderately complex moving-particle problems.

The refilling scheme for a new fluid node has been considered in several previous studies [27–31]. Lallemand and Luo [27] suggested a scheme based on quadratic extrapolation along the solid local outer normal direction. Fang et al. [28] averaged

the information from all neighboring fluid nodes as an approximation to the unknown distribution functions at the new fluid nodes, known as averaged extrapolation refilling scheme [28]. Caiazzo [29] partitioned the unknown distribution functions into the equilibrium and non-equilibrium parts, and obtained each part separately by direct calculation and propagation. In all the above schemes, the non-equilibrium parts of the unknown distribution functions are obtained from extrapolation. Different from them, Shin et al. [30] constructed the non-equilibrium parts as well from the local strain rate. Later, based on the same idea, Krithivasan et al. [31] used a more general Grad approximation to refill the information at the newly uncovered node points. Because the refilling process only happens locally around the particle surfaces, it might have only limited effects on the overall results of LBM simulation, especially for viscous flows with low Reynolds number. However, the effects of a refilling scheme on the calculated local flow and the hydrodynamic force acting on a solid particle have not been systemically studied using an adequate range of benchmark cases. From some preliminary results in a turbulent flow (*e.g.,* Fig. 8 in [9]), the noisy results of the local physics are likely related to the flaws in the refilling scheme. We will demonstrate that, if the refilling scheme is not properly implemented, it can be a source of significant problems affecting both the accuracy of the force evaluation and the numerical stability of fluid–particle systems, especially when the flow and particle Reynolds numbers are high.

This study has two general objectives. First, by reviewing and critically re-examining previous studies especially those recent ones, we attempt to delineate implementation issues in LBM simulations with moving fluid–solid interfaces and their potential impacts on the physical results. Second, by comparing different methods using a wide range of benchmark cases, we attempt to develop a few general guidelines for choosing optimized implementations in moving particle simulations. All three aspects mentioned above will be considered. We shall emphasize that these aspects are inter-connected, and it is important to treat all of them correctly in order to achieve a successful and accurate simulation of a moving particle system.

This paper shall be arranged as follows. Section 2 provides a brief description of the lattice Boltzmann approach. Different interpolated bounce back schemes for moving fluid–solid boundary are reviewed, tested, and compared in Section 3. In Section 4, we focus our discussions on different MEM implementations to compute hydrodynamic forces acting on a moving solid particle. Recent developments and improvements are particularly stressed. In Section 5, different filling schemes are discussed and compared, and a new velocity-constrained filling scheme is introduced. A summary and outlook are presented in Section 6. Based on the criteria of the least unphysical fluctuations in the resulting hydrodynamic force and the best numerical stability, we conclude that the velocity-constrained filling scheme provides the best overall performance when compared to other filling schemes.

2. Lattice Boltzmann method

The lattice Boltzmann (LB) equation is a fully discrete approximation in velocity, time and space, of continuous Boltzmann equation. Different from the conventional CFD methods based on the N–S equation, in the LB approach we solve the kinetic equation for the molecular distribution function $f_\alpha(\mathbf{x}, t)$ at position $\mathbf{x}$, time t and with molecular velocity $\mathbf{e}_\alpha$. A typical LB equation with the single-relaxation-time or BGK collision term is written as

$$f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \delta t, t + \delta t) - f_\alpha(\mathbf{x}, t) = -\frac{1}{\tau}\left[f_\alpha(\mathbf{x}, t) - f_\alpha^{(eq)}(\mathbf{x}, t)\right], \tag{1}$$

where δt is the time step size, τ is the dimensionless relaxation time and is related to the kinematic viscosity $\nu = (\tau - 0.5)c_s^2\delta t$, c_s is the model speed of sound. The number of the discrete velocities ($\mathbf{e}_\alpha$) depends the model details and dimensionality of the physical space. The equilibrium distribution function $f_\alpha^{(eq)}$ is typically expressed as

$$f_\alpha^{(eq)} = \rho w_\alpha \left[1 + \frac{1}{c_s^2}(\mathbf{e}_\alpha \cdot \mathbf{u}) + \frac{1}{2c_s^4}(\mathbf{e}_\alpha \cdot \mathbf{u})^2 - \frac{1}{2c_s^2}\mathbf{u}^2\right], \tag{2}$$

where w_α is the weighting factor, ρ is fluid density, and $\mathbf{u}$ is the hydrodynamic velocity. Further details can be founded in [32–34]. The above LB equation can be divided into two processes

$$\text{The collision step: } \tilde{f}_\alpha(\mathbf{x}, t) = f_\alpha(\mathbf{x}, t) - \frac{1}{\tau}\left[f_\alpha(\mathbf{x}, t) - f_\alpha^{(eq)}(\mathbf{x}, t)\right] \tag{3}$$

$$\text{The streaming step: } f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \delta t, t + \delta t) = \tilde{f}_\alpha(\mathbf{x}, t). \tag{4}$$

With proper lattice symmetry and equilibrium distributions, the LB equation can be designed to reproduce the N–S equation under the multiscale Chapman–Enskog expansion [35]. Multiple designs of the lattice structure, lattice velocities, and equilibrium distribution are possible. The most popular lattice models are the D2Q9 model (nine velocities in two space dimensions) and the D3Q19 model (nineteen velocities in three space dimensions), as shown in Fig. 1. The moments of f_α

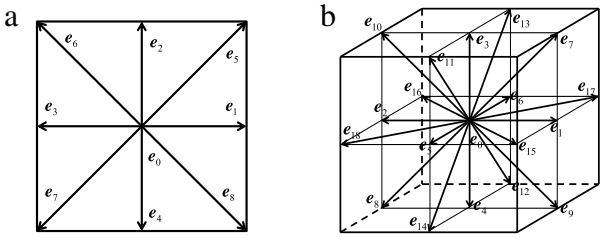

Fig. 1. Lattice structure and lattice velocities in (a) D2Q9 and (b) D3Q19.

provide the macroscopic quantities, namely, the density, velocity and stress tensor can be obtained as

$$\begin{aligned}
&\rho = \sum_{\alpha} f_{\alpha} \\
&\rho \mathbf{u} = \sum_{\alpha} f_{\alpha} \mathbf{e}_{\alpha} \\
&p\delta_{ij} + \rho u_i u_j = \sum_{\alpha} f_{\alpha}^{(eq)} e_{\alpha i} e_{\alpha j} \\
&\Pi_{ij} = -\left(1 - \frac{1}{2\tau}\right) \sum_{\alpha} \left(f_{\alpha} - f_{\alpha}^{(eq)}\right) e_{\alpha i} e_{\alpha j}
\end{aligned} \tag{5}$$

where $e_{\alpha i}$ denotes the i-component of the microscopic velocity $\mathbf{e}_{\alpha}$, p is the pressure, and $\Pi_{ij} = \rho\nu\left(\partial u_i/\partial x_j + \partial u_j/\partial x_i\right)$ is the viscous stress tensor.

2.1. Multi-relaxation time (MRT) LBM

Given a set of discrete velocities $\mathbf{e}_{\alpha}$ and their corresponding distribution functions f_{α}, in principle one can construct the same number of independent moments of f. These moments can be linked to macroscopic modes such as density, momentum, stress, energy, energy flux, *etc*. In kinetic theory it is well known that various physical processes in fluids can be approximately described by the coupling and interactions among these modes [36]. Unlike the LBGK model given by Eq. (1), different modes may be associated with different relaxation processes (or transport coefficients), thus relaxation in the moment space provides more flexibility. For the D2Q9 model, the 9 independent moments in the moment space are

$$|m\rangle = \left(\rho, e, \varepsilon, j_x, q_x, j_y, q_y, p_{xx}, p_{xy}\right)^T, \tag{6}$$

where ρ is the density, e is related to the kinetic energy, ε is related to the kinetic energy squared, j_x and j_y are the x and y components of the momentum, q_x and q_y are proportional to the energy flux in x and y directions, p_{xx} and p_{xy} are related to the diagonal and off-diagonal components of the viscous stress tensor. The distribution functions and moments are related through a transform matrix $\mathbf{M}$

$$|m\rangle = \mathbf{M}|f\rangle, \qquad |f\rangle = \mathbf{M}^{-1}|m\rangle \tag{7}$$

where $|f\rangle = (f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8)^T$.

The density and moment moments are conserved variables during collision. The other six non-conserved moments could relax at different rates, leading to the Multiple-relaxation-time (MRT) LBM as

$$f_{\alpha}(\mathbf{x} + \mathbf{e}_{\alpha}, t + \delta t) - f_{\alpha}(\mathbf{x}, t) = -\mathbf{M}^{-1}\mathbf{S}\mathbf{M}\left[f_{\alpha}(\mathbf{x}, t) - f_{\alpha}^{(eq)}(\mathbf{x}, t)\right], \tag{8}$$

where $\mathbf{S}$ is a diagonal relaxation matrix $\mathbf{S} = \text{diag}\,(0, s_1, s_2, 0, s_4, 0, s_4, s_{\nu}, s_{\nu})$. Specifically, s_{ν} is related to the kinematic viscosity as $\nu = (s_{\nu}^{-1} - 1)c_s^2\delta t$, s_1 is related to the bulk viscosity, s_2 and s_4 are adjustable parameters that have no effects on the N–S equation. At low Mach number, s_1, s_2, and s_4 may be used to improve numerical stability [36,37] and accuracy of the boundary condition implementation [19,38]. Therefore, a main benefit of the MRT model over LBGK model is its better numerical stability, especially for flows with high Reynolds numbers [38,37].

It should be noted that in MRT LBM, the equilibrium distribution functions $f_{\alpha}^{(eq)}$ may not be identical to its counterparts in LBGK, as the equilibrium distributions for some moments may be irrelevant to the N–S equation. As mentioned in [36], the equilibrium of the energy squared moment in D2Q9 MRT model has a form of $\epsilon^{(eq)} = a\rho + b(j_x^2 + j_y^2)$, with a and b being free parameters. By choosing $a = 1$ and $b = -3$, the standard $f_{\alpha}^{(eq)}$ given in Eq. (2) can be recovered. Furthermore, by modifying the configuration of the moment space and related relaxation processes, the lattice geometry of the LB approach can be expanded. For example, using MRT LBM, a correct N–S hydrodynamic model can be constructed even on a rectangular lattice grid, as shown in [39]. In this study, all results from different benchmark cases are obtained by using the MRT LBM. We note that the LBGK model is a special case of MRT LBM where all relaxation parameters are equal to $1/\tau$. More details can be found in [36] for 2D MRT model and [37] for 3D MRT models.

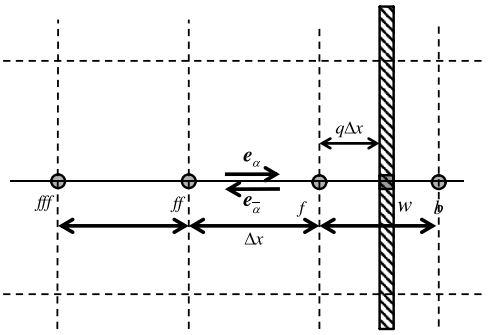

Fig. 2. Sketch to show notations used to construct a bounce back scheme at the fluid–solid boundary.

2.2. Nearly incompressible formulation of the lattice Boltzmann equation

For all benchmark cases to be developed in this paper, since the Mach number Ma is assumed to be low, we take the suggestion made in [40] to use the nearly incompressible formulation of the lattice Boltzmann equation. In the incompressible formulation, it is assumed the local density fluctuation is small compared to the mean density and we write

$$\rho = \rho_0 + \delta\rho, \tag{9}$$

where the density fluctuation $\delta\rho$ has the magnitude of $\mathcal{O}\left(Ma^2\right)$. By substituting the above relation into the equilibrium distribution, Eq. (2), and neglecting terms of $\mathcal{O}\left(Ma^3\right)$ or higher, the equilibrium distribution becomes

$$f_\alpha^{(eq)} = w_\alpha \left\{ \delta\rho + \rho_0 \left[\frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} \right] \right\}. \tag{10}$$

Though the Chapman–Enskog procedure, the LBGK model yields the following hydrodynamic equations

$$\frac{\partial \delta\rho}{\partial t} + \nabla \cdot \mathbf{u} = 0 \tag{11a}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho_0} \nabla p + \nu \nabla^2 \mathbf{u} \tag{11b}$$

where $p = \delta\rho c_s^2$, is the pressure.

It is obvious that Eq. (11a) will enforce incompressibility for steady flows. For transient flows, the incompressible formulation also has less compressibility effects than the regular model [40]. For example, the incompressible formulation can significantly reduce the intensity of numerical pressure waves and improve the precision of transient simulations with respect to incompressibility and predicted pressure [41]. Therefore, in this paper, all benchmark cases will be developed using this incompressible formulation. The same density decomposition can be applied within MRT LBM to achieve the same benefits.

3. Boundary treatments in the solid–fluid interface

In the macroscopic fluid dynamics, the no-penetration no-slip boundary condition is usually assumed at the fluid–solid interface. Since the number of unknown mesoscopic distribution functions at the fluid–solid interface is larger than the number of macroscopic boundary conditions, the implementation of the boundary condition in terms of the distribution functions is not unique and numerous possibilities have been developed over the years [12,14–19,21,27,42–44], all with the goal to fill in the missing distribution functions streaming from solid nodes. Among all these different implementations, the bounce back scheme is the most popular one due to its physical simplicity and robustness. On the fluid–solid boundary, the bounce back scheme can be written as:

$$f_{\bar{\alpha}}\left(\mathbf{x}_f, t+\delta t\right) = \tilde{f}_\alpha\left(\mathbf{x}_f, t\right) - \left(\tilde{f}_\alpha^{(eq)}\left(\mathbf{x}_w, t^+\right) - f_{\bar{\alpha}}^{(eq)}\left(\mathbf{x}_w, t^+\right)\right) = \tilde{f}_\alpha\left(\mathbf{x}_f, t\right) + 2w_\alpha\rho_0 \frac{\mathbf{e}_{\bar{\alpha}} \cdot \mathbf{u}_\mathbf{w}}{c_s^2}, \tag{12}$$

where x_f indicates the boundary fluid node as shown in Fig. 2, $\mathbf{e}_\alpha$ points into the solid region and $\mathbf{e}_{\bar{\alpha}} = -\mathbf{e}_\alpha$, t^+ denotes the time when the distribution $\tilde{f}_\alpha\left(\mathbf{x}_f, t\right)$ arrives at the fluid–solid boundary. Basically, it is assumed that the non-equilibrium part of the distribution on a boundary link remains unchanged during bounce back.

The simple bounce back scheme, Eq. (12), is known to be of second-order accuracy only when the solid surface is located half lattice spacing from the boundary node f. When the solid surface is not at the mid-link position, its accuracy is of first order only which is inconsistent with the deign of the LB equation. Or equivalently, the curved surface is approximated by zig-zag staircases [12]. To capture the actual shape of the solid surface and maintain the second-order accuracy, an interpolation scheme is used to construct the missing distribution function $f_{\bar{\alpha}}\left(\mathbf{x}_f, t+\delta t\right)$. Some of the known interpolation schemes are discussed next, followed by a systematic comparison using a few benchmark problems.

3.1. Interpolated bounce back schemes

3.1.1. Bouzidi's interpolated bounce back scheme [14]

The scheme of Bouzidi et al. [14] considers two different situations based on the relative location of the boundary node $\mathbf{x}_f$ relative to the solid boundary $\mathbf{x}_w$, using the parameter $q = |\mathbf{x}_f - \mathbf{x}_w|/|\mathbf{x}_f - \mathbf{x}_b|$ (see Fig. 2). For $q \leq 0.5$, the distributed is interpolated at the temporary location D between $\mathbf{x}_f$ and $\mathbf{x}_{ff}$ by the distributions at $\mathbf{x}_f$ and $\mathbf{x}_{ff}$ (and $\mathbf{x}_{fff}$ for a quadratic interpolation), such that the interpolated population at D will stream exactly to $\mathbf{x}_f$ after bounce back during one time step. For $q \geq 0.5$, the population at $\mathbf{x}_f$ is first streamed and bounced back to a temporary location D' located between $\mathbf{x}_w$ and $\mathbf{x}_f$, then together with the population at $\mathbf{x}_{ff}$ (and $\mathbf{x}_{fff}$ for a quadratic interpolation) to obtain the unknown population at $\mathbf{x}_f$ again by interpolation. For $q \leq 0.5$, the interpolation expressions are

$$\text{Linear}: f_{\bar{\alpha}}\left(\mathbf{x}_f, t+\delta t\right) = 2q\tilde{f}_{\alpha}\left(\mathbf{x}_f, t\right) + (1-2q)\,\tilde{f}_{\alpha}\left(\mathbf{x}_{ff}, t\right) + 2w_{\alpha}\rho_0\frac{\mathbf{e}_{\bar{\alpha}}\cdot\mathbf{u}_{\mathbf{w}}}{c_s^2} \tag{13}$$

$$\begin{aligned}\text{Quadratic}: f_{\bar{\alpha}}\left(\mathbf{x}_f, t+\delta t\right) &= q\,(2q+1)\,\tilde{f}_{\alpha}\left(\mathbf{x}_f, t\right) + (1+2q)\,(1-2q)\,\tilde{f}_{\alpha}\left(\mathbf{x}_{ff}, t\right)\\ &\quad - q\,(1-2q)\,\tilde{f}_{\alpha}\left(\mathbf{x}_{fff}, t\right) + 2w_{\alpha}\rho_0\frac{\mathbf{e}_{\bar{\alpha}}\cdot\mathbf{u}_{\mathbf{w}}}{c_s^2}\end{aligned} \tag{14}$$

and for $q \geq 0.5$:

$$\text{Linear}: f_{\bar{\alpha}}\left(\mathbf{x}_f, t+\delta t\right) = \frac{1}{2q}\left(\tilde{f}_{\alpha}\left(\mathbf{x}_f, t\right) + 2w_{\alpha}\rho_0\frac{\mathbf{e}_{\bar{\alpha}}\cdot\mathbf{u}_{\mathbf{w}}}{c_s^2}\right) + \frac{2q-1}{2q}f_{\bar{\alpha}}\left(\mathbf{x}_f, t\right) \tag{15}$$

$$\begin{aligned}\text{Quadratic}: f_{\bar{\alpha}}\left(\mathbf{x}_f, t+\delta t\right) &= \frac{1}{q\,(2q+1)}\left(\tilde{f}_{\alpha}\left(\mathbf{x}_f, t\right) + 2w_{\alpha}\rho_0\frac{\mathbf{e}_{\bar{\alpha}}\cdot\mathbf{u}_{\mathbf{w}}}{c_s^2}\right)\\ &\quad + \frac{2q-1}{q}f_{\bar{\alpha}}\left(\mathbf{x}_f, t\right) - \frac{2q-1}{1+2q}f_{\bar{\alpha}}\left(\mathbf{x}_{ff}, t\right).\end{aligned} \tag{16}$$

3.1.2. Mei's interpolated bounce back scheme [15–17]

This interpolation scheme was originally proposed by Filippova and Hänel [15] and later improved by Mei et al. [16,17] for better numerical stability. The interpolation is expressed as:

$$f_{\bar{\alpha}}\left(\mathbf{x_f}, t+\delta t\right) = (1-\chi)\,\tilde{f}_{\alpha}\left(\mathbf{x}_f, t\right) + \chi f_b^{*}\left(\mathbf{x_b}, t\right) + 2w_{\alpha}\rho_0\frac{\mathbf{e}_{\bar{\alpha}}\cdot\mathbf{u}_{\mathbf{w}}}{c_s^2} \tag{17}$$

where $f_b^{(*)}\left(\mathbf{x_b}, t\right)$ is the fictitious equilibrium distribution:

$$f_b^{(*)}\left(\mathbf{x}_b, t\right) = w_{\alpha}\left\{\delta\rho\left(\mathbf{x}_f, t\right) + \rho_0\left[\frac{1}{c_s^2}\mathbf{e}_{\alpha}\cdot\mathbf{u}_{bf} + \frac{1}{2c_s^4}\left(\mathbf{e}_{\alpha}\cdot\mathbf{u}_f\right)^2 - \frac{1}{2c_s^2}\mathbf{u}_f\cdot\mathbf{u}_f\right]\right\}. \tag{18}$$

This interpolation scheme also considers two cases, namely, the parameter of χ and $\mathbf{u}_{bf}$ are given as

$$\text{for } q < 0.5: \ \mathbf{u}_{bf} = \mathbf{u}_f, \qquad \chi = \frac{2q-1}{\tau-1} \tag{19}$$

$$\text{for } q \geq 0.5: \ \mathbf{u}_{bf} = \frac{q-1}{q}\mathbf{u}_f + \frac{1}{q}\mathbf{u}_w, \qquad \chi = \frac{2q-1}{\tau}. \tag{20}$$

Later, through numerical tests, it was noticed that the scheme becomes unstable when the absolute value of χ is large. According to Eq. (19), there is a large unstable region around $\tau = 1$ and stretching all the way close to $\tau = 0.5$ when $q < 0.2$ [16]. Therefore, this scheme can only have good performance for large values of τ. For high *Re* number flows, where the low viscosity is usually achieved by having small τ, the original scheme is inadequate. To improve the numerical stability, Mei et al. [16] suggested that χ and $\mathbf{u}_{bf}$ can be modified as:

$$\begin{aligned}&\text{for } q < 0.5: \ \mathbf{u}_{bf} = \mathbf{u}_{ff}, \qquad \chi = \frac{2q-1}{\tau-2}\\ &\text{for } q \geq 0.5: \ \mathbf{u}_{bf} = \frac{q-1}{q}\mathbf{u}_f + \frac{1}{q}\mathbf{u}_w, \qquad \chi = \frac{2q-1}{\tau}\end{aligned} \tag{21}$$

where the unstable region is moved to τ close to 2.

For higher Reynolds number flows, Mei et al. [17] further suggested the following modification

$$\begin{aligned}&\text{for } q < 0.5: \ \mathbf{u}_{bf} = \mathbf{u}_{ff}, \qquad \chi = \frac{2q-1}{\tau-2}\\ &\text{for } q \geq 0.5: \ \mathbf{u}_{bf} = \left(1-\frac{3}{2q}\right)\mathbf{u}_f + \frac{3}{2q}\mathbf{u}_w, \qquad \chi = \frac{2q-1}{\tau+0.5}.\end{aligned} \tag{22}$$

Again, this modification originates from the consideration that for high Re flow, when τ is only slightly greater than 0.5, the value of χ could exceed the stability region when q is close to 1 [17]. In the followed test, unless explicitly specified, Mei's scheme implies the scheme presented by Eq. (22).

3.1.3. Yu's double interpolation [18]

The two bounce-back interpolation schemes introduced above need to treat $q < 0.5$ and $q \geq 0.5$ separately. In the double interpolation treatment originated by Yu et al. [18], all values of q are handled with a same expression. Three steps are included in the realization of this boundary treatment. First, by using the existing populations at $f_\alpha\left(\mathbf{x}_f, t\right)$ and $f_\alpha\left(\mathbf{x}_{ff}, t\right)$ (and $f_\alpha\left(\mathbf{x}_{fff}, t\right)$), the population at a temporary location is interpolated, which, during the streaming process, will propagate exactly to the wall

$$\text{Linear}: \tilde{f}_\alpha\left(\mathbf{x}_w, t+\delta t\right) = q\tilde{f}_\alpha\left(\mathbf{x}_f, t\right) + (1-q)\,\tilde{f}_\alpha\left(\mathbf{x}_{ff}, t\right) \tag{23}$$

$$\text{Quadratic}: f_\alpha\left(\mathbf{x}_w, t+\delta t\right) = \frac{q\,(1+q)}{2}\tilde{f}_\alpha\left(\mathbf{x}_f, t\right) + (1-q)\,(1+q)\,\tilde{f}_\alpha\left(\mathbf{x}_{ff}, t\right) - \frac{q\,(1-q)}{2}\tilde{f}_\alpha\left(\mathbf{x}_{fff}, t\right). \tag{24}$$

Second, an instantaneous bounce-back operation at the wall yields

$$f_{\bar{\alpha}}\left(\mathbf{x}_w, t+\delta t\right) = f_\alpha\left(\mathbf{x}_w, t+\delta t\right) + 2w_\alpha\rho_0\frac{\mathbf{e}_{\bar{\alpha}}\cdot\mathbf{u}_\mathbf{w}}{c_s^2}. \tag{25}$$

Finally, the unknown population is interpolated from $f_{\bar{\alpha}}\left(\mathbf{x}_w, t+\delta t\right)$ and $f_{\bar{\alpha}}\left(\mathbf{x}_{ff}, t+\delta t\right)$ $\left(\text{and } f_{\bar{\alpha}}\left(\mathbf{x}_{fff}, t+\delta t\right)\right)$ as

$$\text{Linear}: f_{\bar{\alpha}}\left(\mathbf{x}_f, t+\delta t\right) = \frac{1}{1+q}f_{\bar{\alpha}}\left(\mathbf{x}_w, t+\delta t\right) + \frac{q}{1+q}f_{\bar{\alpha}}\left(\mathbf{x}_{ff}, t+\delta t\right) \tag{26}$$

$$\begin{aligned}\text{Quadratic}: f_{\bar{\alpha}}\left(\mathbf{x}_f, t+\delta t\right) \;=\; & \frac{2}{(1+q)\,(2+q)}f_{\bar{\alpha}}\left(\mathbf{x}_w, t+\delta t\right) + \frac{2q}{1+q}f_{\bar{\alpha}}\left(\mathbf{x}_{ff}, t+\delta t\right) \\ & - \frac{q}{2+q}f_{\bar{\alpha}}\left(\mathbf{x}_{fff}, t+\delta t\right).\end{aligned} \tag{27}$$

3.1.4. Ginzburg's multireflection [19]

While the above bounce-back schemes are claimed to be of second-order accuracy, the relative errors of those boundary schemes are viscosity-dependent [19]. Ginzburg [19] proposed a viscosity-independent bounce back scheme known as multireflection boundary condition. This boundary scheme is derived using the Chapman–Enskog expansion at planar boundary and involves 6 populations at the fluid boundary node and their neighboring nodes. The general mathematical form is written as

$$\begin{aligned}f_{\bar{\alpha}}\left(\mathbf{x}_f, t+\delta t\right) \;=\; & \kappa_1\tilde{f}_\alpha\left(\mathbf{x}_f, t\right) + \kappa_0\tilde{f}_\alpha\left(\mathbf{x}_{ff}, t\right) + \kappa_{-1}\tilde{f}_\alpha\left(\mathbf{x}_{fff}, t\right) \\ & + \bar{\kappa}_{-1}\tilde{f}_{\bar{\alpha}}\left(\mathbf{x}_f, t\right) + \bar{\kappa}_{-2}\tilde{f}_{\bar{\alpha}}\left(\mathbf{x}_{ff}, t\right) + w_q w_\alpha\rho_0\frac{\mathbf{e}_{\bar{\alpha}}\cdot\mathbf{u}_\mathbf{w}}{c_s^2} + w_\alpha\frac{F_\alpha^{pc}}{c_s^2}.\end{aligned} \tag{28}$$

All coefficients κ_i on the RHS are functions of q, which are given in TABLE II in [19]. The last term is proportional to the second-order non-equilibrium part of the distribution so it will be purposely ignored for the sake of simplicity. By specifying the values of those coefficients, the simple bounce back, Eq. (12), and Bouzidi's interpolated bounce back schemes, Eqs. (13)–(16) can be recovered [19].

3.1.5. Chun's equilibrium interpolation [21]

For flows with dense particle suspension, the gap between two particles might be too small that no sufficient fluid nodes could be used for interpolation. To address this problem, Chun and Ladd proposed an equilibrium interpolation [21]. The idea is originated from the observation that, according to the Chapman–Enskog expansion, the non-equilibrium part of a distribution function is always one order smaller than the equilibrium part, which was also noted in Guo et al. [20]. Hence, an interpolated bounce back scheme for the equilibrium part combined with a simple bounce back for the non-equilibrium part can still give the second-order accuracy. This scheme also needs to be divided into two situations according to q, when $q \leq 0.5$, the scheme is similar to Bouzidi's linear interpolation, Eq. (13), but only interpolate for the equilibrium part:

$$f_{\bar{\alpha}}^{(eq)}\left(\mathbf{x}_f, t+\delta t\right) = 2qf_\alpha^{(eq)}\left(\mathbf{x}_f, t\right) + (1-2q)\,f_\alpha^{(eq)}\left(\mathbf{x}_{ff}, t\right) + 2w_\alpha\rho_0\frac{\mathbf{e}_{\bar{\alpha}}\cdot\mathbf{u}_\mathbf{w}}{c_s^2}. \tag{29}$$

One the other hand, when $q > 0.5$, the fictitious equilibrium distribution at the wall point $\mathbf{x}_w$ (see Fig. 2) can be defined based on the local density ρ_w and wall velocity $\mathbf{x}_w$. Then, an interpolation based on the equilibrium distributions at $\mathbf{x}_w$ and

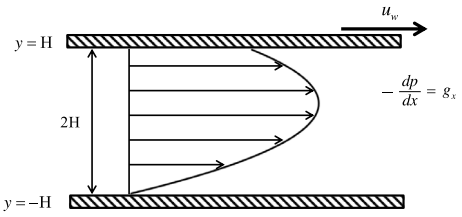

Fig. 3. The 2D Poiseuille flow with a moving top wall.

$\mathbf{x}_f$ can be done before streaming happens. By streaming, the bounced back population at the boundary node $\mathbf{x}_f$ is obtained. The process can be expressed as:

$$f_{\bar{\alpha}}^{(eq)}\left(\mathbf{x}_f, t\right) = \frac{1-q}{q} f_{\alpha}^{(eq)}\left(\mathbf{x}_f, t\right) + \frac{2q-1}{q} f_{\alpha}^{(eq)}\left(\mathbf{x}_w\right) \tag{30}$$

where $f_{\alpha}^{(eq)}\left(\mathbf{x}_w\right) = f_{\alpha}^{(eq)}\left(\delta\rho_w, \mathbf{u}_w\right)$. The density here can be either local fluid density or mean fluid density. When there is only one fluid node between two solid surfaces, in the case of $q < 0.5$, one can construct equilibrium distribution at both surface to conduct the interpolation:

$$f_{\bar{\alpha}}^{(eq)}(\mathbf{x}_f, t+\delta t) = \frac{q'+2q-1}{q'} f_{\alpha}^{(eq)}(\mathbf{x}'_w) + \frac{2q-1}{q'} f_{\alpha}^{(eq)}(\mathbf{x}_w) \tag{31}$$

where $f_{\alpha}^{(eq)}(\mathbf{x}_w)$ and $f_{\alpha}^{(eq)}(\mathbf{x}'_w)$ are the equilibrium distributions at the wall surface points $\mathbf{x}_w$ and $\mathbf{x}'_w$, respectively. q is the relative distance from the boundary node point $\mathbf{x}_f$ to $\mathbf{x}_w$, while q' is the relative distance between $\mathbf{x}_f$ and $\mathbf{x}'_w$. Note that the interpolation only happens when $q' > 1-2q$, otherwise one should use the simple bounce back, Eq. (12), instead.

3.2. Benchmark cases and comparison results

To maintain the overall accuracy of the LBM simulations, the boundary treatment schemes should at least be of second-order accuracy. At the same time, they should be numerically stable and robust to handle different physical situations, especially for flows with complex geometries or at high Reynolds number. For the implementations listed in Section 3.1, the accuracy and stability have been well established in each original publication and can be examined both theoretically and numerically. However, even though these boundary schemes are frequently used in different flows involving moving surfaces [45,46,26], their performances have not been compared adequately and systematically to allow others to chose one from the others. In this sections, these schemes are compared for three different benchmark cases: (1) Poiseuille channel flow with one wall moving; (2) a cylinder translating in a horizontal channel, and (3) an ellipsoidal particle falling freely in a vertical channel. In all the test cases discussed here, the Galilean invariance momentum exchange method (to be introduced in Section 4) by Wen et al. [25] is employed to obtain the hydrodynamic forces and torque acting on a solid particle. The velocity-constrained normal extrapolation scheme (to be introduced in Section 5) is used to refill the missing information at new fluid nodes.

3.2.1. Case problem 1 (CP1): 2D Poiseuille flow with a moving top wall

The physical problem is sketched in Fig. 3. The viscous flow is driven by both a moving top wall and a constant pressure gradient. The theoretical solution with zero initial velocity can be obtained as

$$u = -\frac{g_x}{2\nu}y^2 + \left(\frac{g_x}{2\nu}H + \frac{u_w}{H}\right)y + \sum_{n=0}^{\infty} A_n e^{-\frac{-n^2\pi^2}{H^2}\nu t} \sin\left(\frac{n\pi y}{H}\right) \tag{32}$$

where H is the width of the channel, $A_n = \frac{g_x}{\nu}\frac{H^2}{n^3\pi^3}\left[(-1)^n - 1\right] + \frac{2u_w}{n\pi}(-1)^n$. At steady state, the theoretical solution reduces to

$$u = -\frac{g_x}{2\nu}y^2 + \left(\frac{g_x}{2\nu}H + \frac{u_w}{H}\right)y. \tag{33}$$

First, the accuracy of different interpolate bounce back schemes are numerically examined for different values of q. We first consider two values of q (0.25 and 0.75), and examine the relative error of the numerical solution relative to the analytical solution at the steady state. The number of grid points N in the channel width direction varies from 30 to 120, which a step of 30. In *CP1*, the relaxation parameters being used are $\mathbf{S} = \text{diag}\,(0, 5/3, 1.54, 0, 1.9, 0, 1.9, 5/3, 5/3)$. In Fig. 4, we show the results of relative numerical errors, defined as $(u - u_{theo})/u_{theo}$ and averaged over the channel width, where u_{theo} is the theoretical results obtained by Eq. (33). The red lines show a slope of -2 in the log–log plots. It is confirmed that all boundary schemes can maintain a second-order accuracy even with the moving top boundary. However, the absolute value of errors

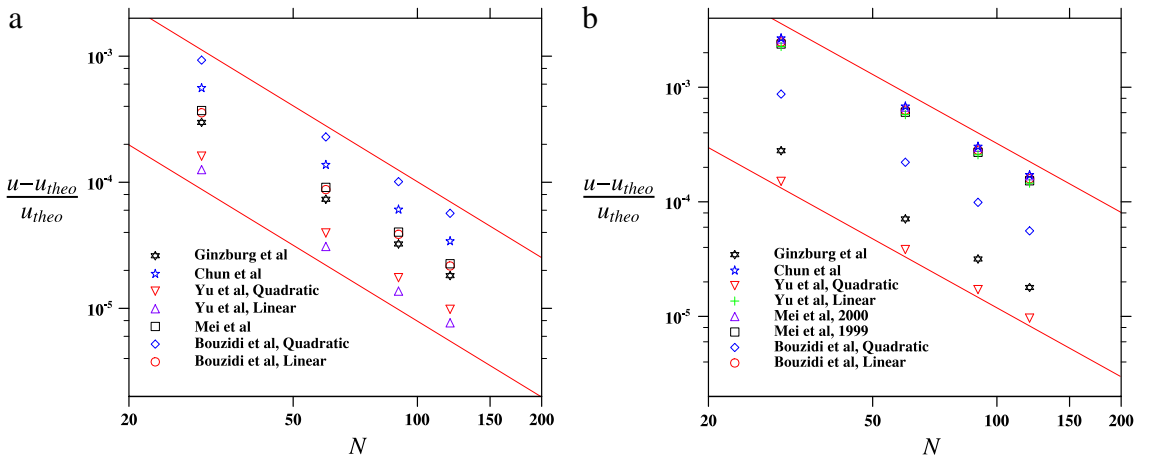

Fig. 4. The magnitude of relative error in velocity as a function of H in lattice units, (a): $q = 0.25$, Mei's method are identical in [16] and [17]; (b): $q = 0.75$, the results indicated as Mei et al., 1999 and Mei et al., 2000 are obtained by using the scheme in [16,17], respectively.

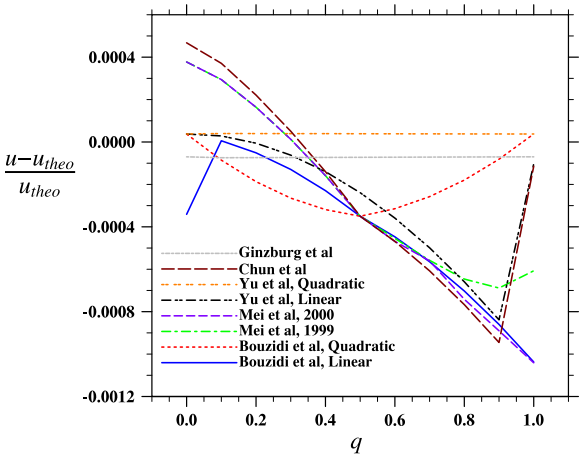

Fig. 5. The relative errors in velocity as a function of q.

can differ by up to one order of magnitude. It is interesting to observe that while the quadratic schemes are more accurate than the linear schemes for $q = 0.75$, the linear schemes perform better at $q = 0.25$.

The dependence of relative errors on q for each case is shown in Fig. 5. The value of q is varied from 0 to 1.0, with N fixed at 60. Several observations can be made from Figs. 4 and 5. The errors from the linear interpolation schemes (Chun, Mei, Bouzidi's linear and Yu's linear) not only change more dramatically with the value of q, but also have larger magnitudes than those from quadratic interpolation schemes (Bouzidi's quadratic, Yu's quadratic and Ginzburg). These results show that the quadratic schemes have better overall accuracy in the sense that the error is not strongly dependent on q. It is worth pointing out that this weaker dependence on q for the quadratic interpolation schemes could be an important advantage for flows with moving solid surfaces, where q is constantly changing with time. This could imply smaller fluctuations in the computed hydrodynamic quantities, especially the forces acting on solid particles (to be discussed later). Moreover, it is observed that none of the linear scheme has converged error at $q = 0$ and $q = 1$, while all quadratic schemes have identical error at those two extremes, indicating better consistency for quadratic schemes.

3.2.2. Case problem 2 (CP2): a cylinder translating with a constant velocity in 2D Couette flow

In this case, a cylinder is translating with a constant velocity in a horizontal Couette flow driven by two oppositely moving walls. The cylinder is located slightly away from the centerline of the channel and moves horizontally with a prescribed constant velocity and zero angular velocity (see Fig. 6). The parameters are $N_x = 201$, $N_y = 101$, $D = 25.25$, $a = 54.0$, $u_w = 0.1$, and $u_p = 0.02$. The kinematic viscosity is $1/9$.

We shall compare the resulting hydrodynamic forces acting on the cylindrical particle computed by the same momentum exchange method (the Galilean-invariant momentum exchange method [25]) and the same refilling scheme (velocity-constrained normal extrapolation scheme, see Section 5). The resulting hydrodynamic forces acting on the cylinder in x

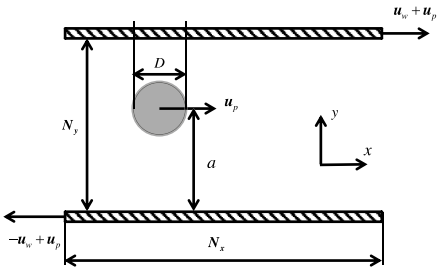

Fig. 6. Sketch showing a cylinder translating in a 2D Couette flow.

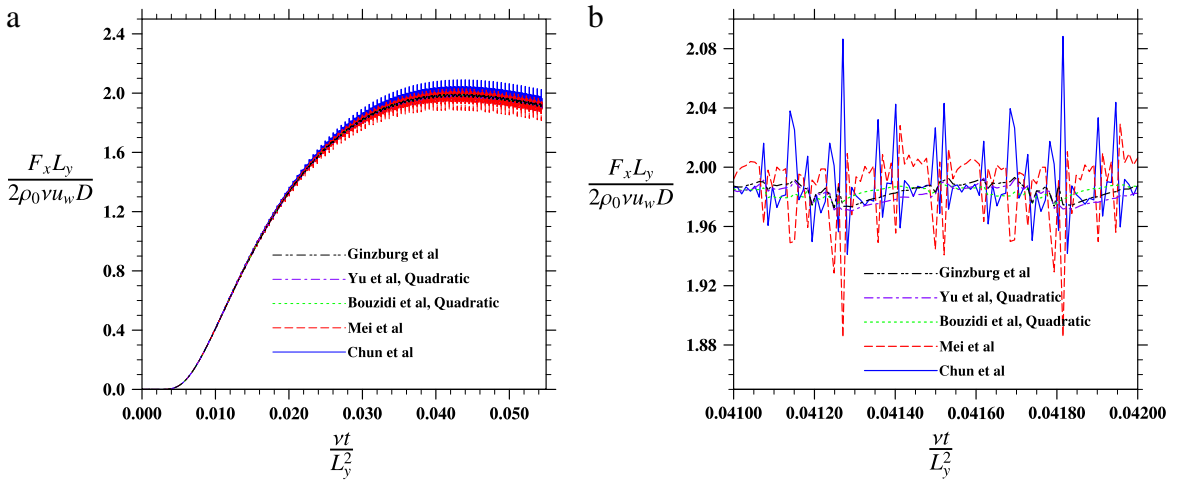

Fig. 7. Time evolution of the hydrodynamic force component F_x: (a) the whole time interval, (b) a zoom-in view.

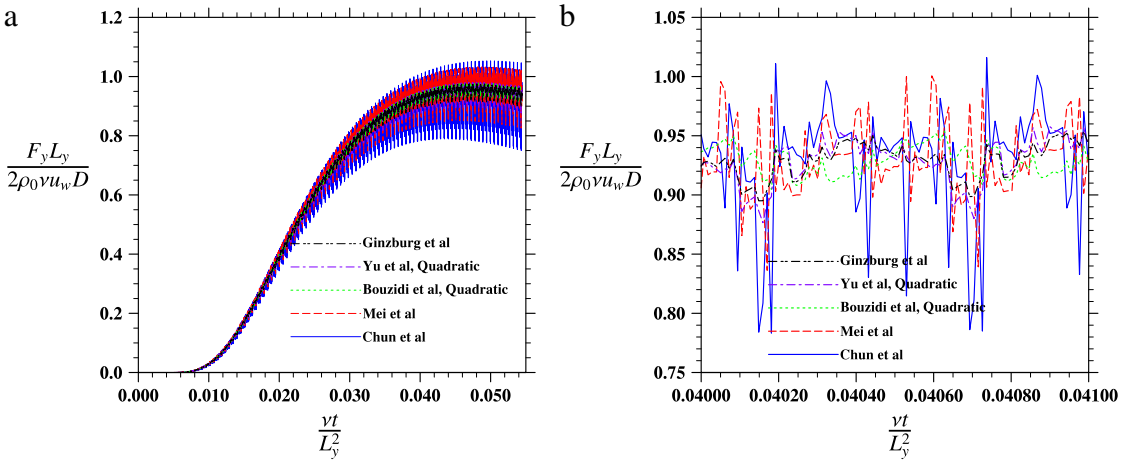

Fig. 8. Time evolution of the hydrodynamic force component F_y: (a) the whole time interval, (b) a zoom-in view.

and y directions are shown in Figs. 7 and 8, respectively. There are significant force fluctuations, with the fluctuation level depending on the interpolation scheme. According to Galilean invariance, the physical problem is unchanged when $\mathbf{u}_p$ is set to zero. In this case, the resulting force is smooth and shall be used as a benchmark to compute the level of fluctuations. For the case of $u_p = 0$, different schemes yield similar results (with relative difference less than 0.3%). When quantifying force fluctuations, we use the results of the respective scheme with u_p set to zero as the benchmark. The unphysical fluctuations, as discussed in [27], partly result from the fact that the relative position of the each boundary node is always changing as the cylinder moves in time. Therefore, carefully choosing the interpolation scheme can play an important role in reducing this fluctuation. As shown in Figs. 7 and 8, the fluctuations when the quadratic interpolation schemes (Bouzidi, Yu and Ginzburg) are used are generally much smaller than those based on the linear interpolation schemes (Mei and Chun). We computed

Table 1
The level of normalized force fluctuations for the case of cylinder translating in a 2D Couette flow: [a] $u_p = 0.02$; [b] $u_p = 0.03$.

Bounce back schemes	$\sigma_x^{[a]}$	$\sigma_y^{[a]}$	$\sigma_x^{[b]}$	$\sigma_y^{[b]}$
Chun et al. (linear)	1.3545×10^{-2}	5.8856×10^{-2}	1.4715×10^{-2}	7.1256×10^{-2}
Mei et al. (linear)	1.3412×10^{-2}	4.0017×10^{-2}	1.5267×10^{-2}	5.0011×10^{-2}
Bouzidi et al. (quadratic)	1.9575×10^{-3}	1.6082×10^{-2}	2.7440×10^{-3}	1.9372×10^{-2}
Yu et al. (quadratic)	3.0536×10^{-3}	2.2175×10^{-2}	4.0325×10^{-3}	2.5513×10^{-2}
Ginzburg et al. (quadratic)	2.9348×10^{-3}	1.7314×10^{-2}	4.0760×10^{-3}	2.1588×10^{-2}

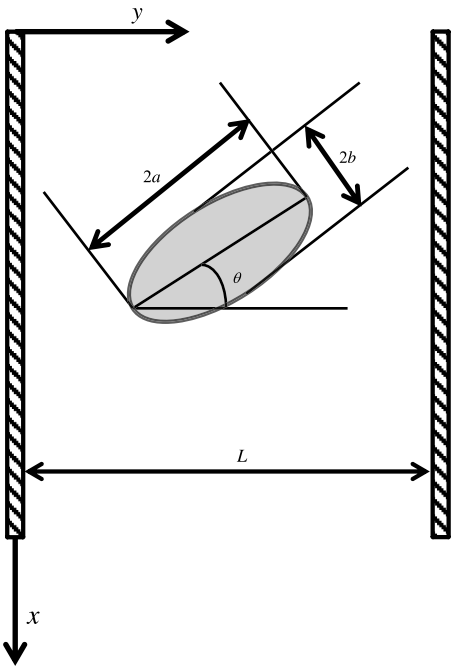

Fig. 9. The initial configuration of an elliptic particle settling freely in a vertical channel.

the standard deviations (normalized by the maximum F_x value for the fixed-cylinder benchmark case) in F_x and F_y for each moving particle simulation, using data from roughly the last 1/3 time interval. The results are compiled in Table 1, which demonstrates that, by choosing the higher-order interpolation schemes, the force fluctuation can be reduced by a factor of 5–7 in the x direction and a factor of 2–4 in the y direction. All quadratic schemes have a similar level of fluctuations, with Bouzidi's scheme being the best. The two linear interpolation schemes also yield similar results. When the particle velocity increases, the force fluctuations from all five interpolation schemes become larger, indicating that the force fluctuation will become a more severe issue for high-Re flows. Therefore, for the force fluctuations in moving particle simulations, we conclude that the quadratic (or higher-order) interpolation schemes are strongly preferred over linear interpolation schemes.

3.2.3. Case problem 3 (CP3): an elliptic particle settling in a 2D vertical channel

Here we consider an elliptic particle settling freely in a vertical channel. This is a more complex case as the particle shape is not rotationally symmetric. The results could depend more sensitively on the choice of boundary interpolation schemes as the errors result from different origins may not cancel each other. The results are compared with those based on an finite element method (FEM) in [45]. The FEM results are widely accepted as the benchmark for different numerical methods mainly for two reasons. First, because of the body fitted mesh used in the FEM, it represents the boundary geometry better than the numerical methods using a structured mesh. Second, the benchmark results come from arbitrary Lagrangian–Eulerian (ALE) method, in which the mesh was reconstructed as the solid boundary moved. This feature handles the moving boundary more accurately compared to the use of a fixed grid.

The physical problem is sketched in Fig. 9. An elliptic particle is released at the top region of the channel, and settles freely by the effect of gravity. The density ratio between particle and fluid is $\rho_p/\rho_f = 1.1$. Initially the particle is centered in the channel with an angle $\pi/8$ between its major axis and the horizontal direction, so the particle also rotates because of the unbalanced torque. The geometric parameters in this case are: $L_x \times L_y = 12$ cm $\times$ 0.4 cm, $a = 0.05$ cm and $b = 0.025$ cm. The fluid viscosity is $\nu = 0.01$ cm^2 s^{-1} and the gravity is $g_x = 980$ cm/s^2.

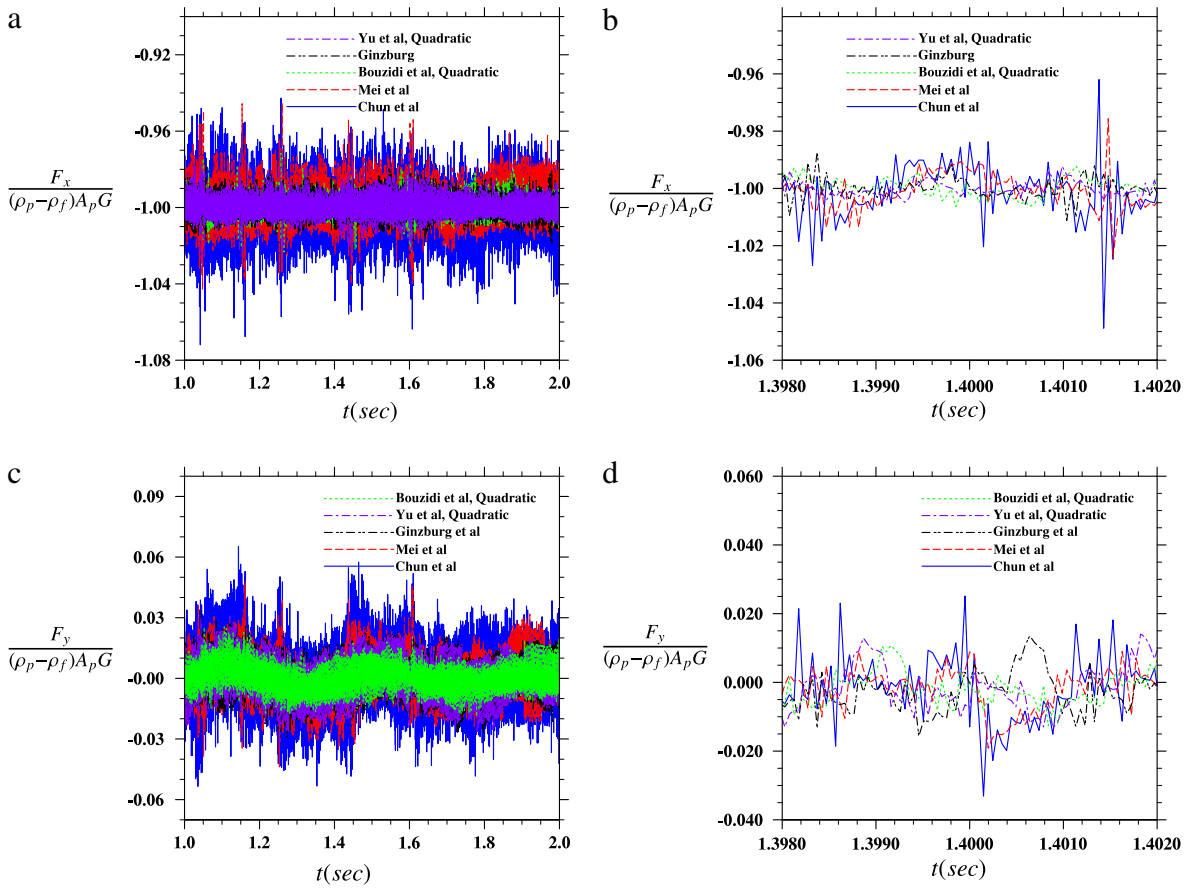

Fig. 10. The time dependent hydrodynamic force on of elliptic particle: (a) x direction; (b) x direction zoom-in; (c) y direction; (d) y direction zoom-in.

Table 2
The level of fluctuations in the calculated F_x and F_y normalized by $(\rho_p - \rho_f)A_p g_x$, for the elliptic particle setting freely in a 2D vertical channel.

Bounce back schemes	σ_x	σ_y
Chun et al.	1.0416×10^{-2}	1.0777×10^{-2}
Mei et al.	6.2602×10^{-3}	7.0853×10^{-3}
Bouzidi et al.	4.1732×10^{-3}	4.6196×10^{-3}
Yu et al.	3.5729×10^{-3}	6.8134×10^{-3}
Ginzburg et al.	3.8693×10^{-3}	7.1484×10^{-3}

We first examine the time dependent hydrodynamic forces acting on the elliptic particle in x and y directions, as shown in Fig. 10. For a better quantitative comparison, the standard deviation (normalized by $(\rho_p - \rho_f)A_pG$, where $A_p = \pi ab$) of time-independent force fluctuation is also calculated and shown in Table 2. In this case, because of the unsteadiness of particle motion, the benchmark value is defined as the averaged forces in every 1/400 second. Only the results after 1.5 s are considered in the calculation of r.m.s. fluctuations. Once again, we conclude that quadratic interpolation schemes can significantly reduce the level of fluctuations in the resulting forces acting on the particle. Bouzidi's scheme appears to be the best overall.

Furthermore, the effect of instantaneous force fluctuations may have a long-term effect on the particle trajectory, namely, they may not cancel out when integrated over time. The center trajectory and angular orientation of the particle are shown in Fig. 11 for each scheme. Indeed, at long times the trajectory obtained from the linear interpolation schemes departs from that based on the quadratic schemes. The quadratic schemes yield almost the same trajectory. Note that even for the time interval where FEM benchmark data exist, a close look indicates that the results from the quadratic schemes fit better to the FEM data. The above observation shows that, for complex flows, the choice of interpolation schemes not only affects the force evaluation, but also has an impact on the accuracy of particle velocities and trajectories. This implies that in a more complex flow such as a turbulent flow the simulated result of fluid–solid particle interaction could be more sensitive to the choice of the interpolation scheme. Especially, in a turbulent carrier flow, the frequencies of force fluctuations could overlap

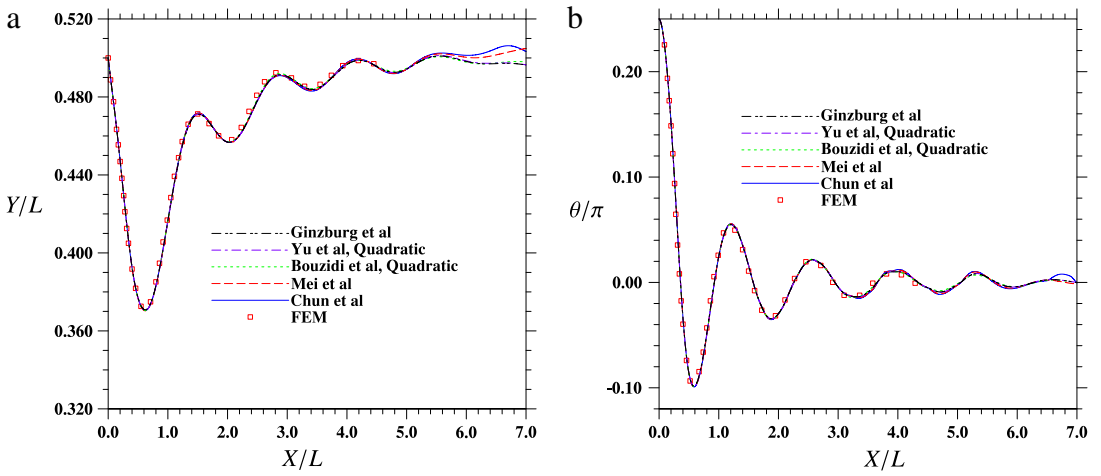

Fig. 11. The time evolution of (a) the trajectory and (b) orientation angle for the case of the sedimentation of elliptic particle.

with fast time scales in the flow, which can make the choices of interpolation schemes even more important. Also, from Table 1 we observe that the magnitude of force fluctuations increases with the solid particle velocity. For these two reasons, we recommend that quadratic interpolation schemes be used instead of linear schemes in a turbulent carrier flow.

4. Force calculation methods in lattice Boltzmann method

In the context of particle-resolved simulations of turbulent particle-laden flows [8,9], it is important to capture the motion of suspended solid particles accurately. This requires accurate and efficient computation of hydrodynamic force and torque acting on each solid particle. In LBM, there are two methods to calculate the hydrodynamic force. The first is the stress integration method (SIM), which, as in the N–S equation based CFD simulations, integrate the contribution of local pressure and stress on the solid surface. The second approach is known as the momentum exchange method (MEM), which simply sums up, over all boundary links, the local momentum change of the fluid lattice particles during the bounce back process at the fluid–solid interface. In this section, the two methods are briefly introduced following the previous studies [12,13,18,22,26,46,47]. The focus here is to review and compare different implementations of MEM in order to reveal the best approach. Specifically, the local Galilean invariance of MEM is stressed. All numerical simulations shown in this section utilize Bouzidi's boundary implementation and the velocity-constrained refilling, as we focus only on comparing different methods for evaluating hydrodynamic forces acting on solid particles.

4.1. Stress integration method

In SIM, the force and torque acting on the surface Ω of a solid particle is computed as

$$\mathbf{F} = \int_{\Omega} \left[-p\mathbf{I} + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)\right] \cdot \mathbf{n} dA, \tag{34}$$

$$\mathbf{T} = \int_{\Omega} \mathbf{r} \times \left\{\left[-p\mathbf{I} + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)\right] \cdot \mathbf{n}\right\} dA, \tag{35}$$

where $\mathbf{n}$ is the unit normal pointing into the fluid region on the fluid–solid boundary, $\mathbf{r}$ is the position vector relative to the particle center. Obviously SIM is based on the continuum hydrodynamics description. In LBM, the local pressure p can be easily evaluated using the equation of states $p = \delta\rho c_s^2$. The viscous stress can be calculated mesoscopically as

$$\tau_{ij} = \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right) = -\left(1 - \frac{1}{2\tau}\right)\sum_{\alpha}\left(f_{\alpha} - f_{\alpha}^{(eq)}\right)\mathbf{e}_{\alpha i}\mathbf{e}_{\alpha j}. \tag{36}$$

The integration must be performed on a set of points defined on the surface of the solid particle, where the distribution functions must be obtained through extrapolation from fluid nodes outside of the solid particle. This extrapolation process could be tedious and is not unique, making SIM computationally inefficient [22,46], particularly when many solid particles are present in a fluid–solid system. For this reason, we shall not use this approach in this study.

4.2. Momentum exchange method

The momentum exchange method, on the other hand, is inherited from the kinetic theory, the description at the microscopic level on which LBM is based. The fictitious fluid lattice particles act as molecules. To obtain the force acting on

the solid surface one can sum up momentum changes of fluid lattice particles with the fluid–solid boundary during bounce-back. In early studies, this is expressed as the net loss of fluid momentum during the steaming substep, relative to the *fixed* frame of reference, as

$$\mathbf{F} = \sum_{all\ \mathbf{x}_f} \sum_{\alpha_{bl}} \left[\tilde{f}_\alpha(\mathbf{x}_f, t) + f_{\bar{\alpha}}(\mathbf{x}_f, t + \delta t) \right] \mathbf{e}_\alpha. \tag{37}$$

The torque can be evaluated as:

$$\mathbf{T} = \sum_{all\ \mathbf{x}_f} \sum_{\alpha_{bl}} (\mathbf{x}_w - \mathbf{Y}_c) \times \left[\tilde{f}_\alpha(\mathbf{x}_f, t) + f_{\bar{\alpha}}(\mathbf{x}_f, t + \delta t) \right] \mathbf{e}_\alpha \tag{38}$$

where $\mathbf{Y}_c$ is the center position of the solid particle, $\mathbf{x}_w$ is the boundary point where the boundary link intercepts with the solid surface, the double summations are first over the boundary links α_{bl} pointing from a given boundary node $\mathbf{x}_f$ into the solid surface, then over all the boundary nodes. We shall refer to Eq. (37) as the conventional MEM or CMEM. CMEM is extremely easy to implement in LBM as all the information needed in the above equation is known as part of the implementation of the no-penetration and no-slip condition. For the force evaluation on a curved boundary, CMEM is thought to be superior to SIM for its simplicity and robustness [22]. However, CMEM as it is written, has a serious problem with regard to local Galilean invariance.

4.3. Local Galilean invariance of MEM implementations

By *local* Galilean invariance, we mean that the force on any local area of the fluid–particle interface remains the same if a constant velocity is added to the whole system. As pointed out previously [24,25], Eq. (37) is not Galilean invariant locally. As an illustration, consider a fully relaxed system (*i.e.*, no flow relative to the boundary) in a moving frame of reference with a constant velocity along x direction. In this case, the distribution function at each node point is equal to the equilibrium distribution. For the specific boundary arrangement shown in Fig. 12, the momentum exchange happened at point $\mathbf{x}_f$ according to Eq. (37) can be evaluated explicitly for a given boundary node $\mathbf{x}_f$, as

$$\mathbf{F}_{\mathbf{x}_f} = \sum_{\alpha=1,5,8} \mathbf{e}_\alpha \left(f_\alpha^{(eq)} + f_{\bar{\alpha}}^{(eq)} \right) = \sum_{\alpha=1,5,8} w_\alpha \mathbf{e}_\alpha \left[2\delta\rho + \rho_0 \frac{(\mathbf{e}_\alpha \cdot \mathbf{u}_w)^2}{c_s^4} - \rho_0 \frac{\mathbf{u}_w^2}{c_s^2} \right] \tag{39}$$

where the $\mathbf{u}_w = u_w \hat{\mathbf{x}}$ is the velocity of the moving reference surface, $\hat{\mathbf{x}}$ and is the unit vector in the x direction. This implies that the net hydrodynamic force acting on the *local* boundary depends on the value of u_w, thus it is not locally Galilean invariant. Under certain conditions, such as those with very low Mach number or those where a perfect symmetry exists for both solid particle geometry and flow pattern, the *integrated* hydrodynamic force acting on the solid particle may still appear to be Galilean invariant, because the unphysical error term is either negligible, or happens to cancel one another on two opposite sides of a solid particle surface. However, for more general situations, especially when the local flow field is nonuniform or turbulent, the above conditions are not met. A significant error may be present in the calculated hydrodynamic force due to this violation of local Galilean invariance. The source of this problem is rather simple, namely, the momentum exchange in Eq. (37) was based on the fixed reference frame, rather on the frame of reference moving with the local boundary.

To address this problem, a simple-minded approach is to subtract the Galilean invariance error in the original momentum exchange model on *each boundary link* [23,24].

$$\bar{\mathbf{F}}_\alpha = \mathbf{F}_\alpha - \mathbf{F}_\alpha^{err} = \mathbf{F}_\alpha - w_\alpha \mathbf{e}_\alpha \left[\rho_0 \frac{(\mathbf{e}_\alpha \cdot \mathbf{u}_w)^2}{c_s^4} - \rho_0 \frac{\mathbf{u}_w^2}{c_s^2} \right] \tag{40}$$

where $\bar{\mathbf{F}}_\alpha$ and $\mathbf{F}_\alpha$ are the momentum exchange after and before this specific Galilean invariance correction, $\mathbf{F}_\alpha^{err}$ is the Galilean invariance error obtained in Eq. (39). However, the linkwise correction has no physical basis, as the local Galilean invariance is based on the *summation over boundary links* on a local solid surface. Therefore, this linkwise correction is a stronger correction than what is necessary per local Galilean invariance of the hydrodynamic force.

The correct approach to restore local Galilean invariance is in fact rather straightforward and was introduced in the recent work by Wen et al. [25]. The similar idea was mentioned in the study of Krithivasan et al. [31]. In the kinetic theory, when a population of molecules hits the wall, the momentum exchange between the population and the wall is a function of the relative velocity so the net impulse is

$$\mathbf{F}dt = m_{in}(\mathbf{v}_{in} - \mathbf{u}_w) - m_{out}(\mathbf{v}_{out} - \mathbf{u}_w) \tag{41}$$

where m_{in} and m_{out} are the mass of the population before and after the population hits the wall, respectively. $\mathbf{v}_{out}$ and $\mathbf{v}_{in}$ indicate the velocity of the molecules after and before they collide with the wall. Obviously, the Galilean invariance issue results from the fact that the mass is not necessary conserved, *i.e.* m_{in} and m_{out} are not equal (here mainly due the

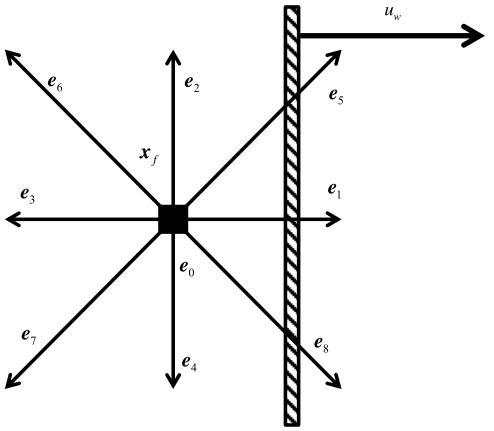

Fig. 12. The link contact of a fluid lattice node with a solid wall.

interpolation schemes being used). In LBM, the mass does not conserved when a population hits a curved wall. Therefore, by applying the same concept to MEM, we have

$$\begin{aligned}\mathbf{F} &= \sum_{all\ \mathbf{x}_f}\sum_{\alpha_{bl}}\left[\tilde{f}_\alpha\left(\mathbf{x}_f,t\right)\left(\mathbf{e}_\alpha-\mathbf{u}_w\right)-f_{\bar{\alpha}}\left(\mathbf{x}_f,t+\delta t\right)\left(\mathbf{e}_{\bar{\alpha}}-\mathbf{u}_w\right)\right],\\ \mathbf{T} &= \sum_{all\ \mathbf{x}_f}\sum_{\alpha_{bl}}\left(\mathbf{x}_w-\mathbf{Y}_c\right)\times\left[\tilde{f}_\alpha\left(\mathbf{x}_f,t\right)\left(\mathbf{e}_\alpha-\mathbf{u}_w\right)-f_{\bar{\alpha}}\left(\mathbf{x}_f,t+\delta t\right)\left(\mathbf{e}_{\bar{\alpha}}-\mathbf{u}_w\right)\right].\end{aligned} \tag{42}$$

For the spacial case shown in Fig. 12 with no relative fluid motion, it is easy to show that the net force calculated at $\mathbf{x}_f$ is

$$\mathbf{F}_{\mathbf{x}_b}=\sum_{\alpha=1,5,8}\left[\tilde{f}_\alpha^{(eq)}(\mathbf{e}_\alpha-\mathbf{u}_w)-f_{\bar{\alpha}}^{(eq)}(\mathbf{e}_{\bar{\alpha}}-\mathbf{u}_w)\right]=\sum_{\alpha=1,5,8}2\delta\rho w_\alpha\mathbf{e}_\alpha \tag{43}$$

which is independent of u_w and thus is *locally* Galilean invariant. Compared with the linkwise correction, this updated Galilean-invariant momentum exchange method (GIMEM) is based on a solid physical ground. The Galilean invariance error is eliminated on the level of node points (rather than on a single boundary link), which is fully consistent with the hydrodynamics requirement.

For the spacial case shown in Fig. 12, we can compare the corrections to Eq. (37) in the linkwise correction method and in the GIMEM. For links 1, 5, 8, the corrections in GIMEM are $(-2\rho_0u_w^2/3,\ 0)$, $(-\rho_0u_w^2/6,\ 0)$, and $(-\rho_0u_w^2/6,\ 0)$, respectively. But in the linkwise method, they are $(-2\rho_0u_w^2/3,\ 0)$, $(-\rho_0u_w^2/6,\ -\rho_0u_w^2/6)$, and $(-\rho_0u_w^2/6,\ \rho_0u_w^2/6)$, respectively, where the corrections in the y direction are unphysical. For a curved surface, when link 5 and link 8 are not paired, these over-corrections could potentially lead to significant force fluctuations.

We shall now discuss an alternative approach suggested by Aidun et al. [13]. When a solid body is moving relative to fixed fluid lattice nodes, it may cover some fluid nodes or uncover some solid nodes. Every time a fluid node is covered, the momentum at this node will go into the solid body as a momentum increment. This momentum increment, is interpreted as an impulse force given away by the fluid, which is added to the hydrodynamic force. On the other hand, when a solid node is uncovered, the new fluid node is filled with fluid momentum, which represents an impulse force that needs to be subtracted. Based on this interpretation, Aidun et al. [13] suggested to express the hydrodynamic force and torque as

$$\begin{aligned}\mathbf{F} &= \sum_{all\ \mathbf{x}_f}\sum_{\alpha_{bl}}\left[\tilde{f}_\alpha\left(\mathbf{x}_f,t\right)+f_{\bar{\alpha}}\left(\mathbf{x}_f,t+\delta t\right)\right]\mathbf{e}_\alpha+\sum_C\rho\left(\mathbf{x}_C\right)\mathbf{u}\left(\mathbf{x}_C\right)-\sum_U\rho\left(\mathbf{x}_U\right)\mathbf{u}\left(\mathbf{x}_U\right)\\ \mathbf{T} &= \sum_{all\ \mathbf{x}_f}\sum_{\alpha_{bl}}\left(\mathbf{x}_w-\mathbf{Y}_c\right)\times\left[\tilde{f}_\alpha\left(\mathbf{x}_f,t\right)+f_{\bar{\alpha}}\left(\mathbf{x}_f,t+\delta t\right)\right]\mathbf{e}_\alpha+\sum_C\rho\left(\mathbf{x}_C\right)\left(\mathbf{x}_w-\mathbf{Y}_c\right)\times\mathbf{u}\left(\mathbf{x}_C\right)\\ &\quad-\sum_U\rho\left(\mathbf{x}_U\right)\left(\mathbf{x}_w-\mathbf{Y}_c\right)\times\mathbf{u}\left(\mathbf{x}_U\right)\end{aligned} \tag{44}$$

where $\sum_C\rho(\mathbf{x}_C)\mathbf{u}(\mathbf{x}_C)$ and $\sum_U\rho(\mathbf{x}_U)\mathbf{u}(\mathbf{x}_U)$ are momenta at all covered and uncovered nodes, respectively, during a given time step. In most cases, the momentum gain and loss resulting from the covered and uncovered nodes cannot cancel with each other for two reasons. First, the pressure on the two sides of the solid body is usually unequal because of the relative motion. For instance, when a particle is translating in a quiescent fluid, the pressure in front of the particle is larger than the pressure behind the particle. Second, the total number of fluid nodes is not conserved because the covering and uncovering do not necessarily happen at the same time. For an unsteady flow, the velocities at the covered and uncovered nodes are

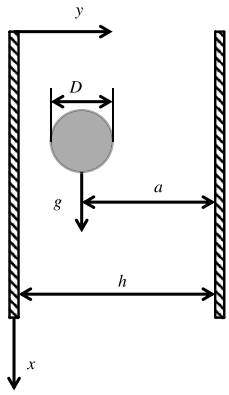

Fig. 13. Sketch to show a cylindrical particle settling freely in a vertical channel.

different. Hence, if the momenta at the covered and uncovered nodes are not considered, the cumulative error results from the covering and uncovering may lead to quantitatively inaccurate or even qualitative incorrect answers.

However, Aidun's correction happens in a discretized way, which could generate very large force fluctuations. The origin of this drawback is that LBM is not based on volumetric formulation, but define the populations only at discretized node points. To alleviate this, Chen et al. [26] uniformly distributes the momentum along the whole lattice link when accounting for the effect of covering and uncovering. Every time step a portion of lattice link is covered, it is equivalent to a portion of fluid originally sitting on part of the link is lost. On the other hand, when a portion of lattice link is uncovered, some fluid momentum is added. The lost fluid mass associated with the partial covering of a link during one time step can be approximated as $-2w_\alpha\rho_0\mathbf{e}_\alpha\cdot\mathbf{u}_w/c_s^2$. Therefore, Chen et al. [26] proposed to the following formulation

$$\mathbf{F}=\sum_{all\ \mathbf{x}_f}\sum_{\alpha}\left\{\left[\tilde{f}_\alpha\left(\mathbf{x}_f\right)+f_{\tilde{\alpha}}\left(\mathbf{x}_f,t+\delta t\right)\right]\mathbf{e}_\alpha-2w_\alpha\rho_0\frac{\mathbf{e}_\alpha\cdot\mathbf{u}_w}{c_s^2}\mathbf{u}_w\right\} \tag{45}$$

which will be referred to as the modified MEM or MMEM. In addition to the benefit of having much smaller fluctuations, when the particle velocity changes rapidly, this continuous correction could be more accurate than the impulse force correction. The reason is that, in the continuous correction, the correction term is determined by the instantaneous boundary velocity during a time interval instead of the velocity at the end of the time step implied by the discrete correction.

To bridge the gap between the two prospectives (*i.e.*, the first based on the fixed frame of reference and the second based on the frame of reference moving with $\mathbf{u}_w$), one can imagine the situation that the lattice grids move with the solid wall. In that case, there is neither any node point being covered or uncovered, nor any error associated with local Galilean invariance during the bounce-back momentum exchange, as the lattice node velocity is synchronized with the boundary velocity. In other words, the demand of this covering/uncovering correction arises spontaneously when local Galilean invariance becomes an issue.

One can also realize this connection by the apparent correction in GIMEM, Eq. (42), which is

$$\mathbf{F}_\alpha^{cor}=-\left[\tilde{f}_\alpha\left(\mathbf{x}_f,t\right)-f_{\tilde{\alpha}}\left(\mathbf{x}_f,t+\delta t\right)\right]\mathbf{u}_w\approx-\left(\tilde{f}_\alpha^{(eq)}-f_{\tilde{\alpha}}^{(eq)}\right)\mathbf{u}_w=-2w_\alpha\rho_0\frac{\mathbf{e}_\alpha\cdot\mathbf{u}_w}{c_s^2}\mathbf{u}_w, \tag{46}$$

which reduces to the expression in Eq. (45) for Chen et al.'s modified MEM. Our analysis indicates that the GIMEM in [25] is equivalently to the modified MEM in [26] if the following is true

$$\tilde{f}_\alpha\left(\mathbf{x}_f,t\right)-f_{\tilde{\alpha}}\left(\mathbf{x}_f,t+\delta t\right)\approx\tilde{f}_\alpha^{(eq)}-f_{\tilde{\alpha}}^{(eq)}. \tag{47}$$

This is essentially the assumption of the bounce-back of non-equilibrium distribution that is used in the implementation of the on-slip boundary condition, namely, Eq. (12). Therefore, we expect that GIMEM and MMEM lead to very similar results, even for the calculated force acting over a local area on the boundary. Still, we believe that GIMEM is the better formulation due to its clear physical interpretation and its simplicity.

4.4. Results from numerical tests of benchmark problems

We now compare the different versions of MEM implementations by considering several benchmark problems. The first problem is a cylinder settling freely in a vertical channel, as shown in Fig. 13. The channel length and width are $L=4$ cm and $h=0.4$ cm, respectively. The diameter of the cylinder is $D=0.1$ cm and its density is $\rho_p=1.03\rho_f$. Initially, the cylinder is placed at $a=0.324$ cm away from the right wall so the particle will rotate due to the unbalanced torque. The results are compared to those from the finite element method (FEM) in [48].

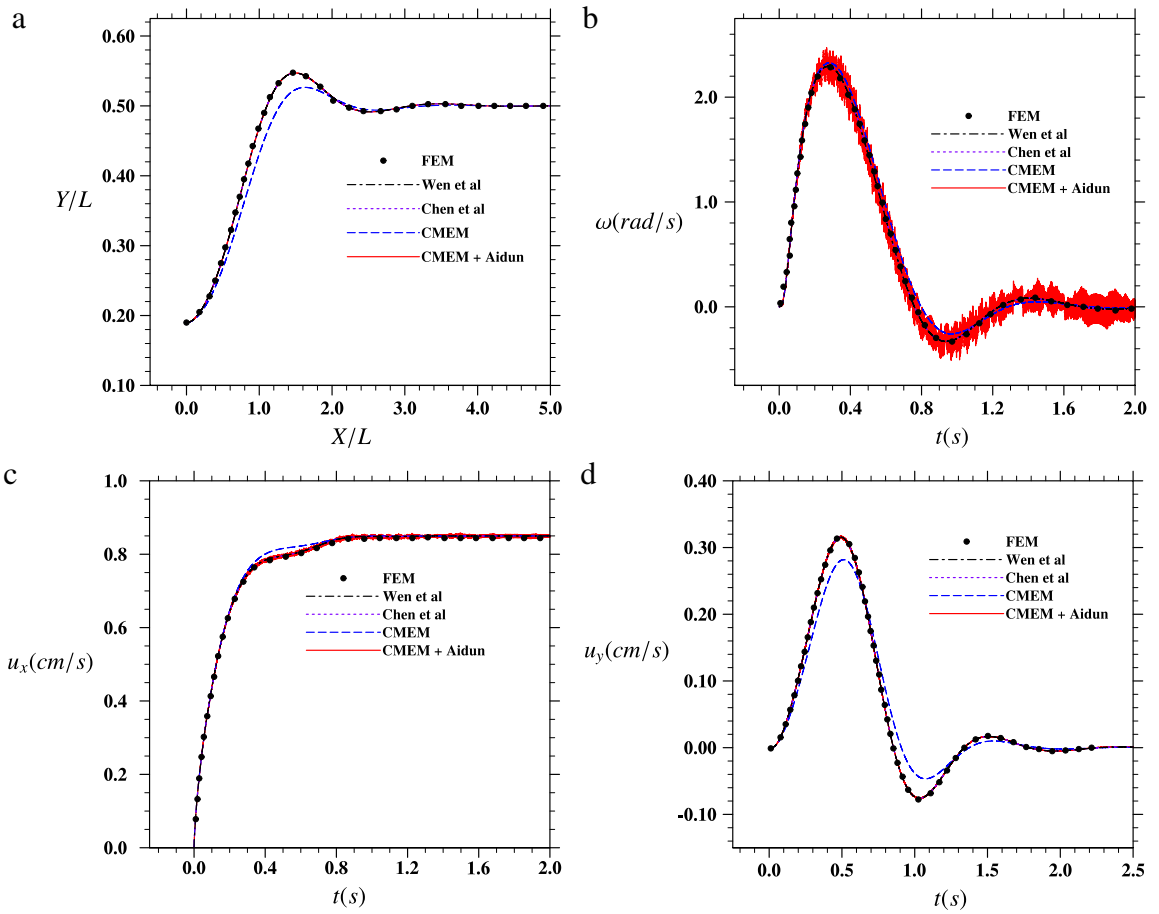

Fig. 14. The results of time dependent cylindrical particle motion: (a) trajectories; (b) angular velocities; (c) vertical velocities; (d) horizontal velocities.

In Fig. 14 we show the time evolution of particle trajectory, velocity and angular velocity, for different MEM implementations. Aidun's correction to the conventional MEM is also included here. As expected, without correction, the conventional MEM does not yield an accurate result when compared to the FEM benchmark. When Aidun's correction is included, it gives generally accurate results but can lead to remarkable fluctuation in the angular velocity. One the other hand, the results from two new momentum exchange models are essentially identical and are in excellent agreement with the FEM benchmark.

The evolutions of hydrodynamic forces and torque are shown in Fig. 15. Now we observe that CMEM with Aidun's correction contains extremely large force fluctuations when compared to the other two recent formulations, GIMEM and MMEM. The results from Aidun's correction [13] have already been smoothened by taking the local-in-time average of 30 time steps, while the results from the other two implementations are not averaged at all. The high level of fluctuations in Aidun's correction is due to the discretized correction manner noted previously. Here the Aidun's correction is not the link-wise correction, but by Eq. (43). The reason for large fluctuation is because the correction happens in a discretized manner. Therefore, even though using the CMEM with Aidun's correction can yield the physically correct trajectory and velocities, the two recently improved MEMs (GIMEM and MMEM) are highly recommended for their much better accuracy in force evaluation and potentially better numerical stability (due to weaker force fluctuations).

Furthermore, we shall demonstrate that, using CMEM can sometimes lead to qualitatively incorrect results, as noted already in [25]. Consider the Segré–Silberberg effect for a neutrally buoyant particle suspended in a pressure driven pipe flow (as shown in Fig. 16). The radii of the pipe and the particle are $R = 2.5$ cm and $r = 0.375$ cm, respectively. The flow Reynolds number defined as $Re = 2U_m R/\mu$ is 100, where U_m is the mean flow velocity and μ is the kinematic viscosity that is set to 1 cm^2/s. Initially, the particle is released at $a/R = 0.21$. The streamwise boundary condition in this case is periodic. The trajectories of the particle are shown in Fig. 16 for different MEM implementations. We find that both GIMEM and MMEM successfully reproduce the Segré–Silberberg effect while CMEM cannot. The horizontal straight line marks the correct equilibrium position obtained by arbitrary Lagrangian–Eulerian (ALE) method [49].

Finally, GIMEM is applied and tested in a direct numerical simulation of a particle-laden turbulent channel flow [50]. The channel flow has a resolution of $N_x \times N_y \times N_z = 600 \times 299 \times 300$, where N_x, N_y and N_z are the grid points used in streamwise,

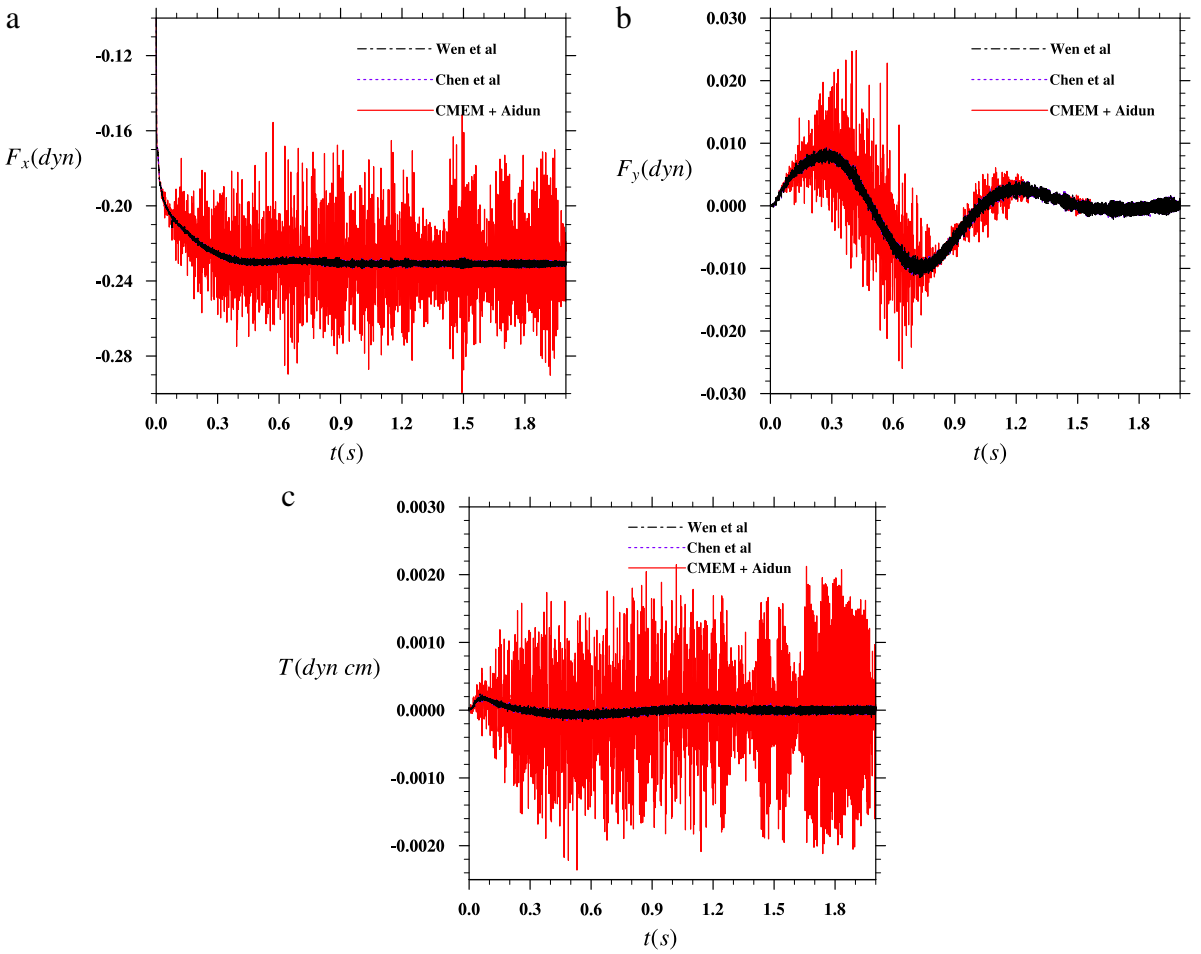

Fig. 15. The results of time dependent hydrodynamic forces and torques acting on the particle: (a) vertical forces; (b) horizontal forces; (c) torques.

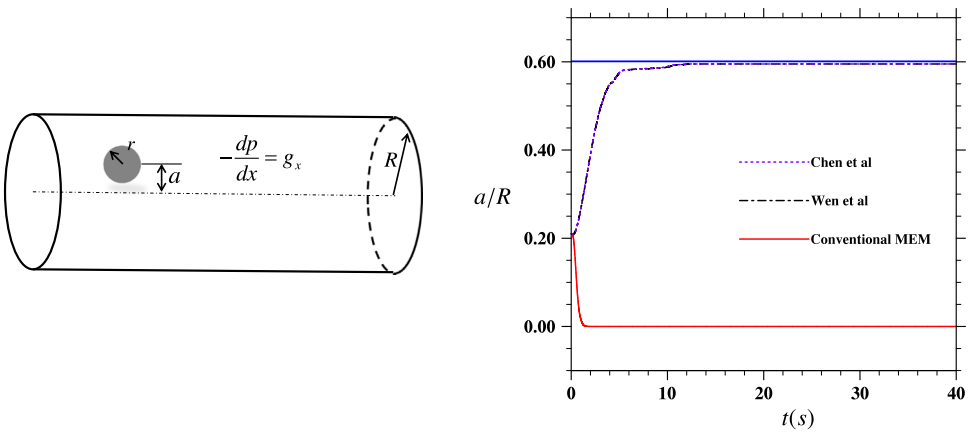

Fig. 16. A neutrally buoyant particle in an infinite long pressure driven pipe flow.

transverse and spanwise direction, respectively. The Reynolds number, defined by the friction velocity of wall bounded turbulence is $Re_\tau = 180$. 270 neutrally buoyant spherical particles with diameters $a = 30.0$ are dispersed randomly in the channel, with a total volume fraction of $\phi_p = 7.07\%$. The streamwise boundary condition is periodic. The non-slip boundary condition on the particle surface is realized by the bounce back rules in Eqs. (15) and (17). The refilling scheme used in this case is the velocity-constrained extrapolation scheme (see Section 5.1.4). The non-uniform shear distribution in a turbulent channel flow could produce effects similar to the Segré–Silberberg effect, therefore, it is very important to have an accurate

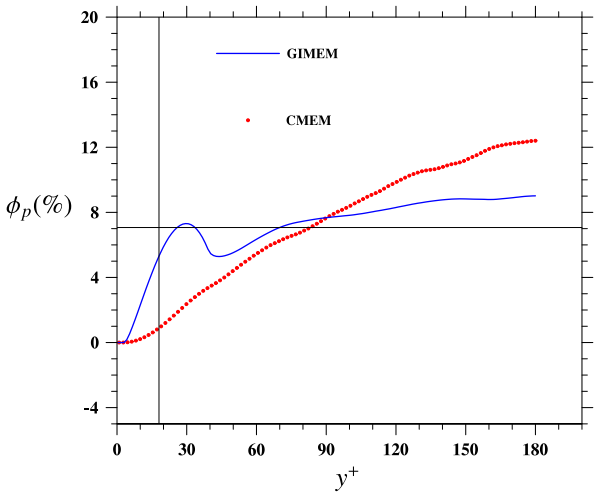

Fig. 17. The averaged particle volume fraction at a function of y^+ in a particle-laden turbulent channel flow. $y^+ = 0$ represents the channel wall, and $y^+ = 180$ is the center of the channel.

MEM. In Fig. 17, we show averaged particle volume fraction (APVF), defined as the number of node points that are covered by the solid particles over the total number of grid points on a given two-dimensional plane parallel to the channel wall, for different y^+ locations. Results based on CMEM and GIMEM are compared in Fig. 17, showing a significant disagreement. Using CMEM, the APVF increases monotonically away from the channel wall, indicating that there are more particles in the center region. In contrast, when GIMEM is used, the APVF shows a local maximum near the wall that is also mentioned in [51] and a more uniform overall distribution along the channel width, which has also been observed in [4]. Therefore, the results based on CMEM are physically incorrect, while GIMEM produces sound results.

5. Refilling

We shall now consider the refilling implementation for new fluid nodes as a solid particle moves relative to the fixed lattice grid. Early studies for suspension flow simulation [12] assumed that fluid nodes occupied even the region inside a solid particle. The fluid–solid interface is treated as a moving shell. The same bounce-back scheme is used for two sides of the shell to guarantee the no-slip boundary condition. The hydrodynamic force acting on the particle is affected by the momentum exchanges on both sides. This approach avoids the refilling problem as the distribution functions are defined on all lattice nodes at all times. In this approach, an implicit assumption is that the distribution functions from inside the solid particle can be used for the flow outside the solid particle. This, however, has no rigorous physical basis. Although the velocity is continuous across the solid boundary when both the flow inside and flow outside the solid particle are considered, there could be jumps in pressure and shear stresses. Therefore, this approach is not valid and usually result in numerical inaccuracy.

The more accepted approach is to exclude the lattice nodes inside the solid surfaces [13]. In this approach, only the outside fluid contributes to the hydrodynamic force, which is more physically meaningful than the shell model. However, the refilling problem has to be addressed carefully [13,27,29]. For complex flows involving many moving solid boundaries, the method to refill the distribution functions at the new fluid nodes may significantly affect the accuracy and stability.

5.1. Refilling schemes

There have not been too many proposals to address the refilling problem. We review below three implementations [28,27,29] from the literature and an improved implementation we developed recently. Those schemes are selected due to their simplicity and similarity. The schemes requires more complicated evaluation of the stress tensor are therefore not considered here.

5.1.1. Averaged extrapolation refilling

The extrapolation scheme was originally proposed as a boundary condition in [43], and later used in the refilling process [28]. As shown in Fig. 18, for a newly uncovered node $\mathbf{x}_{new}$, there could be multiple extrapolation directions ($i = 1, 2, 3, 5, 6$) that point into the fluid. In each extrapolation direction α, a linear extrapolation is applied to all directions to obtain the distribution functions at $\mathbf{x}_{new}$

$$f_{\alpha,i}(\mathbf{x}_{new}, t + \delta t) = 2f_\alpha\left(\mathbf{x}_{new} + \mathbf{e}_i\delta t, t + \delta t\right) - f_\alpha\left(\mathbf{x}_{new} + 2\mathbf{e}_i\delta t, t + \delta t\right). \tag{48}$$

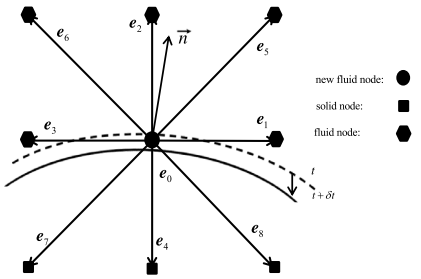

Fig. 18. The lattice structure of new fluid node near the solid boundary.

Then, we construct the missing populations by taking the average over these extrapolation directions

$$f_\alpha(\mathbf{x}_{new}, t+\delta t) = \frac{1}{N}\sum_{i=1,2,3,5,6} f_{\alpha,i}(\mathbf{x}_{new}, t+\delta t) \tag{49}$$

where N is the total number of possible extrapolation directions.

5.1.2. Normal extrapolation refilling

Different from the averaged extrapolation refilling scheme, where the unknown distribution functions are contributed by information from every possible extrapolation direction, in this scheme, only the direction that maximizes the quantity $\vec{n}\cdot\mathbf{e}_c$ is used for the extrapolation, where $\vec{n}$ is the outer unit normal of local solid surface from where the new fluid point was uncovered (as in Fig. 18). After determining the direction $\mathbf{e}_c$, all unknown distribution functions at the new fluid node are obtained by a quadratic extrapolation.

$$f_\alpha(\mathbf{x}_{new}, t+\delta t) = 3f_\alpha(\mathbf{x}_{new}+\mathbf{e}_c\delta t, t+\delta t) - 3f_\alpha(\mathbf{x}_{new}+2\mathbf{e}_c\delta t, t+\delta t) + f_\alpha(\mathbf{x}_{new}+3\mathbf{e}_c\delta t, t+\delta t). \tag{50}$$

Compared with the averaged extrapolation, this scheme may be more tedious when dealing with solid surfaces with irregular shape due to the need of finding the surface normal and $\mathbf{e}_c$.

5.1.3. Equilibrium plus non-equilibrium refilling

This scheme partitions the unknown distribution functions at the new fluid node into two parts, the equilibrium part and non-equilibrium part. The equilibrium part is computed in terms of the local wall velocity $\mathbf{u}_w$ and a fluid density which is usually taken as the averaged density $\overline{\delta\rho}$ of available neighboring fluid nodes. The non-equilibrium part is defined by simply copying the non-equilibrium part from the neighboring fluid node along the extrapolation direction $\mathbf{e}_c$ (same as defined in Section 5.1.2). Namely,

$$f_\alpha(\mathbf{x}_{new}, t+\delta t) = f_\alpha^{(eq)}\left(\mathbf{u}_w, \overline{\delta\rho}\right) + f_\alpha^{(neq)}\left(\mathbf{x}_{new}+\mathbf{e}_c\delta t, t+\delta t\right). \tag{51}$$

Note that in this scheme, populations in all directions at the new fluid node are re-computed as above, instead of only the unknown populations in the averaged extrapolation refilling and the normal extrapolation refilling.

5.1.4. Velocity-constrained normal extrapolation refilling

This scheme begins with an observation that the distribution functions in the normal extrapolation refilling scheme may not precisely satisfy the Dirichlet boundary condition at the new fluid node (which is very close to the solid surface). In addition, in MRT LBM, we can take the advantage that the velocity can be constrained to the wall velocity without changing other macroscopic properties (such as pressure and stress components). The procedure of this refilling scheme is as follows. First, complete the normal extrapolation refilling to obtain temporary distribution functions for the missing distributions. Then, compute all moments at the new fluid node by multiplying the transfer matrix $\mathbf{M}$ as

$$\mathbf{m}(\mathbf{x}_{new}, t+\delta t) = \mathbf{M}\hat{\mathbf{f}}(\mathbf{x}_{new}, t+\delta t) \tag{52}$$

where $\hat{\mathbf{f}}$ indicates the temporary distribution function after the unconstrained normal extrapolation. Next, enforce the no-slip boundary condition by constraining the momentum moments.

$$j_x = \rho_0 u_w, \qquad j_y = \rho_0 v_w, \qquad j_z = \rho_0 w_w. \tag{53}$$

Under the nearly incompressible formulation, this is equivalent to constrain the velocity to the local solid surface velocity. Finally, transfer the moments $\mathbf{m}^*$ after the above modification back to the distribution functions as

$$\mathbf{f}(\mathbf{x}_{new}, t+\delta t) = \mathbf{M}^{-1}\mathbf{m}^*(\mathbf{x}_{new}, t+\delta t) \tag{54}$$

where $\mathbf{m}^*$ means the moment vector after the velocity is constrained. In this procedure, except the velocity, no other macroscopic quantities are changed. We find that this constraint can significantly reduce the fluctuations in the hydrodynamic forces when compared to the unconstrained normal extrapolation.

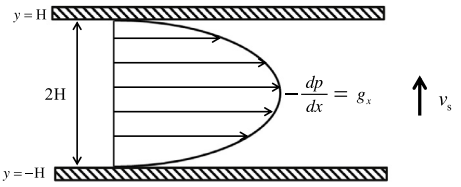

Fig. 19. The Poiseuille flow with a constant system velocity v_s added in the normal direction.

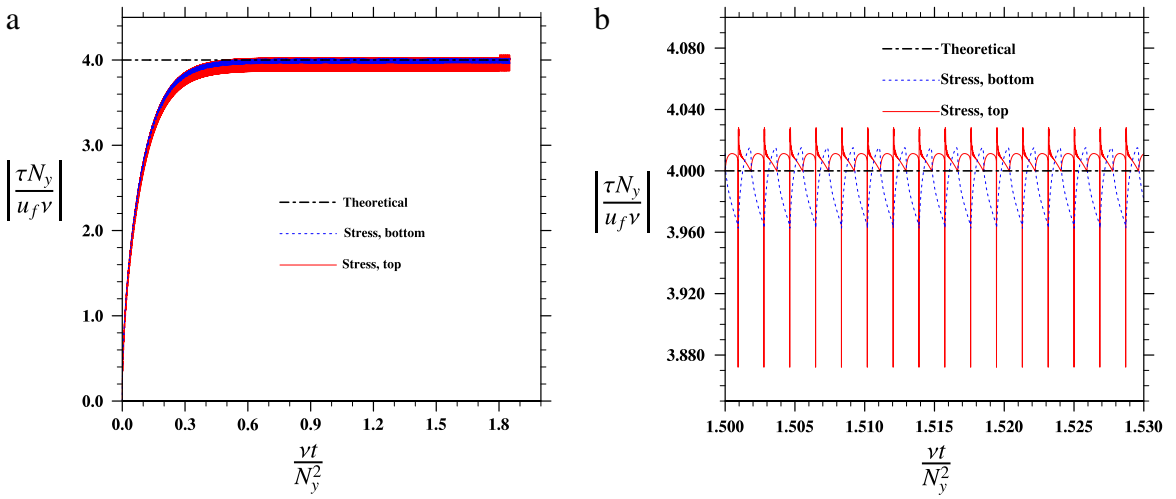

Fig. 20. The time evolution of the shear stress on the top and bottom walls, the results are obtained with the equilibrium plus non-equilibrium refilling scheme: (a) full view, (b) zoom-in view.

5.2. The effects of the refilling scheme on the hydrodynamic force

Refilling the information at the new fluid nodes plays an important role in a successful LBM simulation. Even though all refilling schemes have already been shown to produce physically correct results, they could also contribute to strong fluctuations in the resulting hydrodynamic force [27]. In addition, other factors, such as the change of distance between boundary nodes and the moving boundary (as discussed in Section 3), as well as the fluctuations in the number of lattice nodes covered by a solid particle, can affect those fluctuations. It is not clear how much of the total force fluctuation is directly caused by the refilling process.

Before comparing different refilling schemes, we shall isolate the effect of the refilling process from other sources. For this purpose, we consider a steady-state Poiseuille channel flow with a constant normal velocity being added to the whole system (shown in Fig. 19). In this case, both top and bottom walls move at a same speed, the number of fluid nodes remain unchanged. Refilling is only needed for the top wall. Therefore, by examining the difference between the shear stress acting on the top and bottom wall (if there are different), the only source of this difference is the refilling process associated with the top wall. Therefore, the contribution of refilling to the force fluctuation is isolated in this simple benchmark case.

In Fig. 20 we show the shear stresses on both the top and bottom wall. In this test, the channel width is 60 lattice units. The x-component velocity at the centerline position is set to $U_x = 0.1c$ and $U_y = 0.005c$, where c is the lattice velocity unit $\delta x/\delta t$. The Reynolds number of the flow is $Re = U_x H/\nu = 180$. In this case we apply the interpolated bounce back scheme in Eqs. (14) and (16) to the two moving walls. The stress on the wall are extrapolated from the stress tensor at the first three fluid nodes from each wall. For demonstration purpose, only the results from the equilibrium plus non-equilibrium refilling are shown. It is clear that the long spikes in the result of stress on the top wall is related to the refilling process (by the times they occur, which match with the crossing times of the top wall and a layer of new fluid nodes). These long spikes are due to the inaccurate construction of unknown populations at the new fluid nodes near the top wall. They may have a strong effect on the accuracy and stability of the LBM simulation. The maximum deviations of shear stress from its theoretical value, defined as $\frac{\max(|\tau-\tau_{theo}|)}{|\tau_{theo}|}$ is 0.94% on the bottom wall but 3.2% on the top wall.

In more complex cases, such as a particle-laden turbulent flow, the simulation could be more sensitive to the selection of refilling schemes. In Fig. 21 we present visualizations of pressure field on a 2D slice from a particle-laden turbulent channel flow simulation with two different refilling schemes at 1600 time steps after the exactly same initial conditions (with the same physical setting as what used in Fig. 17). Comparing the two pressure visualizations, differences between

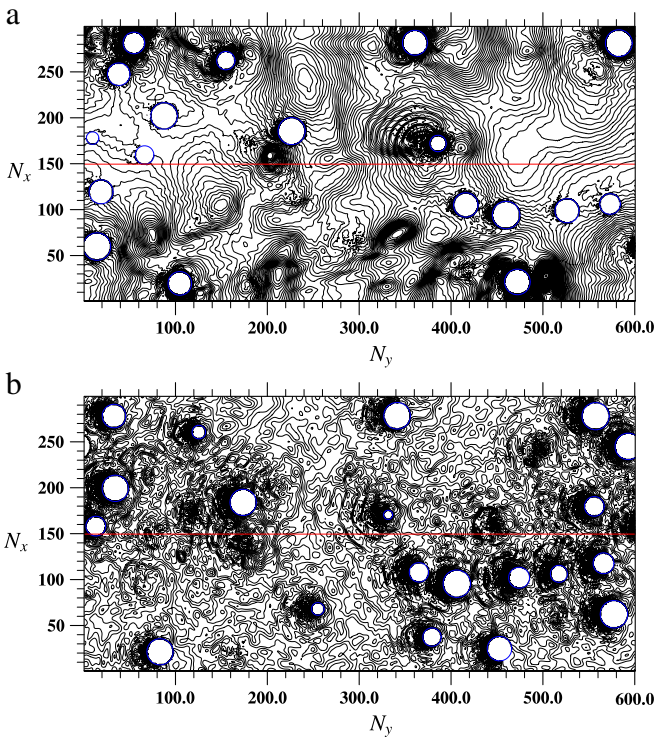

Fig. 21. Visualizations of the pressure fields on a 2D slice from a particle-laden turbulent channel flow direct numerical simulation with different refilling schemes. The results are obtained at 1600 time steps after the exactly same initial condition. (a) The velocity constrained normal extrapolation scheme, (b) the normal extrapolation scheme.

Table 3
The level of normalized force fluctuations for the case of vertically moving Poiseuille flow.

Refilling schemes	$\bar{\tau}$	σ_1	σ_2
Average	1.0028	5.0024×10^{-2}	6.3648×10^{-3}
Eq + Neq	1.0020	3.9135×10^{-2}	2.6546×10^{-3}
Normal	0.9999	1.3083×10^{-2}	1.5516×10^{-3}
Constrained	1.0005	1.0077×10^{-2}	1.3666×10^{-3}

particle positions are clearly observed, indicating a greater impact of the refilling scheme on the particle dynamics. Also, in Fig. 21(b) the noises are much more severe than in Fig. 21(a). These noises can lead to numerical instability. In fact, the simulation with the normal extrapolation scheme diverged after another 200 time steps from the snapshot in Fig. 21, while the simulation with the velocity constrained normal extrapolation scheme is always stable. This comparison implies that the velocity-constrained refilling scheme enhances numerical stability.

5.3. Comparison results from benchmark cases

To compare the performance of different refilling schemes, two benchmark cases are considered.

5.3.1. Vertically moving Poiseuille flow

The first case is the horizontal Poiseuille flow with a vertical system velocity introduced in Fig. 19. The Galilean invariance states that constant system velocity v_s does not affect the streamwise velocity profile and the stress profile. The results of shear stress on the top wall using different refilling schemes are shown in Fig. 22. The boundary interpolation and stress calculation follow the same schemes as the case in Fig. 20. Even though every scheme yield the correct mean value to a good approximation, they have quite different magnitudes in the force fluctuations. In general, the stress results with velocity-constrained normal extrapolation has the smallest fluctuation magnitude (Table 3) with only 20%–77% maximum fluctuation and 21%–88% overall fluctuation of those counterparts from other schemes. Here, the $\bar{\tau}$, σ_1 and σ_2 are defined as the mean values of the stress, the maximum variance of stress ($\tau_{\max} - \tau_{\min}$) and the r.m.s stress when the flow reaches steady state, all results are normalized by the theoretical stress.

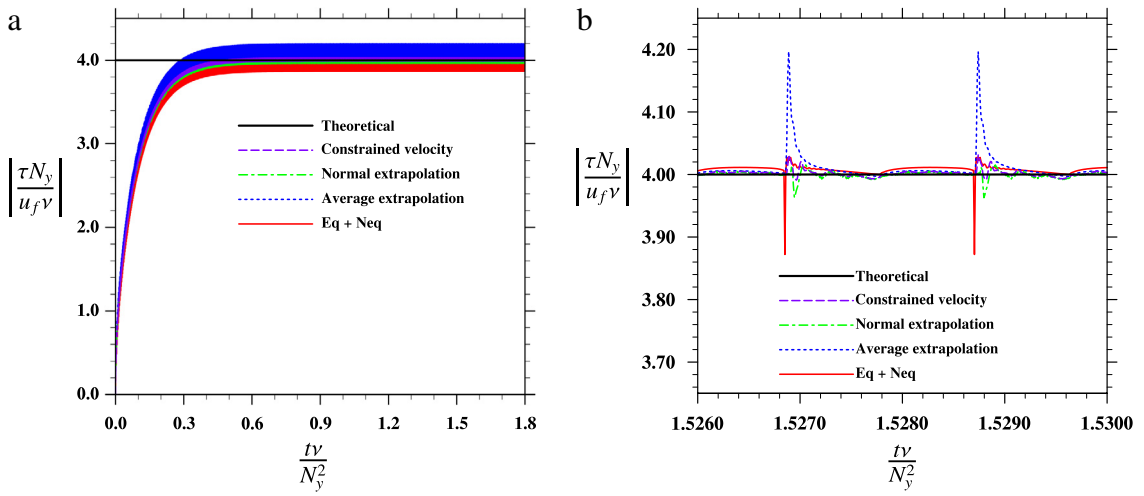

Fig. 22. The shear stress on the top wall as a function of time, with different refilling schemes. (a) Full view; (b): zoom-in view.

Table 4
The level of fluctuations in the calculated F_x and F_y normalized by $(\rho_p - \rho_f)A_p g_x$, for the elliptic particle setting freely in a 2D vertical channel.

Bounce back schemes	σ_x	σ_y
Average	9.2138×10^{-3}	8.6088×10^{-3}
Eq + Neq	4.8648×10^{-3}	5.3872×10^{-3}
Normal	5.6871×10^{-3}	5.9911×10^{-3}
Constrained	4.2861×10^{-3}	4.9108×10^{-3}

5.3.2. Elliptic particle settling freely in a vertical channel

Next, different refilling schemes are tested in the case of elliptic particle settling freely in vertical channel (same as in Section 3.2.3). The resulting particle trajectories and rotation angles are compared in Fig. 23. Even though all results generally fit well with the corresponding FEM results, a slight deviation is observed near the turning points in the trajectory with the averaged extrapolation refilling, leading to inaccurate trajectory later on. When examining the force evaluations in Fig. 24, we observe the greatest fluctuation from the averaged extrapolation scheme, which explains the slightly inaccurate trajectory in Fig. 23. Again, the constrained normal extrapolation yields the smallest fluctuation level among all the schemes as shown in Table 4. The improvement is from 12% to 200% over other schemes. The fluctuation magnitudes (normalized by $(\rho_p - \rho_f)A_p g_x$) are quantified using the same method as in Section 3 for the same case. It should be noted that although in those simple cases the improvement in the force evaluation does not bring significant improvement in the overall performance, as we show in Fig. 21, in the more complex cases, the selection of a proper refilling scheme could be essential for an accurate and numerically stable simulation.

6. Summary and outlook

In this paper, three important implementation issues, namely, fluid–solid boundary treatments, hydrodynamic force evaluation on a solid particle, and the refilling problem, are reviewed and studied systematically, when the MRT LBM approach is used to simulate fluid-moving particle system. The work was motivated by our target problem of simulating turbulent flow laden with many moving solid particles. The question is whether LBM can provide a reliable and robust approach for this complex problem with flow around individual particles fully resolved (i.e., the particle-resolved direct numerical simulation). Specifically, whether the trajectory of a solid particle in a turbulent flow and hydrodynamic force acting on the particle can be accurately simulated.

We provide our own interpretations and a critical review of different implementations. To compare different implementation methods systematically, we develop a series of benchmark cases by isolating a given issue under consideration and keeping the identical implementations for the remaining two issues. We have demonstrated that some of the issues are inter-related. Our systematic approach allows us to reach a set of conclusions that would help optimize the implementation details of MRT LBM when applied to high-Reynolds number particle-laden flow systems.

First, for implementing the solid boundary condition at the fluid-moving solid interface, all tested interpolated bounce-back schemes can maintain the second order accuracy. However, different bounce back schemes exhibit different levels of errors and force fluctuations on a moving particle. We conclude that quadratic (or higher order) interpolations are in general better in the linear interpolation schemes, in that they yield more accurate results with weaker unphysical fluctuations. The

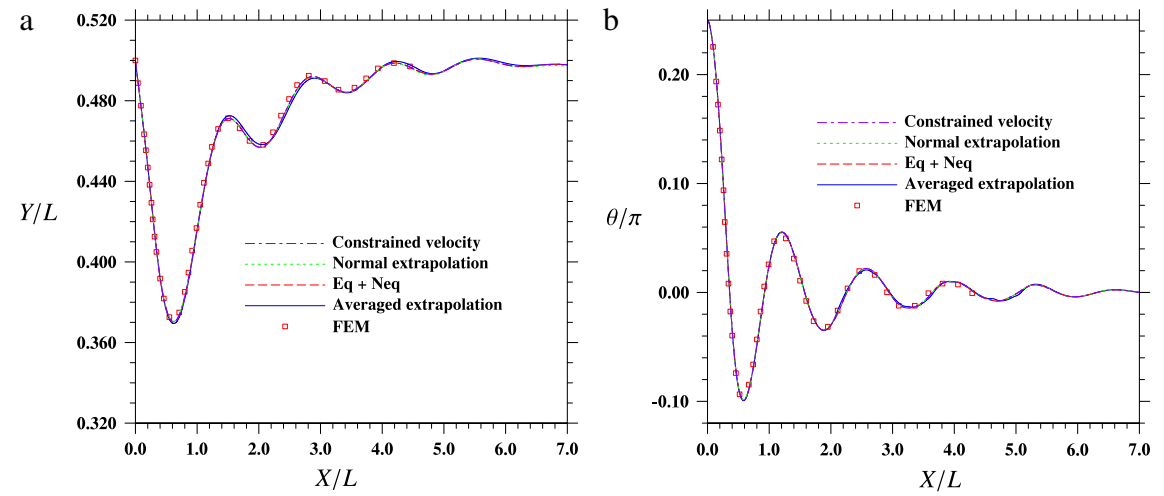

Fig. 23. Trajectories (a) and rotation angles (b) of the elliptic particle sedimentation problem obtained with different refilling schemes.

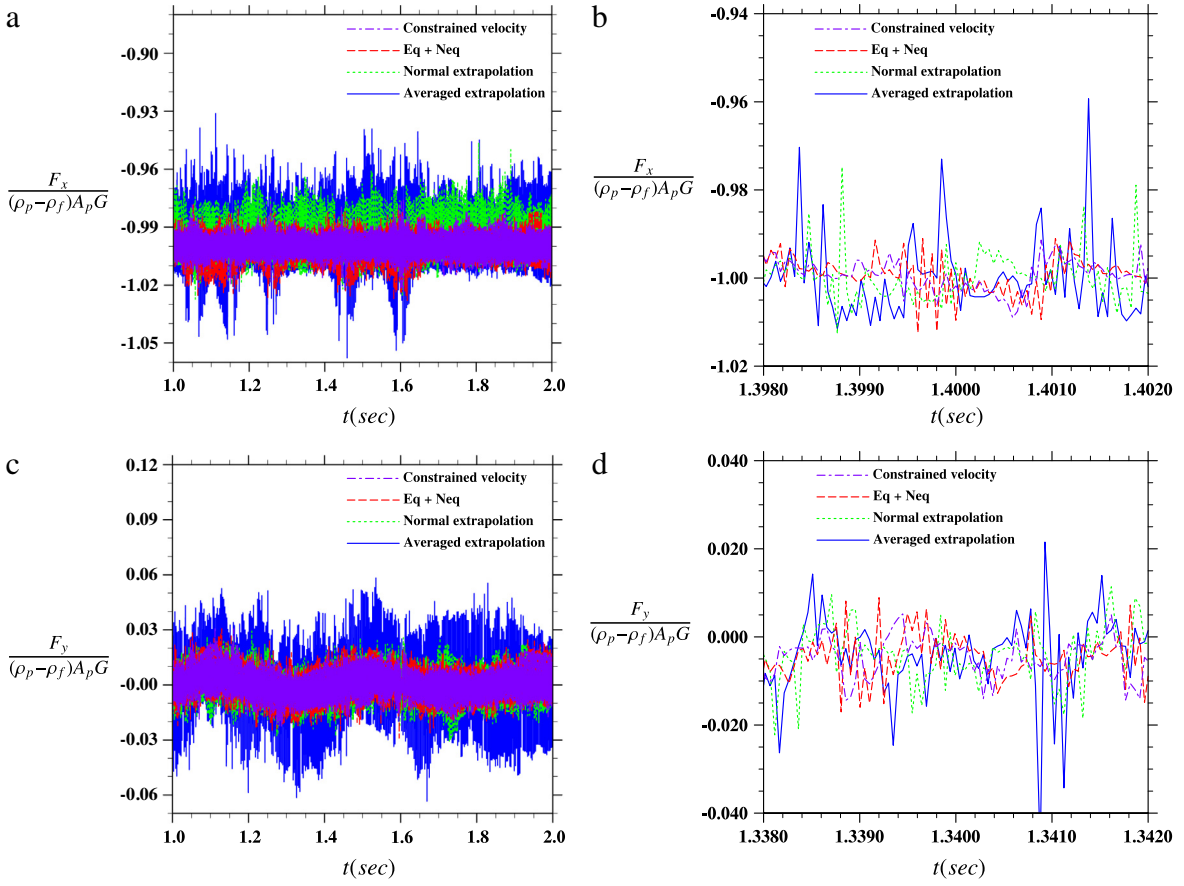

Fig. 24. Time dependent forces acting on an elliptic particle settling in a vertical channel with different filling schemes: (a) F_x, full view; (b) F_x, zoom-in view; (c) F_y, full view; (d) F_y, zoom-in view.

unphysical force fluctuations are an inherent characteristic of all numerical methods for moving solid particles using a fixed grid. They can affect numerical stability when complex flows at high Reynolds numbers are considered.

Second, we made an effort to contrast different MEM schemes and identify the most accurate (MEM) schemes for computing force and torque acting on a moving particle. Due to its simplicity and direct connection to the kinetic theory, MEM has been widely used to evaluate the hydrodynamic force at the fluid–solid surface in LBM. However, the related issue of local

Galilean variance has only very recently been successfully addressed [26,25]. As in these recent studies, we have demonstrated that a scheme with rigorous local Galilean variance can yield more accurate results of hydrodynamic force. We have shown both theoretically and numerically that the Galilean invariant MEM of Wen et al. [25] and the modified MEM of Chen et al. [26] are essentially identical, although they address the issue of local Galilean invariance from two different perspectives. We also pointed out that over-corrections made in the literature [23,24] may lead to large force fluctuations, and this should be avoided. Schemes that fail to satisfy the local Galilean invariance can even lead to qualitatively incorrect results.

Finally, the refilling process has a particular significance in the LBM simulation with moving solid boundaries. If the information at new fluid nodes is not properly constructed, it can strongly affect the accuracy of force evaluation or even ruin the whole simulation by introducing numerical instability. We compared several different refilling schemes. Although the averaged extrapolation scheme provides reasonable results of particle trajectory and velocity, the high-level time-dependent force fluctuations clearly imply that it is not a good approach for moving particle simulations. A new velocity-constrained normal extrapolation refilling provides the best performance in terms of both accuracy and numerical stability. This velocity-constrained refilling takes advantage of the flexibility offered by MRT.

We wish to stress that the interpolated bounce-back and GIMEM are both consistent with the lattice Boltzmann scheme. It does not require any local regularization or smoothing as in the typical immersed boundary method [52,53]. In this sense, the overall optimized implementation of MRT LBM implied by this work provides a better representation of the local profiles near a moving solid surface [9]. We also wish to point out that while this study focuses on the mesoscopic lattice Boltzmann approach, some of the discussions related to moving fluid–solid interfaces are relevant to conventional CFD using the Navier–Stokes equations.

Acknowledgments

This work has been supported by the US National Science Foundation (NSF) under grants CBET-1235974 and AGS-1139743 and by Air Force Office of Scientific Research under grant FA9550-13-1-0213. LPW also acknowledges support from the Ministry of Education of PR China and Huazhong University of Science and Technology through Chang Jiang Scholar Visiting Professorship. Computing resources are provided by National Center for Atmospheric Research through CISL-P35751014, and CISL-UDEL0001 and by University of Delaware through NSF CRI 0958512.

References

[1] T.M. Burton, J.K. Eaton, Fully resolved simulations of particle-turbulence interaction, J. Fluid Mech. 545 (2005) 67–111.
[2] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2005) 448–476.
[3] M. Uhlmann, Interface-resolved direct numerical simulation of vertical particulate channel flow in the turbulent regime, Phys. Fluids 20 (2008) 053305.
[4] X. Shao, T. Wu, Z. Yu, Fully resolved numerical simulation of particle-laden turbulent flow in a horizontal channel at a low Reynolds number, J. Fluid Mech. 693 (2012) 319–344.
[5] Z. Zhang, A. Prosperetti, A second-order method for three-dimensional particle simulation, J. Comput. Phys. 210 (2005) 292–324.
[6] K. Yeo, S. Dong, E. Climent, M.R. Maxey, Modulation of homogeneous turbulence seeded with finite size bubbles or particles, Int. J. Multiphase Flow 36 (2010) 221–233.
[7] H. Homann, J. Bec, Finite-size effects in the dynamics of Neutrally buoyant particles in turbulent flow, J. Fluid Mech. 651 (2010) 81–91.
[8] H. Gao, H. Li, L.P. Wang, Lattice Boltzmann simulation of turbulent flow laden with finite-size particles, Comput. Math. Appl. 65 (2013) 194–210.
[9] L.-P. Wang, O. Ayala, H. Gao, et al., Study of forced turbulence and its modulation by finite-size solid particles using the lattice Boltzmann approach, Comput. Math. Appl. 67 (2014) 363–380.
[10] A. Ten Cate, J.J. Derksen, L.M. Portela, H.E.A. van den Akker, Fully resolved simulations of colliding monodisperse spheres in forced isotropic turbulence, J. Fluid Mech. 519 (2004) 233–271.
[11] J. Latt, B. Chopard, O. Malaspinas, M. Deville, A. Michler, Straight velocity boundaries in the lattice Boltzmann method, Phys. Rev. E 77 (2008) 056703.
[12] A.J.C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical fundation, J. Fluid Mech. 271 (1994) 285–309.
[13] C.K. Aidun, Y. Lu, E.J. Ding, Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation, J. Fluid Mech. 373 (1998) 287–311.
[14] M. Bouzidi, M. Fordaouss, P. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, 13 (2001) 3452–3459.
[15] O. Filippova, D. Hänel, Grid refinement for lattice-BGK models, J. Comput. Phys. 147 (1998) 219–228.
[16] R. Mei, L.-S. Luo, W. Shyy, An accurate curved boundary treatment in the lattice Boltzmann method, J. Comput. Phys. 155 (1999) 307–330.
[17] R. Mei, W. Shyy, D. Yu, et al., Lattice Boltzmann method for 3-D flows with curved boundary, J. Comput. Phys. 161 (2000) 680–699.
[18] D. Yu, R. Mei, L.-S. Luo, et al., Viscous flow computations with the method of lattice Boltzmann equation, Prog. Aerosp. Sci. 39 (2003) 329–367.
[19] I. Ginzburg, D. d'Humières, Multireflection boundary conditions for lattice Boltzmann models, Phys. Rev. E 68 (2003) 066614.
[20] Z. Guo, C. Zheng, B. Shi, An extrapolation method for boundary conditions in lattice Boltzmann method, Phys. Fluids 14 (2002) 2007–2010.
[21] B. Chun, A.J.C. Ladd, Interpolated boundary condition for lattice Boltzmann simulations of flows in narrow gaps, Phys. Rev. E 75 (2007) 066705.
[22] R. Wei, D. Yu, W. Shyy, et al., Force evaluation in the lattice Boltzmann method involving curved geometry, Phys. Rev. E 65 (2002) 041203.
[23] A. Caiazzo, M. Junk, Boundary forces in lattice Boltzmann: Analysis of momentum exchange algorithm, Comput. Math. Appl. 55 (2008) 1415–1423.
[24] J.R. Clausen, C.K. Aidun, Galilean invariance in the lattice-Boltzmann method and its effect on the calculation of rheological properties in suspensions, Int. J. Multiphase Flow 35 (2009) 307–311.
[25] B. Wen, C. Zhang, Y. Yu, et al., Galilean invariant fluid–solid interfacial dynamics in lattice Boltzmann simulations, J. Comput. Phys. 266 (2014) 161–170.
[26] Y. Chen, Q. Cai, Z. Xia, et al., Momentum-exchange method in lattice Boltzmann simulations of particle–fluid interactions, Phys. Rev. E 88 (2013) 013303.
[27] P. Lallemand, L.-S. Luo, Lattice Boltzmann method for moving boundaries, J. Comput. Phys. 184 (2003) 406–421.
[28] H. Fang, Z. Wang, Z. Lin, et al., Lattice Boltzmann method for simulating the viscous flow in large distensible blood vessels, Phys. Rev. E 65 (2002) 051925.
[29] A. Caiazzo, Analysis of lattice Boltzmann nodes initialisation in moving boundary problems, Prog. Comput. Fluid Dyn. 8 (2008) 3–10.
[30] J. Shin, J. Lee, S. Lee, A strain-rate model for a lattice Boltzmann BGK model in fluid–structure interactions, Comput. Fluids 88 (2013) 126–135.
[31] S. Krithivasan, S. Wahal, S. Ansumali, Diffused bounce-back condition and refill algorithm for the lattice Boltzmann method, Phys. Rev. E 89 (2014) 033313.

[32] Y. Qian, D. d'Humières, P. Lallemand, Lattice BGK models for Navier–Stokes equation, Europhys. Lett. 17 (1992) 479–484.
[33] X. He, L.S. Luo, Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation, Phys. Rev. E 56 (1997) 6811.
[34] S. Chen, G. Doolen, Lattice Boltzmann method for fluid flows, Annu. Rev. Fluid Mech. 30 (1998) 329–364.
[35] S. Hou, Q. Zou, S. Chen, et al., Simulation of cavity flow by the lattice Boltzmann method, J. Comput. Phys. 118 (1995) 329–347.
[36] P. Lallemand, L.S. Luo, Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability, Phys. Rev. E 61 (2000) 6546.
[37] D. d'Humières, I. Ginzburg, M. Krafczyk, et al., Multiple-relaxation-time lattice Boltzmann models in three-dimensions, Philos. Trans. R. Soc. Lond. Ser. A 360 (2002) 437–451.
[38] L.-S. Luo, W. Liao, X. Chen, et al., Numerics of the lattice Boltzmann method: Effects of collision models on the lattice Boltzmann simulations, Phys. Rev. E 83 (2011) 056710.
[39] Y. Zong, C. Peng, Z. Guo, L.P. Wang, Designing correct fluid hydrodynamics on a rectangular grid using MRT lattice Boltzmann approach, Comput. Math. Appl. 72 (2016) 288–310.
[40] X. He, L.S. Luo, Lattice Boltzmann model for the incompressible Navier–Stokes equation, J. Stat. Phys. 88 (1997) 927–944.
[41] P. Lammers, K.N. Beronov, R. Volkert, et al., Lattice BGK direct numerical simulation of fully developed turbulence in incompressible plan channel flow, Comput. Fluids 35 (2006) 1137–1153.
[42] Q. Zou, X. He, On pressure and velocity boundary conditions for the lattice Boltzmann BGK model, Phys. Fluids 9 (1997) 1591–1598.
[43] S. Chen, D. Martinez, R. Mei, On boundary conditions in lattice Boltzmann methods, Phys. Fluids 8 (1996) 2527–2536.
[44] A.J.C. Ladd, R. Verberg, Lattice-Boltzmann simulations of particle–fluid suspensions, J. Stat. Phys. 104 (2001) 1191–1251.
[45] Z. Xia, K.W. Connington, S. Rapaka, et al., Flow patterns in the sedimentation of an elliptical particle, J. Fluid Mech. 625 (2009) 249–272.
[46] B. Wen, H. Li, C. Zhang, et al., Lattice-type-dependent momentum exchange method for moving boundaries, Phys. Rev. E 85 (2012) 016704.
[47] X. He, G. Doolen, Lattice Boltzmann method on curvilinear coordinates system: Flow around a circular cylinder, J. Comput. Phys. 134 (1997) 306–315.
[48] H. Li, H. Fang, Z. Lin, et al., Lattice Boltzmann simulation on particle suspension in a two-dimensional symmetric stenotic artery, Phys. Rev. E 69 (2004) 031919.
[49] B.H. Yang, J. Wang, D.D. Joseph, et al., Migration of a sphere in tube flow, J. Fluid Mech. 540 (2005) 109–131.
[50] L.-P. Wang, C. Peng, Z. Guo, Z.S. Yu, Lattice Boltzmann simulation of particle-laden turbulent channel flow, Comput. Fluids (2015) in press. http://dx.doi.org/10.1016/j.compfluid.2015.07.008.
[51] B. Milici, M.D. Marchis, G. Sardina, et al., Effects of roughness on particle dynamics in turbulent channel flows: a DNS analysis, J. Fluid Mech. 739 (2014) 465–478.
[52] A.L. Fogelson, C.S. Peskin, A fast numerical method for solving the three-dimensional Stokes' equations in the presence of suspended particles, J. Comput. Phys. 79 (1988) 50–69.
[53] Z.G. Feng, E.E. Michaelides, The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems, J. Comput. Phys. 195 (2004) 602–628.