Computers and Mathematics with Applications 65 (2013) 194–210

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

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

Lattice Boltzmann simulation of turbulent flow laden with finite-size particles

Hui Gao, Hui Li, Lian-Ping Wang *

Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA

ARTICLE INFO

Keywords:
Particle-laden flow
Turbulence
Lattice Boltzmann equation
Finite-size effect
Particle–particle interactions
Particle-resolved simulation

ABSTRACT

The paper describes a particle-resolved simulation method for turbulent flow laden with finite size particles. The method is based on the multiple-relaxation-time lattice Boltzmann equation. The no-slip boundary condition on the moving particle boundaries is handled by a second-order interpolated bounce-back scheme. The populations at a newly converted fluid lattice node are constructed by the equilibrium distribution with non-equilibrium corrections. MPI implementation details are described and the resulting code is found to be computationally efficient with a good scalability. The method is first validated using unsteady sedimentation of a single particle and sedimentation of a random suspension. It is then applied to a decaying isotropic turbulence laden with particles of Kolmogorov to Taylor microscale sizes. At a given particle volume fraction, the dynamics of the particle-laden flow is found to depend mainly on the effective particle surface area and particle Stokes number. The presence of finite-size inertial particles enhances dissipation at small scales while reducing kinetic energy at large scales. This is in accordance with related studies. The normalized pivot wavenumber is found to not only depend on the particle size, but also on the ratio of particle size to flow scales and particle-to-fluid density ratio.

1. Introduction

Turbulent flows laden with finite-size particles (solid particles, liquid droplets, or air bubbles) are relevant to a wide variety of natural processes and engineering applications, such as aerosol and pollutant transport, dust storms, sediment transport, interaction of cloud droplets, plankton dynamics, spay combustion, and chemical processes. Over the past two decades, advanced computational and experimental methods have been developed and utilized to quantify the interactions between the dispersed solid particles and the carrier fluid phase. Since turbulent particle-laden flows usually encompass a wide range of length and time scales, it is computationally demanding to simultaneously resolve both the carrier-phase turbulent flow and the disturbance flows due to the particles. Most studies typically employed the point–particle model [1–7], namely, the particle size is not explicitly considered. Within the point–particle model, the motion of an individual particle is modeled by an equation of motion [8] and only the scales of motion that are much larger than the particle size are directly resolved. This approach implicitly assumes that the particle size is much smaller than the Kolmogorov length scale of the carrier fluid turbulence and particle Reynolds number is small so the disturbance flow around a particle is a Stokes flow. The point–particle model has been used by many researchers to study preferential concentration of inertial particles [2,9], turbulent modulation by inertial particles [1,3], turbulent collision rate of inertial particles [4–6,10,11]. The point–particle model is computationally efficient, but its validity is often questionable when particle mass loading is significant or particles form aggregates yielding strong multiscale couplings between the particulate phase and the carrier phase.

* Corresponding author. Tel.: +1 302 831 8160; fax: +1 302 831 3619.
E-mail addresses: hgao@UDel.Edu (H. Gao), lwang@udel.edu (L.-P. Wang).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.06.028

Table 1
Particle-resolved simulations of turbulent particle-laden flows.

Chronological	Method	Physical issues studied
Ten Cate et al. [25]	LBM	Turbulent modulation, particle–particle hydrodynamic interaction & collision. Forced.
Burton & Eaton [28]	Finite volume/overset grid	Turbulent flow on a single fixed particle: dissipation rate and kinetic energy as a function of distance from the particle surface; force acting on the particle. Decaying.
Zhang & Prosperetti [22]	Finite-difference/Stokes flow exp.	Turbulence modulation and force on particle. Decaying.
Uhlmann [18]	LBM with IBM	Turbulent suspension in a vertical channel.
Yeo et al. [29]	Force coupling method	Turbulent modulation by particles and bubbles; Lagrangian statistics. Forced.
Lucci et al. [30]	Finite-difference with IBM	Turbulent modulation; local variation around particle; energy spectra. Decaying.
Naso & Prosperetti [31]	Finite-difference/Stokes flow exp.	Turbulent flow on a single fixed particle. Forced.
Homann & Bec [32]	Pseudo-penalization method	A single neutrally buoyant particles in a forced turbulent flow.

In many engineering applications, however, particle size can be of the same order as or larger than Kolmogorov scale, the scales contained in the disturbance flows then overlap with the scales of motion in the carrier turbulence. In this situation, the point–particle model is no longer a valid description and the finite-size effect of the dispersed phase must be resolved together with the carrier fluid turbulence. Furthermore, in concentrated suspensions such as fluidized beds, particle–particle collisions can also have a significant impact on the rheology of the system.

The motivation of this work is to understand the motion and hydrodynamic interactions of finite-size inertial particles suspended in a turbulent flow. One of the major challenges is to develop an efficient and accurate approach to resolve the disturbance flows around particles suspended in a turbulent carrier fluid.

In recent years, several computational methods have been proposed along this direction. Finite element methods [12,13] employ body-fitted mesh to implement solid particle boundary conditions. The frequent mesh regeneration due to the geometry change as a result of particle motion is computationally expensive, especially in three-dimensional simulations. Therefore, methods using a fixed and structured grid have received more attention in recent years. The fictitious domain method [14,15] applies a field of Lagrange multipliers to enforce constraints on the particle so that the fluid inside the particle domain is forced to mimic rigid body motion. The immersed boundary method (IBM) [16–18] realizes the no-slip boundary condition on particle surface by imposing a localized forcing field. Similarly, the force coupling method [19] uses smoothed body force field to represent the effects of particles on the fluid phase, in terms of low order force multiple expansion. The hybrid method Physalis [20–22] handles the no-slip boundary condition by coupling an analytical Stokes expansion valid in a narrow but finite region near a particle surface with the numerical solution outside the particle.

In contrast to the above macroscopic CFD approaches based on the Navier–Stokes equations, in the mesoscopic lattice Boltzmann method (LBM) [23–25] the fluid field is realized through local moments of a lattice Boltzmann equation on a uniform lattice grid. The no-slip boundary condition can be imposed by using a simple interpolated bounce-back scheme. To reduce force oscillations on the particles, the immersed-boundary-lattice-Boltzmann method (IB-LBM) [26,27] has also been developed by replacing the conventional bounce-back scheme with a direct forcing scheme applied on a set of Lagrangian boundary points representing particle surfaces.

In recent years, many of these particle resolved simulation methods have been applied to study particle–particle interaction or particulate suspension in a non-turbulent fluid [33–36]. There have also been a few studies of particle-resolved simulation of turbulent particle-laden flows as summarized in Table 1; in this case, the challenge is the multiscale nature involving a wide range of scales from the integral scale of the background turbulence to the scales of disturbance flows around each particle. For this reason, particle resolved simulation of turbulent particle-laden flows requires the state-of-the-art supercomputers. The interaction of a single fixed or moving particle with a turbulent flow has been studied by Burton and Eaton [28], Naso and Prosperetti [31] and Homann and Bec [32]. These studies provided a better understanding on the nature of flow modulation near the surface of the particle due to the finite-size effect that depends on the particle Reynolds number.

Ten Cate et al. [25] conducted a particle-resolved simulation of forced turbulent flows laden with solid particles using LBM. The carrier–fluid turbulence was maintained at Taylor microscale Reynolds number $Re_\lambda = 61$ using a spectral forcing scheme. The no-slip boundary condition on the particle surface was implemented by applying a body force field to the fluid domain. A subgrid lubrication force was computed based on relative location and velocities of approaching particles [33]. The simulation considered up to 3868 particles, corresponding to volume fractions ranging from 2% to 10%. The density ratio between the solid and fluid phase ranges from 1.15 to 1.73. The particle diameter is set to 8 grid spacing in the simulations. It was demonstrated that particle–fluid relative motion results in an enhancement of kinetic energy $E(k)$ and energy dissipation rate $\varepsilon(k)$ at large wave numbers and a reduction of both spectra at small wave numbers. Oscillations in $E(k)$ were found at large wave numbers, although they are negligible when compared to the total kinetic energy of the system.

Zhang and Prosperetti [22] developed the hybrid method Physalis and demonstrated its capability in simulating decaying particle-laden turbulence. A periodic cubic domain of 64^3 contains 100 spherical particles with radius equal to 39% of the initial Taylor microscale. Particle–fluid density ratio was set to 1.02. Gravity was neglected and particle–particle short range

interactions were represented by an elastic model. The Taylor microscale Reynolds number changed from 29 to 14 during the time interval of the simulation. It was reported that, compared with the point–particle model, the finite-size particles decrease the kinetic energy, show less diffusion in terms of mean particle displacement, and exhibit a stronger tendency of clustering.

Uhlmann [18] considered particulate suspensions in a vertical turbulent channel flow, using the immersed boundary method to treat the solid particles. The incompressible Navier–Stokes equations were solved by a fractional-step scheme on a staggered grid. Up to 4096 particles were simulated in a turbulent flow sustained at a bulk flow Reynolds number of 2700. The particle diameter was set to approximately 11 wall units, corresponding to a volume fraction of 0.42%. The density ratio varied from 2.2 to 10. It was found that the presence of particles induced large scale streaklike velocity perturbations, although no significant aggregation of particles was observed.

Using the force coupling method, Yeo et al. [29] studied turbulence modulation by finite-size particles and bubbles. The flow was forced at large scales. They demonstrated that the pivot wavenumber characterizing the transition from damped to enhanced energy content is mainly a finite-size effect of particles and that the transition scale is almost independent of the particle to fluid density ratio.

Lucci et al. [30] applied the immersed boundary method to study modulation of decaying turbulence by solid particles of size comparable to the Taylor microscale. The carrier flow had an initial Taylor Reynolds number of 75. Density ratio varied from 2.56 to 10.0. Volume fraction of the dispersed phase was 10% or less, and a maximum number of 6400 particles were considered in their simulations. They studied turbulence modulation by particles and structure of disturbance flow around the particles.

The main objectives of this paper are to describe our own implementation of LBM using interpolated bounce back and to present our preliminary results. The simulation method is described in Section 2. In Section 3, we first validate our code using unsteady sedimentation of a single particle. We then apply the method to study statistics in a random suspension at finite particle Reynolds numbers and compare our results with those reported in [36]. The main results concern decaying turbulent flows laden with finite-size particles. We show that our method can reproduce the results in [30]. A key new physical result discussed in Section 3 is the dependence of normalized pivot wavenumber on the dimensionless particle size. Timing and scalability of the code are also briefly discussed. In Section 5, we conclude the paper with a summary and future outlook.

2. Simulation method

In this section, we first introduce the lattice Boltzmann approach aimed at simulating a turbulent flow laden with finite-size particles. Due to the high computational demand, the code needs to be parallelized to run on a scalable computer such that results can be obtained in a timely manner. Hence the MPI implementation of the code represents an important effort. We will describe the general strategy and some specific details related to the MPI implementation in order to achieve a good computational efficiency.

2.1. Lattice Boltzmann simulation

In this study, the mesoscopic lattice Boltzmann approach, based on the multiple-relaxation-time (MRT) lattice Boltzmann equation (LBE) [37], is applied to simulate decaying homogeneous isotropic turbulence laden with finite size spherical particles in a three-dimensional periodic domain. In the MRT-LBE, the mesoscopic particle distribution function at a fluid lattice point is governed by

$$\mathbf{f}(\mathbf{x}+\mathbf{e}_\alpha\delta t, t+\delta t) = \mathbf{f}(\mathbf{x}, t) - \mathbf{M}^{-1}\cdot\mathbf{S}\cdot[\mathbf{m}-\mathbf{m}^{(\mathrm{eq})}], \tag{1}$$

where $\mathbf{M}$ is an orthogonal transformation matrix converting the distribution function $\mathbf{f}$ from discrete velocity space to the moment space $\mathbf{m}$, in which the collision relaxation is performed. The transformation between the particle velocity space and the moment space is given as

$$\mathbf{m} = \mathbf{M}\cdot\mathbf{f}, \qquad \mathbf{f} = \mathbf{M}^{-1}\cdot\mathbf{m}. \tag{2}$$

The diagonal relaxation matrix $\mathbf{S}$ specifies the relaxation rates for the non-conserved moments. In this work, the D3Q19 model is utilized with the discrete velocities ordered as

$$\mathbf{e}_\alpha = \begin{cases} (0,0,0), & \alpha = 0, \\ (\pm1,0,0),\ (0,\pm1,0),\ (0,0,\pm1), & \alpha = 1-6, \\ (\pm1,\pm1,0),\ (\pm1,0,\pm1),\ (0,\pm1,\pm1), & \alpha = 7-18. \end{cases} \tag{3}$$

The corresponding 19×19 transform matrix $\mathbf{M}$, the 19 components of moment $\mathbf{m}$ and its equilibrium counterpart $\mathbf{m}^{(\mathrm{eq})}$, and the relaxation matrix $\mathbf{S}$ are described in detail in [37].

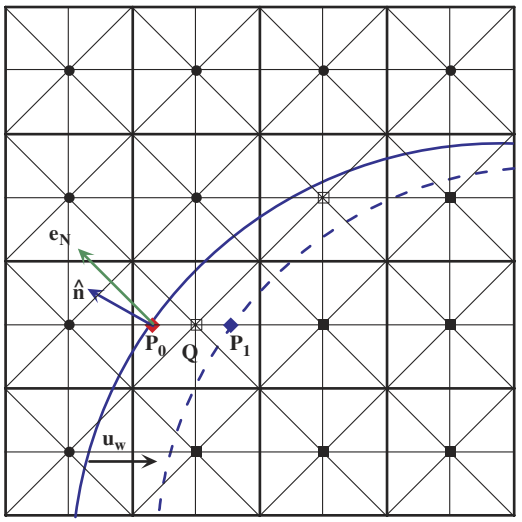

Fig. 1. A two-dimensional sketch illustrates how the lattice velocity direction $\mathbf{e}_N$ is defined, for the purpose of implementing the non-equilibrium correction. The filled circles denote fluid lattice nodes at the beginning of a time step, filled squares are solid lattice nodes, and the open squares represent newly created fluid lattice nodes. The thick curve represents the solid surface at the beginning of the time step, and the dashed curve the new particle surface at the end of the time step.

The macroscopic hydrodynamic variables, including density, momentum, and pressure, are obtained from the moments of the mesoscopic distribution function $\mathbf{f}$, namely,

$$\rho = \rho_0 + \delta\rho, \quad \rho_0 = 1 \tag{4}$$

$$\delta\rho = \sum_\alpha f_\alpha, \qquad \rho_0 \mathbf{u} = \sum_\alpha f_\alpha \mathbf{e}_\alpha, \qquad p = \delta\rho c_s^2 \tag{5}$$

where $\mathbf{u}$ is the macroscopic fluid velocity, and the sound speed c_s is equal to $1/\sqrt{3}$ in lattice units.

The computation domain is covered with a uniform cubic lattice. The periodic boundary condition is applied in all three directions. The no-slip boundary condition on solid particle surfaces are implemented using a second-order interpolated bounce-back scheme [38]. First, for each fluid node near a particle surface, all the links moving into the surface of a solid particle are identified, and the boundary-cutting location on the link is calculated in terms of the percentage (q) of the link outside the surface. For better numerical stability, when $q \leq 0.5$, the interpolation is performed before streaming, while for $q > 0.5$, it is done after streaming [38]. When two particles are in close contact, two fluid lattice nodes may not be available near the missing population to allow for a second-order interpolation. In this case, a linear interpolation or a simple on the node bounce back is used instead. For a moving solid boundary, the boundary links have to be identified and q has to be updated for each time step.

In our implementation, the above LBE evolution only applies to the fluid nodes where the particle distribution functions are defined. No distribution function is defined for solid lattice nodes lying within the particles. When a particle is moved, a solid node may move out of the solid region and become a fluid node with unknown distribution functions. In this work, all the 19 particle distribution functions for the new fluid node are constructed by an equilibrium distribution plus a non-equilibrium correction [39], namely,

$$f_\alpha(\mathbf{x}) = f_\alpha^{(eq)}(\mathbf{x}; \mathbf{u}_w, \bar{\rho}) + f_\alpha^{(neq)}(\mathbf{x} + \mathbf{e}_N) \tag{6}$$

where $\mathbf{u}_w$ is the velocity of the moving boundary at the new fluid node Q (Fig. 1) using the center, translation and angular velocity of the solid particle at the end of time step, $\bar{\rho}$ is the fluid density averaged over all the existing fluid nodes found in the immediate neighborhood using the density value based on the latest distribution functions. On extremely rare occasion when multiple solid particles are almost in contact, no single neighborhood fluid node is available; in this case, $\bar{\rho}$ is set to ρ_0. Since the magnitude of $\mathbf{u}_w$ is assumed to be much smaller than the sound speed c_s, in reality, the three points P_0, Q, and P_1 in Fig. 1 are very close to one another. While the equilibrium part is calculated with $\mathbf{u}_w$ and $\bar{\rho}$ at $\mathbf{x}$, the non-equilibrium part is obtained from a neighboring node at $\mathbf{x} + \mathbf{e}_N$. Here the $\mathbf{e}_N$ is a specified discrete velocity along which direction the quantity $\hat{\mathbf{n}} \cdot \mathbf{e}_N$ takes the maximum value (Fig. 1), where $\hat{\mathbf{n}}$ the unit normal vector pointing outwards of the moving boundary at the point P_0 through which the solid node Q at the beginning of a time step will cross the particle surface to enter the fluid domain. The equilibrium population is calculated as

$$f_\alpha^{(eq)}(\mathbf{x}, t) = W_\alpha \left[\rho + \frac{\rho_0 \mathbf{e}_\alpha \cdot \mathbf{u}}{{c_s}^2} + \frac{\rho_0 \mathbf{u}\mathbf{u} : (\mathbf{e}_\alpha \mathbf{e}_\alpha - {c_s}^2 \mathbf{I})}{2{c_s}^4} \right] \tag{7}$$

where $\mathbf{I}$ is the identity matrix, and the weight W_α is given as

$$W_\alpha = \begin{cases} 1/3, & \alpha = 0, \\ 1/18, & \alpha = 1-6, \\ 1/36, & \alpha = 7-18. \end{cases} \tag{8}$$

Our numerical tests suggested that the refilling technique as given by Eq. (6) yielded smaller force oscillations when compared to other methods such as the extrapolation method used in [38].

The hydrodynamic force $\mathbf{F}_i$ acting on the ith particle is calculated during the interpolated bounce-back procedure by summing up the loss of fluid momentum on all the links across the surface of the ith particle, and the torque $\boldsymbol{\Gamma}_i$ acting on the ith particle is the sum of the cross product of the local position vector relative to the center of the particle and the loss of fluid momentum, over all boundary links. This ensures that the net loss (or gain) of momentum by the fluid lattice particles is exactly equal to the gain (or loss) of momentum of the solid particles, so the total momentum of the system is conserved. This represents a unique advantage of LBM as it avoids spatial differentiations that would be needed for computing the net hydrodynamic force and torque on each solid particle in terms of local stress distribution on the particle surface, for Navier–Stokes based CFD methods.

In case particles are in close contact, the flow in the narrow gap between the particles is no longer fully resolved. The physical lubrication force, which is proportional to the radial relative velocity and inversely proportional to the gap distance, cannot be directly simulated. One could add, by empirical tuning, the portion of physical lubrication force that is not resolved to the portion that is directly resolved, as was done in [33]. Since the lubrication force changes rapidly with the gap distance and a finite time step must be used, a more effective approach to prevent particles from overlap due to under-estimation of the physical lubrication force is to introduce a soft-sphere like short-range repulsive force. In this preliminary study, we simply adopt the following pair-wise repulsive force acting on the ith particle due to its interaction with jth particle [15,27]

$$\mathbf{F}_{ij} = \begin{cases} 0, & r_{ij} > R_{ij} + \zeta, \\ \dfrac{c_{ij}}{\varepsilon_p}\left(\dfrac{r_{ij} - R_{ij} - \zeta}{\zeta}\right)^2 \left(\dfrac{\mathbf{r}_{ij}}{r_{ij}}\right), & r_{ij} \leqslant R_{ij} + \zeta \end{cases} \tag{9}$$

where $\mathbf{r}_{ij} \equiv \mathbf{Y}_i - \mathbf{Y}_j$, $r_{ij} \equiv \|\mathbf{Y}_i - \mathbf{Y}_j\|$, $R_{ij} \equiv R_i + R_j$, c_{ij} is a force scale and is set to be the buoyancy force in this study. $\mathbf{Y}_i$ and $\mathbf{Y}_j$ represent the center location of the ith and jth particle with radius of R_i and R_j, respectively. ζ is a threshold gap distance within which the model repulsive force becomes active. In this work, ζ is set as two lattice spacing. The stiffness parameter, ε_p, is set to be small enough to prevent particles from overlapping.

At each time step, with the resolved hydrodynamic force, repulsive force, and torque acting on particles readily obtained, the particle translational and angular velocities, center position and angular displacement are updated using the Crank–Nicolson scheme for particle position and angular displacement integration and the forward Euler scheme for velocity and angular velocity (with force and torque being averaged over the two half time steps to reduce fluctuations),

$$\mathbf{V}_i^{t+\delta t} = \mathbf{V}_i^t + \frac{1}{M_p}\left[\frac{\mathbf{F}_i^{t+\delta t/2} + \mathbf{F}_i^{t-\delta t/2}}{2} + \sum_j \mathbf{F}_{ij}^t\right]\delta t, \tag{10}$$

$$\boldsymbol{\Omega}_i^{t+\delta t} = \boldsymbol{\Omega}_i^t + \frac{1}{I_p}\left(\frac{\boldsymbol{\Gamma}_i^{t+\delta t/2} + \boldsymbol{\Gamma}_i^{t-\delta t/2}}{2}\right)\delta t, \tag{11}$$

$$\mathbf{Y}_i^{t+\delta t} = \mathbf{Y}_i^t + \frac{1}{2}\left(\mathbf{V}_i^t + \mathbf{V}_i^{t+\delta t}\right)\delta t, \tag{12}$$

$$\boldsymbol{\Theta}_i^{t+\delta t} = \boldsymbol{\Theta}_i^t + \frac{1}{2}\left(\boldsymbol{\Omega}_i^t + \boldsymbol{\Omega}_i^{t+\delta t}\right)\delta t \tag{13}$$

where M_p and $I_p \equiv \frac{2}{5}M_p R_i^2$ are the mass and moment of inertia of the ith particle.

To summarize, for each time step, the sequence of operations is as follows: (1) MRT collision step, (2) interpolation of populations needed for bounce back on the solid particle surfaces and calculations of force and torque acting on each solid particle, (3) streaming, (4) computation of solid particle short-range repulsive interaction forces, (5) updating solid particle locations and velocities, (6) updating lattice links cutting through solid particle surfaces, (7) refilling populations for the new fluid nodes, (8) updating hydrodynamic variables, and (9) updating fluid flow and solid particle statistics.

2.2. MPI implementation

The computational domain or the periodic box is decomposed in one of the spatial directions, e.g., the z direction, as shown in Fig. 2, producing a sequential set of slabs of dimensions (nx, ny, nz/n_{proc}), where nx, ny, and nz are the domain size in units of lattice spacing in the x, y, z direction, respectively, and n_{proc} is the number of processes employed. Each of these slabs is mapped to an individual process according to its sequential number, and this one-to-one mapping is fixed during

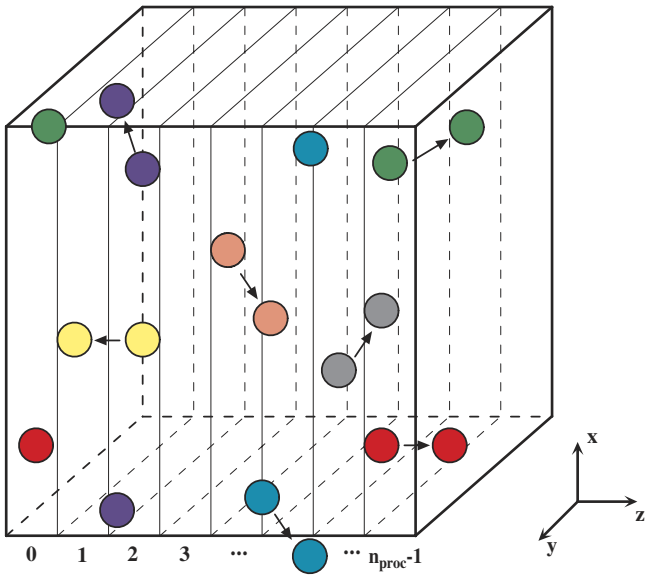

Fig. 2. 1D domain decomposition and periodic boundary condition manifested by particle movement.

the computation. For example, process 0 handles the spatial region of slab 0, process 1 handles the region of slab 1, and so on. Since the 2nd-order interpolated bounce-back scheme may utilize the information of two adjacent layers of fluid lattice nodes in the z direction to construct the missing population at a lattice node in the immediate vicinity of a solid surface, the minimum slab size has to be 2 lattice units in order to limit the data communication to neighboring slabs only. This implies that, for the particle-laden flow simulation, the maximum number of processes that can be utilized is $nz/2$ for our one-dimensional domain decomposition approach. Multiple-dimensional domain decomposition should be explored in the future to allow the use of a much larger number of processes.

The initial positions for the freely-moving particles are generated by a master process, say, process 0, via a random number generator. The use of a fixed seed for random number generation on only the master process ensures the repeatability of the initial positions of the particles. The particles are then distributed to different processes according to the z component of their center locations. If the center of a particle is found to be moved to its neighboring slab at the end of a time step, all other data related to this particle including its global index, center position, angular displacement, force and torque acting on the particle, and translational and angular accelerations, are transferred together to the neighboring slab. In addition to the global particle index, a local particle index is also employed within each slab, to accelerate certain operations involving particles belonging to the same process, for example, the calculation of particle–particle repulsive force, the update of particle dynamic evolution, and the re-ordering of particles in a slab after some new particles moving in and some old particle leaving the slab.

Comparing with the point–particle based simulation, one of the most challenging issues for finite-size particles is on the comprehensive and efficient handling of the scenario when a particle is crossing the computational domain boundary. For instance, as illustrated in a 2D schematic shown in Fig. 3, a particle could be close to a corner of the cubic box, thus its volume could be distributed into eight portions (four in 2D), each of which residing in one of the corners due to the periodic boundary conditions applied in all directions. Clearly each portion of the particle can have either a real center located inside or virtual particle centers located outside the computational domain. Since LBM requires a lattice node to be classified as a fluid node within the fluid region or a solid node inside a solid particle, it is necessary to calculate the minimum distance from the solid particle center to the lattice node of interest. A straightforward method is to calculate and compare all eight distance from the lattice node to the real center and seven virtual centers, but it would be cumbersome.

A much more efficient scheme to find this minimum distance is as follows. First, the x coordinate of the lattice node x_{node} is compared with the x-component of the position of the real particle center x_{part}, which is located within the computational domain. The following three-step procedure allows this minimum distance to be computed. First, if $x_{part} - x_{node} > nx/2$, then the replacement of x_{part} by a virtual x center at $x_{part} - nx$ would provide the minimum separation in magnitude in the x direction. Second, if $x_{part} - x_{node} < -nx/2$, then the replacement of x_{part} by a virtual x center at $x_{part} + nx$ would provide the minimum separation in the x direction. Third, if $-nx/2 \leq x_{part} - x_{node} \leq nx/2$, the use of a real particle center is adequate. The essence of this operation is to ensure that the magnitude of the separation in the x direction is no larger than $nx/2$. This procedure is repeated in the y and z directions, together they provide an efficient method to compute the minimum distance of a lattice node to the particle center, requiring only one-time calculation of projected distance in each spatial direction instead of eight times. Another benefit of this algorithm is that it can handle any particle size relative to the slab width, as long as the particle diameter is less than half of the computational domain size.

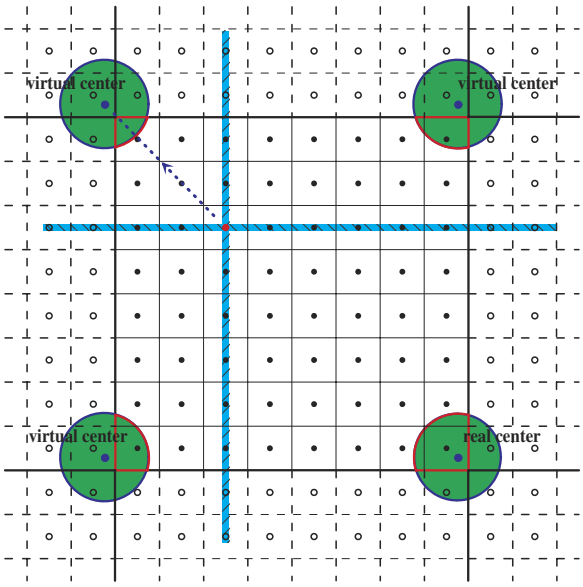

Fig. 3. A 2D schematic used to illustrate the essential concept of the efficient scheme for computing the minimum distance of a lattice node to the center of a solid particle, regardless where the solid particle is located. In this figure, the center of the solid particle is located close the lower right corner of the domain, representing one of the most challenging situations. The three periodic images of the solid particle are also shown, with virtual centers located outside the domain near the other three corners. The red dot represents any lattice node of interest. By simply comparing the coordinate of the real particle center and that of the node, one can divide the domain into four quadrants (in 2D) and then easily find the nearest particle center to the lattice node of interest, and thus the shortest particle-to-node distance. In the figure, the nearest particle center to the red dot is the one located in the upper left quadrants. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

This algorithm has been applied extensively and consistently in the code to reduce computational effort, such as the detection of particle overlap, construction of the lattice links crossing the solid particle boundary, calculation of the interpolated bounce-back on particle surfaces, particle–particle repulsive force, and refilling of the missing populations for the newly-generated fluid lattice nodes.

It is a common practice in an MPI code to have variables localized as much as possible to reduce the memory usage and consequently the time to access memory, which would lead to a higher computational efficiency. However, there exist circumstances that a global variable is preferred, in order to save communication time in between processes and to simplify the code structure.

In this code, a small number of solid particle arrays are set to global arrays, including the array of particle center position, translational and angular velocity, and particle–particle repulsive force. This treatment introduces redundancy and a small impact on data communication, but the extra memory requirement is almost negligible when compared to other large local arrays already allocated in the same process. For example, suppose the the code is set to contain 1 million particles and run at grid resolution of 1024^3 with 512 processes, thus the local distribution function has a dimension of $f(19, 1024, 1024, 2)$, and the global array of particle position is $yp(3, 1M)$. With double precision, the former alone would occupy memory of $19 \times 1024 \times 1024 \times 2 \times 8/1024^2 = 304$ MB, while the latter consumes $3 \times 1M \times 8/1024^2 = 24$ MB, which has only a marginal effect on the total memory usage, but eliminates the need for data communication.

Since the dispersed phase can move freely in the carrier flow, the boundary links and the boundary cutting location q on each link must be updated at every time step. To identify if a fluid lattice node possesses boundary links, one needs to calculate the distance between each particle center and the node of interest. Typically with thousands of particles and millions of nodes available, the task of updating distances could be computationally intensive, if one attempts to blindly compare each node to each individual particle directly.

A solution to this problem is to only consider the fluid lattice nodes surrounding the particle surface in a confined region, e.g., a small cubic box with a lateral length of particle diameter plus four lattice spacings, which would encompass all the potential fluid nodes with links crossing particle boundary. All the other nodes outside this range will be filtered out in a nested conditional screening as follows. First, check the x-coordinate of the node of concern, keep the node for the next level of screening if its x coordinate falls within the truncation region, otherwise drop it and turn to the next node in the x direction. This procedure is repeated in the y, and then in the z direction. This nested screening ensures that only the lattice nodes in a minimal region are considered in the search of boundary links for a given solid particle. Finally, the distances between all selected lattice nodes and the center of the solid particle are computed to identify all boundary links on the solid particle. For 256^3 lattice resolution and 6400 particles of radius of 4, tests have shown that the computational efficiency for this boundary link search algorithm is faster than the unscreened algorithm, by a factor of more than two thousand.

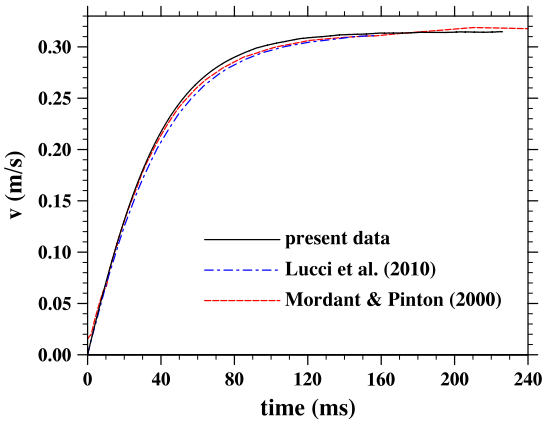

Fig. 4. Time evolution of the velocity of a small steel bead settling under gravity in a large water tank.

In summary, the above MPI implementation details yielded an overall efficient MPI code for particle-resolved simulation of turbulent particle-laden flows, as will be demonstrated by timing and scalability data to be presented in the next section.

3. Results

In this section, we will first present results from code validation simulations involving a single particle and a random suspension in a non-turbulent background carrier flow. The results for a decaying turbulent particle-laden flow will then be discussed to demonstrate the capability and computational efficiency of the approach. Preliminary results concerning turbulence modulation by finite-size solid particles will be provided to reveal the effects of particles on the energy spectrum of the turbulent flow and on the spatial distribution of dissipation rate around a solid particle.

3.1. Validation: single particle settling

Mordant and Pinton [40] measured the velocity of a steel bead settling under gravity (9.8 m/s^2) in a large water tank. The steel bead was released from rest. Here we consider one of their measured cases with the diameter of the bead equal to 0.8 mm. The particle density is $\rho_p = 7710$ kg/m^3. The water density and kinematic viscosity are $\rho_f = 1000$ kg/m^3 and $\nu = 0.9 \times 10^{-6}$ m^2/s. In the LBM simulation, we set the particle diameter to $d = 8$ (in lattice units). The system is governed by two dimensionless parameters: the solid particle to fluid density ratio, ρ_p/ρ_f, and the dimensionless gravity gd^3/ν^3, where g is the gravitational acceleration. These two dimensionless parameters in the LBM simulation are matched with those in the experiment. A domain size of $12.5d \times 12.5d \times 128d$ (or $100 \times 100 \times 1024$ lattice units) is used. Fig. 4 compares the simulated settling velocity with the measured curve. Also shown is the result from Lucci et al. [30] based on a finite-difference immersed-boundary scheme. In both simulations, periodic boundary conditions are assumed in all three directions. All results are in excellent agreement. The steady-state settling velocity obtained in the LBM is 0.315 m/s in physical units, which is very close to 0.316 m/s measured in [40]. This validates our LBM particle-resolved simulation code.

To be complete, we note that, for a free sphere in an unbounded spatial domain, the wake can become unstable. In a spectral simulation of a uniform flow around a sphere in a large spatial domain, Bagchi et al. [41] found steady and axisymmetric flow when the particle Reynolds number is less than 210, steady and nonaxisymmetric flow without vortex shedding when the Reynolds number is between 210 and 270, and unsteady three-dimensional flow with vortex shedding when the Reynolds number is above 270. In our simulation, the final particle Reynolds number is 280. However, we applied periodic boundary conditions in all three directions in a finite domain, so it is not a free sphere in an unbounded domain. We suspect that the particular setup we used delayed the development of non-axisymmetric and unsteady wakes. A closer inspection showed that the particle did start to rotate towards the end of the simulation with the total angular displacements of 8.6°, 1.76°, and 0.66°, respectively, along the x, y, and z axes at the end of the simulation ($t = 0.24$ s), where x and y are the horizontal directions and z is the vertical direction. The particle was also found to drift slowly in the y direction with a total translational displacement of about one particle diameter at $t = 0.24$ s. A non-axisymmetric wake was also observed near the end of the simulation. It could be possible that an unsteady wake will develop if the simulation were to continue for a much longer time than what is presented in Fig. 4.

3.2. Sedimentation of a random suspension

To further validate our implementation of LBM, we first consider a random suspension at finite particle Reynolds numbers and compare our results with those of [36]. Solid spherical particles were randomly seeded in an initially quiescent

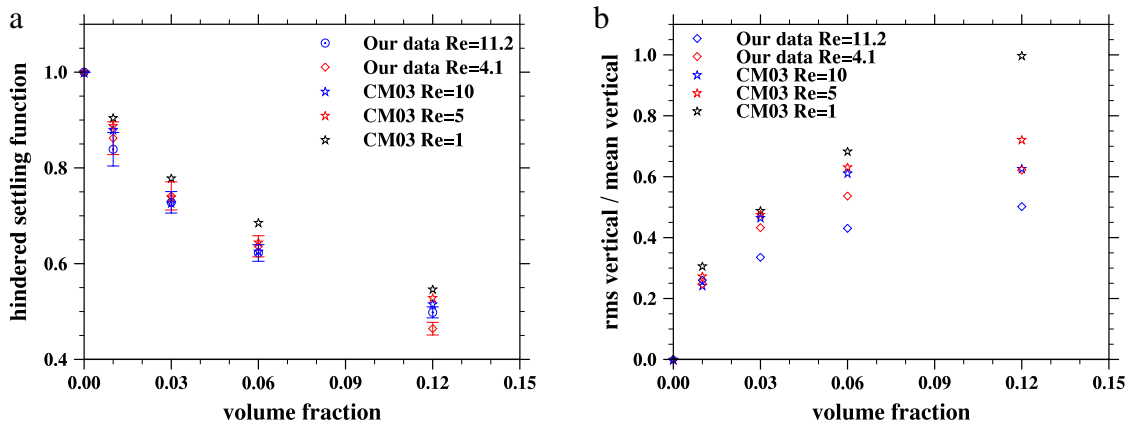

Fig. 5. (a) Average settling velocity normalized by V_0 in 128^3 simulations: comparing the present study with [36]. (b) Fluctuation of settling velocity normalized by its mean value, V_{rms}/V_{mean}, in 128^3 simulations: comparing the present study with [36].

fluid domain of size 128^3. Periodic boundary conditions were applied in all three directions. A series of simulations were conducted to yield five different volume concentrations, using 1, 40, 120, 240, 480 particles of a fixed radius of 5 grid spacing. The single-particle case was used as the reference case. A volumetric body force was applied to balance the excess weight of the particles in the vertical direction. The gravitational acceleration was varied to give two different particle Reynolds numbers of roughly 4.1 and 11.2, based on the single-particle sedimentation velocity and particle diameter. The sedimentation velocity was corrected to ensure that the system has a zero net vertical mass flux. In a Stokes suspension flow, this additional correction would not be needed. However, in a suspension at finite particle Reynolds number, the system is no longer a linear system, a simple balance of the body force and the excess weight cannot guarantee a net zero mass flux, due to nonlinear fluid inertial effects.

An important question for a random suspension is how the mean velocity varies with the particulate volume concentration. Fig. 5(a) shows the mean sedimentation velocity normalized by V_0, the terminal velocity of a single particle sedimenting in the same periodic domain. The estimated statistical uncertainties of our own data are also plotted. This is known as the hindered settling function and it is plotted as a function of particulate volume fraction. Also shown are results from [36] at particle Reynolds numbers of 1, 5, and 10. Clearly, the particle average settling velocity is significantly reduced as its volume fraction is increased, and the larger the particle Reynolds number the smaller the settling velocity. The results are in good agreement with those of [36], with our simulations predict a somewhat smaller settling velocity. The differences could be due to two reasons. First, there are statistical uncertainties in both studies. Second, the force coupling method is an approximate method in which the disturbance flow is not fully resolved.

The relative vertical velocity fluctuations V_{rms}/V_{mean} are shown in Fig. 5(b). The overall trends are similar: the relative vertical velocity fluctuation increases with the volume fraction, but decreases with the particle Reynolds number. Quantitatively, our simulations yield smaller values. The origin for the quantitative differences between our simulations and those of [36] remains to be studied.

3.3. Decaying particle-laden turbulence

Prior to the simulation of particle-laden turbulence, the accuracy of our LBM code was verified by a comparison with a pseudo-spectral code for particle-free decaying homogeneous isotropic turbulence. The particle-free or single-phase flow will be referred to as the Case 1 flow. For most simulations, we consider a periodic domain of 256^3 grid points, indicating the potential range of wave number $1 \leqslant k \leqslant 128$. Following the work of Lucci et al. [30], the initial velocity field at $t = 0$ was specified by a Gaussian field with a prescribed kinetic energy spectrum as

$$E(k) = \left(\frac{3u_0^2}{2}\right)\left(\frac{k}{k_p^2}\right)\exp\left(-\frac{k}{k_p}\right), \tag{14}$$

where k is the wave number, k_p is the wave number containing peak energy, and u_0 is the initial r.m.s. velocity. Both k and k_p have been normalized by the minimum wavenumber $k_0 \equiv 2\pi/L_B$, where L_B is the computational domain size.

Here we performed two simulations of the single-phase decaying flow, one using the LBM, and the second using the golden-standard pseudo-spectral method [42]. The settings of the simulation parameters are shown in Table 2. To separate the time scales of acoustic waves from hydrodynamics, we used a small velocity scale in LBM, so the lattice time is much smaller than the hydrodynamic time scales. For example, the Kolmogorov time scale at the initial time is equal to 117 lattice time units. The relative velocity, length, and time scale ratios are listed Table 2 which can be used to convert flow statistics from one to the another. Note also that the effective time step sizes used in the different approaches are very similar.

Table 2
Parameter settings in our LBM code and in the finite difference simulation of Lucci et al. [30] for the single-phase decaying turbulence. Here L_B is the computational domain size, $k_0 = 2\pi/L_B$, and Δx is the grid spacing.

	Grid	L_B	u_0	k_p/k_0	ν	$dt \cdot u_0/\Delta x$	$\frac{U}{U_{Lucci}}$	$\frac{L}{L_{Lucci}}$	$\frac{T}{T_{Lucci}}$
LBM	256^3	256	0.020494	4	2.4094×10^{-3}	0.02049	0.407437	256	628.32
PS	256^3	2π	0.0503	4	1.4514×10^{-4}	0.02049	1.0	6.2832	6.2832
Lucci et al. [30]	256^3	1.0	0.0503	4	2.31×10^{-5}	0.02515	1.0	1.0	1.0

It can be shown theoretically that this initial energy spectrum implies the following initial flow statistics:

$$\frac{3}{2}u_{rms}^2 \equiv \int_0^\infty E(k)\mathrm{d}k = \frac{3}{2}u_0^2 \quad \text{or} \quad \frac{u_{rms}}{u_0} = 1.0, \tag{15}$$

$$\frac{\epsilon L_B}{u_0^3} \equiv \frac{2\nu L_B}{u_0^3}\int_0^\infty k^2 E(k)\mathrm{d}k = 72\pi^2 k_p^2 \frac{\nu}{u_0 L_B}, \tag{16}$$

$$\frac{L_f}{L_B} \equiv \frac{\pi}{2u_0^2 L_B}\int_0^\infty \frac{E(k)}{k}\mathrm{d}k = \frac{3}{8k_p} \tag{17}$$

$$\frac{\lambda}{L_B} \equiv \frac{1}{L_B}\sqrt{\frac{15\nu u_0^2}{\epsilon}} = \frac{1}{2\pi k_p}\sqrt{\frac{5}{6}} \tag{18}$$

$$\frac{\eta}{L_B} \equiv \frac{1}{L_B}\left(\frac{\nu^3}{\epsilon}\right)^{1/4} = \sqrt{\frac{15\nu u_0^2}{\epsilon}} = \sqrt{\frac{1}{6\pi\sqrt{2}}\frac{1}{k_p}\frac{\nu}{u_0 L_B}} \tag{19}$$

$$R_\lambda \equiv \frac{u_{rms}\lambda}{\nu} = \frac{1}{2\pi k_p}\sqrt{\frac{5}{6}}\frac{u_0 L_B}{\nu} \tag{20}$$

where u_{rms} is the realized component r.m.s. fluctuation velocity, ν is the fluid kinematic viscosity, ϵ is the viscous dissipation rate, L_f is the longitudinal velocity correlation length, λ is the transverse Taylor microscale, η is the Kolmogorov length, and R_λ is the Taylor microscale Reynolds number. It is noted that the peak wavenumber k_p plays an important role in determining the initial flow dissipation rate, all flow length scales, and the flow Reynolds number.

In Table 3, we compare non-dimensional flow statistics for the Case 1 simulation, namely, the single-phase decaying turbulence. At $t = 0$, the above theoretical predictions are also listed. In the pseudo-spectral (PS) simulation, the energy spectrum is truncated at $k = N/3$. For the initial spectrum given by Eq. (14), it an be shown theoretically that this truncation has no noticeable effect on the kinetic energy and dissipation rate. The relative differences between the pseudo-spectral results and the theoretical values are typically about 1% or less, except in L_f where it is 11.6%. These minor differences are due to the discretization effect in the wave vector space. Since the integral length L_f is mostly determined by the large scale, the mode discretization has a significant effect. In our LBM, an identical initial flow as in the pseudo-spectral simulation was used. The initial populations $\mathbf{f}(\mathbf{x}, t)$ were iterated following the procedure described in [43] to achieve a consistent initial pressure field; this amounts to solving a pressure Poisson equation iteratively. It is clear that all flow statistics from the LBM simulation are in excellent agreement with those from the PS simulation. Almost identical kinetic energy and dissipation rate spectra between the LBM and SP simulations were also observed (not shown here). These provide a direct validation of our LBM simulation and are consistent with the results shown in [42].

The flow statistics obtained by Lucci et al. [30], also shown in Table 3, however, have some inconsistencies even at the initial time. For example, the dissipation rate is larger than the theoretical value by about 11.3% and the Taylor microscale is less by 5% when compared to the theoretical value. Part of these inconsistencies could be a result of different random numbers used in the flow initialization. Since our simulated flow statistics are much closer to the theoretical values at $t = 0$, we suspect there are some other problems in Lucci et al. [30], either in their low-order finite-difference scheme or in their processing of flow statistics. Overall, the data in Table 3 imply that the numerical method used in Lucci et al. [30] has a noticeable numerical dissipation at earlier times, while our LBM simulation has negligible numerical dissipation. We shall return to this point again when we compare time evolutions of turbulence kinetic energy and dissipation rate.

For the particle-laden turbulence simulations, the single-phase flow field was evolved for sometime until a converged velocity-derivative skewness of about -0.50 was developed. At this moment, particles were released in the domain at random locations with a gap distance of at least two grid spacing in between. Initial translational velocities of particles are set as the instantaneous fluid velocity at the particle center, and angular velocity is set to zero. A set of five simulations, listed in Table 4, were preformed to study the turbulence modulation by finite-size particles. Cases 2, 3, and 4 roughly match Cases D, E, and G in [30]. The gravity was set to zero. The simulations are run from $t = 0$ to $t = 2.12T_{e,0}$, corresponding to 5000 time steps, where the initial eddy turnover time $T_{e,0} \equiv u_{rms}^2/\epsilon$ and it is 2356 lattice time units. The flow fields are

Table 3
Comparison of simulated flow statistics for single phase turbulence. LBM and SP denote our LBM simulation and our pseudo-spectral simulation. The simulation by Lucci et al. [30] is a second-order finite-difference simulation.

	u_{rms}/u_0	$\epsilon L_B/u_0^3$	L_f/L_B	λ/L_B	η/L_B	R_λ	R_{L_f}	R_{L_B}
$t/T_{Lucci}=0$								
Theoretical	1.00	5.22	0.09375	0.03632	0.002075	79.1	204	2177.5
LBM	0.997	5.27	0.0830	0.0361	0.00207	78.3	180	2169
PS	0.997	5.28	0.0829	0.0360	0.00207	78.3	180	2171
Lucci et al. [30]	1.00	5.81	0.0684	0.0345	0.00202	75.	149	2177
$t/T_{Lucci}=1$								
LBM	0.886	6.919	0.0858	0.0280	0.00193	53.9	165	1926
PS	0.884	6.963	0.0860	0.0278	0.00193	53.5	165	1924
Lucci et al. [30]	0.867	7.70	0.0685	0.0259	0.00188	49	129	1888
$t/T_{Lucci}=5$								
LBM	0.506	1.688	0.108	0.0323	0.00275	35.6	119	1104
PS	0.503	1.668	0.115	0.0323	0.00276	35.4	126	1096
Lucci et al. [30]	0.463	1.57	0.0891	0.0305	0.00280	31	89.8	1078

Table 4
Parameters for particle-laden flow simulations at the release time.

Case	N^3	d	ρ_p/ρ_f	N_p	d/η	d/λ	ϕ_v	ϕ_m	τ_p/τ_k
2	256^3	8.0	2.56	6,400	16.1	1.1	0.10	0.23	36.8
3	256^3	8.0	5.0	6,400	16.1	1.1	0.10	0.36	71.9
3H	512^3	16.0	5.0	6,400	16.1	1.1	0.10	0.36	71.9
4	256^3	11.0	2.56	2,304	22.1	1.5	0.10	0.21	69.6
5	512^3	8.0	5.0	51,200	8.08	0.559	0.10	0.36	18.1

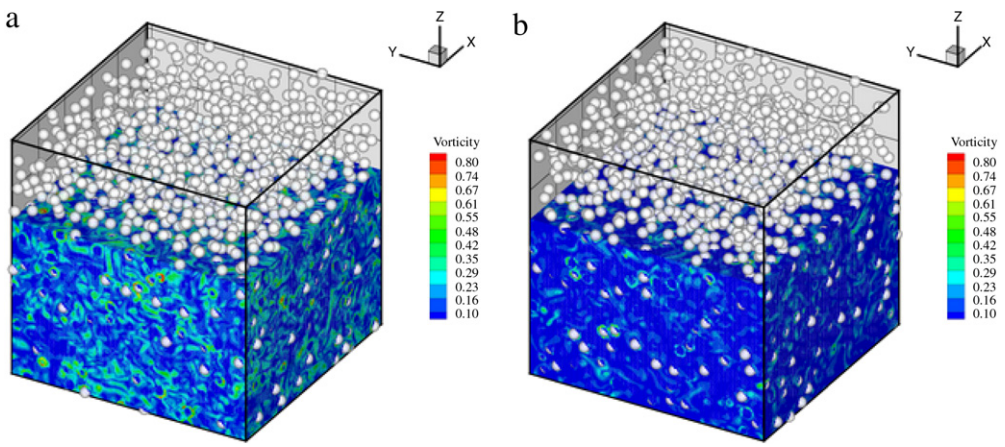

Fig. 6. Vorticity contour at (a) $1.27T_{e,0}$ and (b) at $2.12T_{e,0}$ for case 4. A layer of fluid is removed to show some of the 2304 particles.

well resolved as shown by $k_{max}\eta > 1$ for the whole time interval. All the simulations were performed on an IBM Power6 supercomputer at NCAR with MPI parallelization using 32 processors (at 256^3) to 128 processors (at 512^3). Each simulation takes 6 to 27 h of wall-clock time.

Fig. 6 displays the vorticity contour for Case 4 on the boundaries of the computational domain at 3000 lattice time step ($1.27T_{e,0}$) and 5000 lattice time step ($2.12T_{e,0}$), respectively. A layer of fluid to the top is removed to show the locations of a portion of the 2304 particles. The magnitude of vorticity decreases with time as expected, as can be seen from the color difference of Fig. 6(a) and (b).

In Fig. 7 we provide 2D visualizations of the vorticity magnitude on a slice and locations of the particles cutting through the same slice, for three different times. These are results from the Case 3 run at $z = 128.5$ (a $x-y$ slice near the center of the domain). It is observed that the presence of particles is often associated with high voritcity values (the red spots), suggesting that motion of finite size particles can generate small-scale flow structures near its surface. Note that Case 3 has the largest density ratio and likely the largest slip velocity. We found that the enhanced activities at small scales due to particles are self evident when compared to the particle-free case as shown in Fig. 7(c) and Fig. 7(d). The vortical structures are larger in size and more interconnected in the single-phase flow case. There is very little correlation between the single-phase and two-phase vorticity fields at a given time. At the earlier time $t = 0.522T_{e,0}$, both the particles and the fluid have large kinetic energy and as such there is a large slip velocity between a particle and its neighboring flow, the disturbance near the

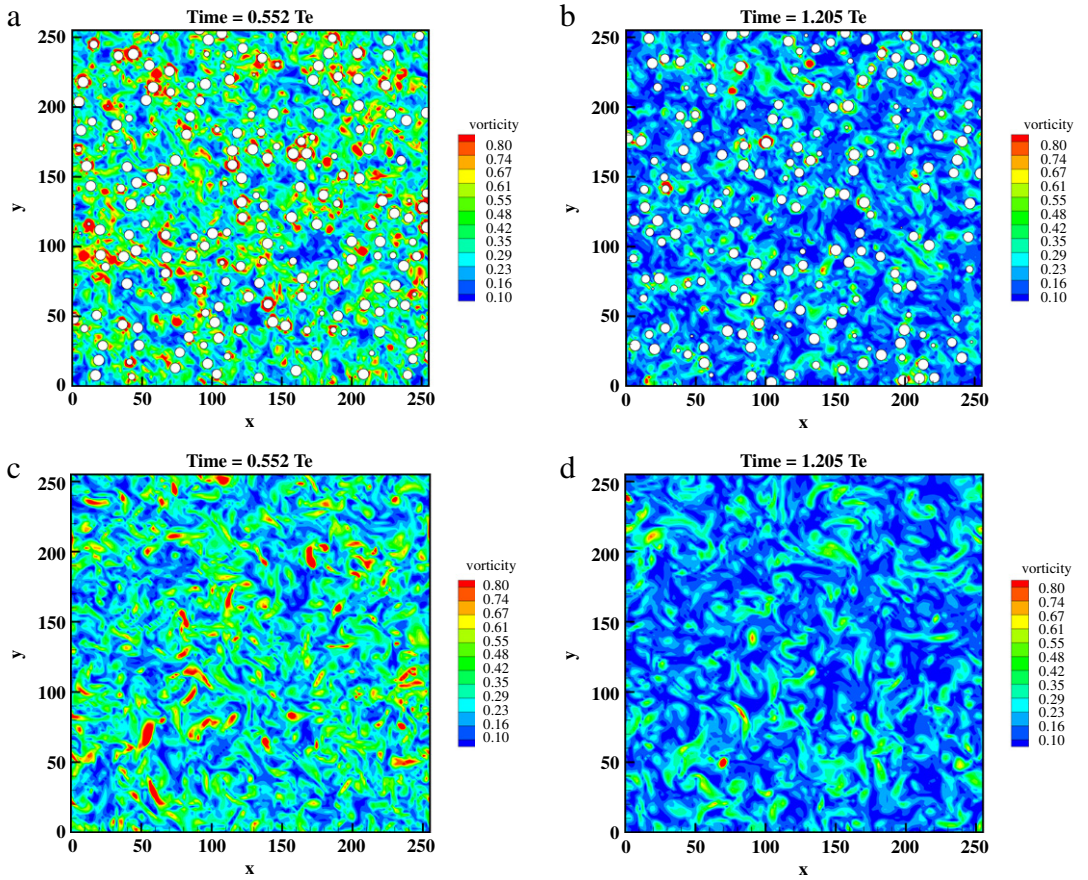

Fig. 7. Snapshots of vorticity contour and particle location on a plane-cut of $z = 128.5$ in the 256^3 simulation of case 3: (a) $0.55T_{e,0}$ and (b) $1.21T_{e,0}$. Note that the presence of particles is often associated with high vorticity values (represented by the colors towards the red end), indicating relatively larger dissipation near particle surfaces. The corresponding vorticity contours for single-phase flow at the two times are shown in (c) and (d), respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

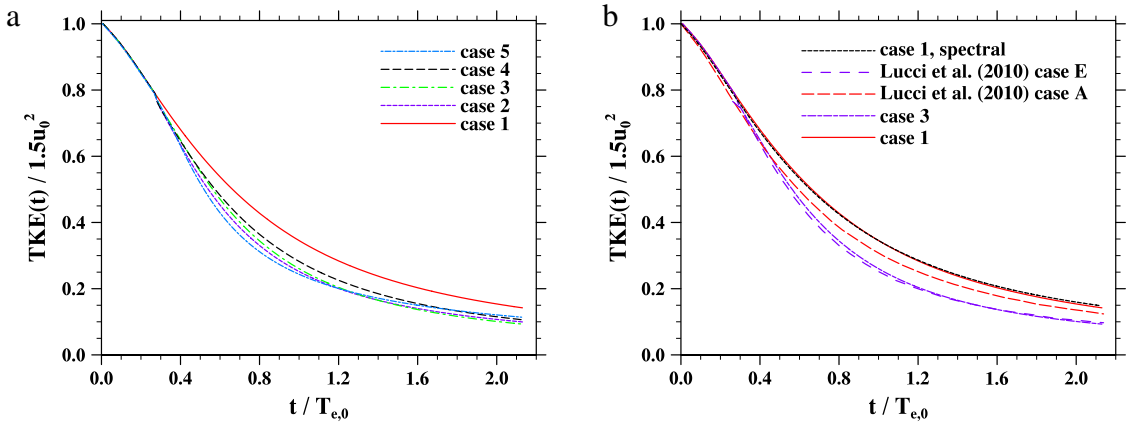

Fig. 8. Temporal evolution of turbulent kinetic energy normalized by its initial value. (a) Different cases simulated here; (b) Comparison with the results of Lucci et al. [30] and PS (for single-phase turbulence only).

particle surface is more visible. At later times, the system has less and less kinetic energy and interphase slip velocity, the disturbance flow near the particles are weaker. The effect of particles on the viscous dissipation thus changes with time, as will be shown in Fig. 9 below.

Fig. 8(a) shows the temporal evolution of turbulent kinetic energy (TKE) normalized by $1.5u_0^2$ computed at time $t = 0$ for all the five cases. It should be noted that the kinetic energy $TKE^{full}(t)$ we computed is the kinetic energy for the full field,

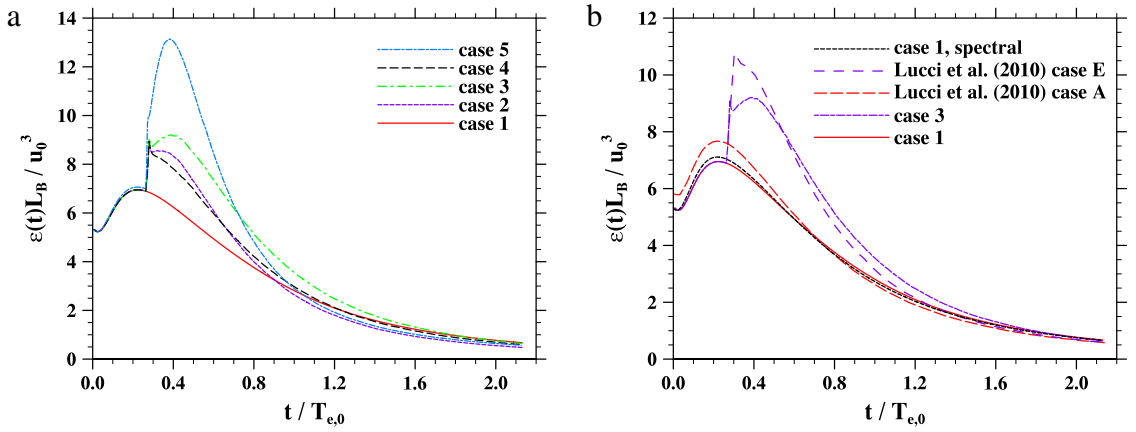

Fig. 9. Temporal evolution of turbulent dissipation rate normalized by its initial value. (a) Different cases simulated here; (b) Comparison with the results of Lucci et al. [30] and PS (for single-phase turbulence only).

including the space occupied by the particles where a rigid body translation and rotation together define the local velocity. The average kinetic energy of the fluid domain, $TKE^{fluid}(t)$, is related to $TKE^{full}(t)$ as

$$TKE^{full}(t) = TKE^{fluid}(t)(1-\phi_v) + \phi_v\left(\frac{1}{2}\langle V_p^2\rangle + \frac{1}{20}d^2\langle \Omega_p^2\rangle\right), \tag{21}$$

Compared with the particle-free turbulence, the TKE decreases faster in the particle-laden cases from the moment of particle release till about $1.6T_{e,0}$, and then the decaying rate tends to be the same in all cases. In addition, the increase of particle number results in an evident reduction of TKE, as shown by comparing the curves from Case 2 and Case 4. This may be associated with more surface areas in Case 2 than in Case 4. The increase of particle density has a marginal effect on decaying rate reduction, based on a comparison of Case 2 with Case 3.

In Fig. 8(b), we compare our results with the results of Lucci et al. [30] and PS (for single-phase turbulence only). Note that Case 1 and Case 3 of our LBM simulations correspond to the parameter setting of Case A and Case E in Lucci et al. [30]. Our LBM and PS results for the single-phase flow overlap, but the curve from Lucci et al. [30] for the single phase flow shows a faster decay of kinetic energy due to their larger viscous dissipation at the early times, as has already been noted in Table 3 (also see in Fig. 9(b) below). The larger viscous dissipation for the single phase flow simulation in [30] could be partially due to the numerical dissipation in the second-order finite-difference scheme that Lucci et al. used. Our result for Case 3 is more closer to the result of Case E in [30]. It should be noted that our data represent $TKE^{full}(t)$ while the result of Lucci et al. is for $TKE^{fluid}(t)$, so the comparison is not a very direct comparison. The good comparison is somewhat surprising as the results for even the single-phase flow case do not match. It may be possible that the regularized three-point delta function used in the immersed boundary implementation in [30] is a local filter that somewhat smooths out large velocity gradient at the particle–fluid boundaries, reducing slightly the local dissipation rate near the particle surface. The numerical dissipation in a low order finite-difference scheme and the regularization filtering may happen to cancel one another, giving results comparable to what observed in our LBM scheme. For these reasons, we believe that our results for both the single-phase and the particle-laden flow are likely to be more accurate than those of Lucci et al. [30].

Fig. 9(a) shows the modification of turbulent dissipation rate due to the presence of particles. The local dissipation inside the particles is zero, so the whole field dissipation and the fluid–region averaged dissipation are related by

$$\epsilon^{full}(t) = (1-\phi_v)\epsilon^{fluid}(t). \tag{22}$$

We computed $\epsilon^{full}(t)$ directly and then divided by $(1-\phi_v)$ to obtain $\epsilon^{fluid}(t)$ which is what shown in Fig. 9. At the moment of about $0.26T_{e,0}$, the injection of particles into the fluid domain causes a sudden jump of the dissipation rate, followed by a peak value at about $0.4T_{e,0}$, and then the total dissipation decreases monotonically. The total particle surface area normalized by the domain size squared is equal to $6\phi_v L/d$, implying that Case 5 has the largest effective surface areas and Case 4 has the smallest effective surface. This explains why the peak dissipation of the system after the introduction of solid particles is the largest for Case 5, and the smallest for Case 4. The two intermediate cases, Case 2 and Case 3, have the same effective surface area, but different Stokes numbers. The larger the Stokes number (Case 3 compared to Case 2), the higher the peak dissipation value due to a larger slip velocity. Another observation is that, after the peak is reached, the system dissipation rate decays faster if the particle Stokes number is smaller, indicating a faster approach to an interphase quasi-equilibrium stage of the system evolution (the later part of the time evolution shown in Fig. 8(a) and Fig. 9(a)).

Fig. 9(b) compared our LBM results of the average flow dissipation rate to SP single-phase simulation and to the results of Lucci et al. [30], in a manner similar to Fig. 8(b). Again for the single-phase flow simulation, our LBM result matches the SP result, but Lucci et al. over-predicted the flow dissipation at earlier times. The two particle-laden cases are in general

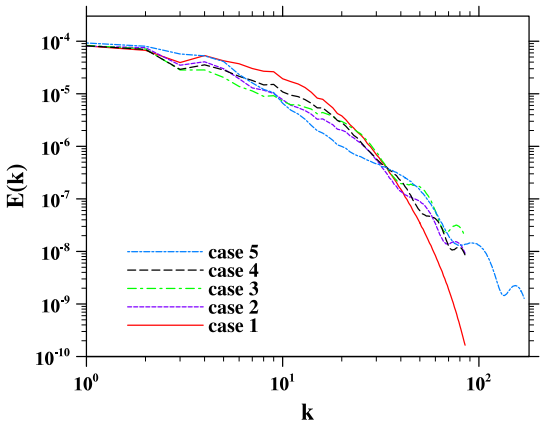

Fig. 10. Kinetic energy spectrum at the end of simulation ($2.12T_{e,0}$).

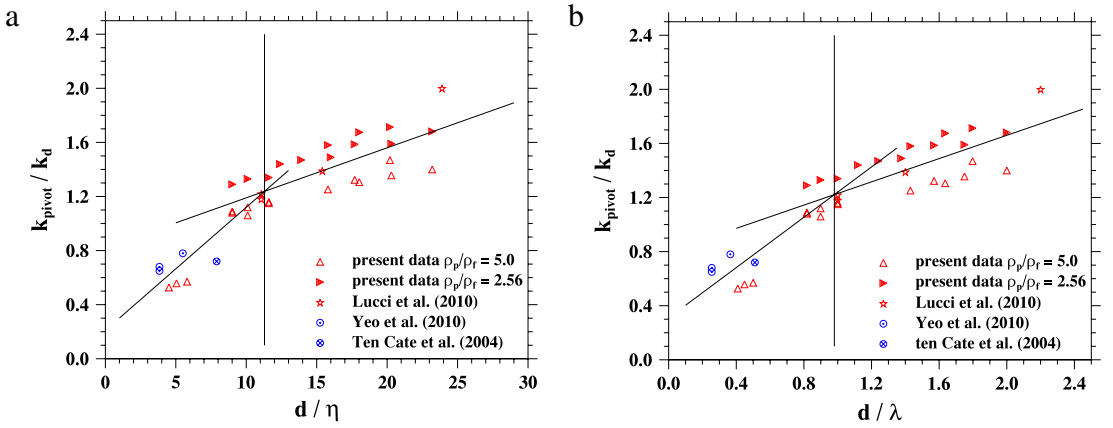

Fig. 11. Normalized pivot wavenumber as a function of particle size scaled by (a) Kolmogorov length scale, and by (b) Taylor microscale. The slopes are 0.091 and 0.037 in (a) and 0.93 and 0.43 in (b). The vertical lines are at 11.3 and 0.98 in (a) and (b), respectively.

agreement in that both have a higher flow dissipation rate when compared to the single-phase turbulence for most of the time after particle release. An important difference between our result for Case 3 and the result of Lucci et al. for Case E is that, immediately after the particle injection, the jump in the dissipation rate in Lucci et al. is much larger than in our result. As the viscous boundary layers develop on the particle surface, the LBM result shows a gradual increase in flow dissipation before reaching a second peak while the Lucci et al. shows a monotonic decay. At the moment of particle injection, there is an unphysical jump or mismatch of fluid velocity and particle surface condition. The LBM scheme is a weakly compressible formulation while Lucci et al.'s is an incompressible flow solver. The regularization function in the Lucci et al.'s immerse boundary treatment would also have a strong impact at the moment of injection. At this moment, it is not completely clear which method better handles the flow physics immediately after the unphysical jump condition. Nevertheless, the time evolutions after the transition stage ($t/T_{e,0} > 0.6$) are very similar.

An important characteristic of the particle-laden turbulent flow is the addition of motion at the scale of particle size, at the cost of reduction of large-scale fluid motion, since no energy is added to the system here. This is clearly shown in Fig. 10 where we compare the kinetic energy spectra at the end of simulations $t = 2.12T_{e,0}$. It should be noted that we use the full velocity field including the rigid-body velocity inside the particles to compute the energy spectrum. Typically, the kinetic energy is higher at higher wavenumbers when compared to the single phase flow. At intermediate and low wavenumbers, this is reversed. Interestingly, wave-like oscillations are observed at the tails of the spectra for turbulent particle-laden flow, as shown previously in [25,30] due to the discontinuity of flow velocity gradient at the finite-size particle surface. Compared to the results of Case 2 and Case 4, Case 3 possesses noticeably larger TKE at large wavenumbers due to large mass loading and large Stokes number.

The result for Case 5 is more complicated. First, the cross-over at large wavenumbers becomes very evident with a clear attenuation of energy at the intermediate wavenumbers. Second there is apparently a second cross-over at $k = 4 \sim 5$. It could be due to some large-scale structures formed by the particles. This will need to be studied in the future.

It is now well known that the cross-over wavenumber or pivot wavenumber, k_{pivot}, at the large-wave number end scales with the particle diameter. We introduce $k_d \equiv L_{box}/d$, where L_{box} is the computational domain size. Fig. 11 shows the

Table 5
The wall-clock times per time step for various runs.

Case	Grid resolution	d	N_p	ϕ_v	Wall-clock per time step (s)	Additional overhead (%)
1 (flow only)	256^3	–	–	–	3.11	–
2	256^3	8.0	6,400	0.1023	3.84	23
3	256^3	8.0	6,400	0.1023	3.85	24
4	256^3	11.0	2,304	0.0957	3.74	20
1 (flow only)	512^3	–	–	–	7.01	–
3H	512^3	16.0	6,400	0.1023	8.84	26
5	512^3	8.0	51,200	0.1023	10.1	44

dimensionless pivot wavenumber k_{pivot}/k_d as a function of nondimensional particle diameter. In Fig. 11(a), the diameter is normalized by the Kolmogorov scale of the single-phase flow, and in Fig. 11(b) by the transverse Taylor microscale of the single phase flow. Each data point represents the result at a given time during the decay. Additional simulations than those mentioned in Table 4 are performed to cover different d ranges. All results available in the studies shown in Table 1 are also shown. Our results are in general agreement with one another, although the simulations in [25,29] are obtained for forced particle-laden turbulent flows. Despite some scattering of the data points partially due to some uncertainty in reading the pivot wavenumbers, two important observations can be made. First, the dimensionless wavenumber increases with the dimensionless particle size. This is due to the different natures of the local flow: a particle of diameter of the order of Kolmogorov scale senses a more uniform local velocity gradient, while a particle much larger than the Kolmogorov scale experiences a more complex local flow field including inertia–subrange fluctuations. Another observation is that there appears to be two slopes in each of the plot, the dimensionless pivot wavenumber is more sensitive to particle size when $d/\eta < 11$ or $d/\lambda < 1$, and becomes less sensitive for large particle sizes. The transition takes place apparently at $d \sim \lambda$, as λ represents the scale of velocity gradient in the undisturbed turbulent flow. Finally, the magnitude of k_{pivot}/k_d depends on the particle-to-fluid density ratio at a given volume fraction, with denser particles leading to a smaller k_{pivot}/k_d.

3.4. Timing and scalability

Finally, we comment on the wall-clock times of the simulations, as shown in Table 5. All 256^3 runs in Table 5 used 32 processes and 512^3 runs used 128 processes. First, we note that the single phase flow simulation at 512^3 with 128 processes takes about twice the wall-clock time as the flow simulation at 256^3 with 32 processes, as expected from simple scaling of the problem size. A remarkable feature is that the particle-resolved simulations only take a moderate overhead in terms of wall-clock time, when compared to the single-phase flow simulation at the same resolution. This moderate overhead is a result of the LBM algorithm and our careful MPI implementation.

Furthermore, the overhead reported in Table 5 depends on the number of solid particles as well as the solid particle size. We could roughly partition the overhead into three parts: (1) the overhead associated with exchanging two layers of fluid–particle distribution data when compared with just one layer in single-phase flow simulation, and additional one-layer slab-to-slab communication associated with the refilling procedure; (2) the overhead for processing the boundary links on the solid particle surface which is proportional to the total solid particle surface area ($\sim N_p d^2$); and (3) the overhead associated with updating the translational and angular locations and velocities of solid particles and with MPI_ALLREDUCE operations to collect single particle information and to store in corresponding solid particle arrays. Assume the overhead associated with the third part is relatively insignificant; we can use the results in Table 5 to estimate the overheads for the first part and the second part. Note that the overhead for the first part is independent of the number of solid particles, and the overhead for the second part is proportional to $N_p d^2$. For example, by comparing Cases 2, 3, and 4, we infer that the overhead for the first part is roughly 12.5% for 256^3 simulation. The overhead for the second part for Case 2 and Case 3 is roughly 11% and for Case 4 it is roughly 7.5%. Likewise, a comparison of Case 3H and Case 5 shows the overhead for the first part is 8% for 512^3 simulations, the overhead for the second part for Case 3H and Case 5 are 18% and 36%, respectively.

We also document the scalability of the code against the number of processes used, in Fig. 12. Case 3 and Case 3H are used and the wall-clock times per time step are shown in the figure. The black line shows the ideal scalability curve of slope -1. Clearly the code running time scales well with n_{proc}.

4. Conclusions

In this paper, we have presented a particle-resolved simulation method for turbulent flow laden with finite size particles. The method was based on the multiple-relaxation-time lattice Boltzmann equation [37]. The no-slip boundary condition on the moving particles boundaries was handled by a second-order interpolated bounce-back scheme [38]. The populations at a new fluid node were constructed by equilibrium distribution with non-equilibrium correction [39]. A simple repulsive force model was utilized to prevent particle–particle overlap. The code was parallelized with MPI and was found to be computationally efficient. A number of specific MPI implementation issues have been resolved and explained. Up to 51,200 particles in 3D have been considered in our simulations. We are currently testing the code for higher grid resolution and

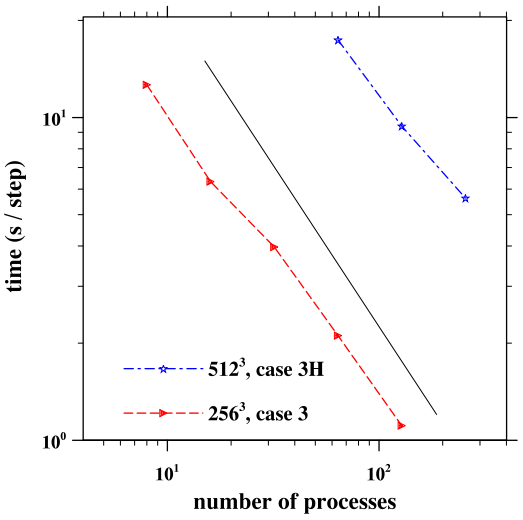

Fig. 12. Scalability: CPU time vs the number of processes for case 3 and case 3H.

larger number of particles. A remarkable feature of the approach is that the computational overhead is rather moderate even for a large number of particles in a turbulent carrier flow. We also demonstrated a good scalability of the code.

The force and torque on a particle were computed based on summing momentum exchanges at the boundary links. Apart from the refill problem, the method essentially conserves the overall momentum of the system at the mesoscopic level, which ensures the correct balance of kinetic energy of the whole system and viscous dissipation.

The code was first validated by studying the settling of a single particle under gravity, yielding excellent results compared to measured data of Mordant and Piton [40] and simulated results of Lucci et al. [30]. The method was then applied to two problems. The first is the sedimentation of a random suspension at finite particle Reynolds numbers. Results on the hindered settling function and relative particle vertical velocity fluctuation are presented in terms of particulate volume fraction and particle Reynolds numbers. These results are in reasonable agreement with the results obtained by Climent and Maxey [36] using a force coupling method.

We then applied the method to study a decaying turbulence seeded with finite-size particles. In the particle-free case, the flow statistics matched precisely the results obtained from the accurate pseudo-spectral method, consistent with the observation of [42]. Several particle-laden flow cases reported in [30] were simulated here, with qualitatively similar results. Some inconsistencies in the results of Lucci et al. [30] have been noted, implying that our LBM may have better overall accuracy than their second-order finite-difference scheme. The results show that particles of Kolmogorov to Taylor microscale sizes introduce small-scale features, enhance the dissipation rate of the system, and alter the shape of energy spectrum. The normalized pivot wavenumber depends on dimensionless particle size and this dependence has a transition at $d/\lambda \sim 1$. This implies that the pivot wavenumber depends not only on particle size, but also the relative ratio of particle size to flow dissipation range scales as well as the particle-to-fluid density ratio. At a given particulate volume fraction, the effective particle surface area and particle Stokes number play a prominent role in the dynamic evolution of overall kinetic energy, energy spectrum, and the flow dissipation rate. These results are preliminary. Further analysis and parametric study are underway.

Further analysis of the simulation results will focus on particle phase characteristics, including fluid and particle radial and transverse relative velocities, particle radial distribution function, and particle collision rates. Towards this end, the numerical method will be critically examined in order to improve the representation of short-range particle–particle interaction. Other fundamental numerical issues related to moving particles such as force oscillations and the refill problem also require further research. Multiple-dimensional domain decompositions, as was already undertaken in [44], should also be explored in view of the current trend of scalable computers having $\mathcal{O}$ (100,000) processors. Particle-laden turbulent flow is a multiscale problem that requires the state of the art computers to include all relevant scales into the simulations with realistic physical parameters.

Acknowledgments

This work was supported by the National Science Foundation (NSF) under grants OCI-0904534, ATM-0527140, ATM-0730766, and National Natural Science Foundation of China (Project Nos. 10628206 and 11072247). Computing resources are provided by National Center for Atmospheric Research through CISL-35751010, CISL-35751014, and CISL-35751015 and by University of Delaware through NSF CRI 0958512. LPW thanks Professor Martin Maxey of Brown University for his helpful discussions and comments.

References

[1] K.D. Squires, J.K. Eaton, Particle response and turbulence modification in isotropic turbulence, Phys. Fluids A 2 (1990) 1191–1203.
[2] L.-P. Wang, M.R. Maxey, Settling velocity and concentration distribution of heavy particles in homodeneous isotropic turbulence, J. Fluid Mech. 256 (1993) 27–68.
[3] S. Elghobashi, G. Truesdell, On the two-way interaction between homogeneous turbulence and dispersed solid particles. I: turbulence modification, Phys. Fluids 5 (1993) 1790–1801.
[4] S. Sundaram, L.R. Collins, Collision Statistics in an isotropic particle-laden turbulent suspension. Part 1. Direct numerical simulations, J. Fluid Mech. 335 (1997) 75–109.
[5] Y. Zhou, A.S. Wexler, L.-P. Wang, Modelling turbulent collision of bidisperse inertial particles, J. Fluid Mech. 433 (2001) 77–104.
[6] O. Ayala, W.W. Grabowski, L.-P. Wang, A hybrid approach for simulating turbulent collisions of hydrodynamically-interacting particles, J. Comput. Phys. 225 (2007) 51–73.
[7] G.D. Jin, G.-W. He, L.-P. Wang, J. Zhang, Subgrid scale fluid velocity timescale seen by inertial particles in large-eddy simulation of particle-laden turbulence, Int. J. Multiphase Flow 36 (2010) 432–437.
[8] M.R. Maxey, J.J. Riley, Equation of motion for a small rigid sphere in a nonuniform flow, Phys. Fluids 26 (1983) 883–889.
[9] K.D. Squires, J.K. Eaton, Preferential concentration of particles by turbulence, Phys. Fluids A 3 (1991) 1169–1179.
[10] L.-P. Wang, B. Rosa, H. Gao, G.W. He, G.D. Jin, Turbulent collision of inertial particles: point-particle based, hybrid simulations and beyond, Int. J. Multiphase Flow 35 (2009) 854–867.
[11] G.D. Jin, G.-W. He, L.-P. Wang, Large eddy simulation of collisional statistics of inertial particles in isotropic turbulence, Phys. Fluids. 22 (2010) 055106.
[12] H. Hu, Direct simulation of solid–liquid mixtures, Int. J. Multiphase Flow 22 (1996) 335–352.
[13] H. Hu, N.A. Patankar, M.Y. Zhu, Direct numerical simulations of solid–liquid systems using the aribitrary Lagrangian–Eulerian technique, J. Comput. Phys. 169 (2001) 427–462.
[14] N.A. Patankar, P. Singh, D. Joseph, R. Glowinski, T.W. Pan, A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 26 (2000) 1509–1524.
[15] R. Glowinski, T. Pan, T. Hesla, D. Joseph, J. Périaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow passed moving rigid bodies: application to particulate flow, J. Comput. Phys. 169 (2001) 363–426.
[16] C. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.
[17] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2005) 448–476.
[18] M. Uhlmann, Interface-resolved direct numerical simulation of vertical particulate channel flow in the turbulent regime, Phys. Fluids 20 (2008) 053305.
[19] M.R. Maxey, B.K. Patel, Localized force representations for particles sedimenting in Stokes flows, Int. J. Multiphase Flow 27 (2001) 1603–1626.
[20] S. Takagi, H. Oğuz, Z. Zhang, A. Prosperetti, Physalis: a new method for particle simulation. Part II: two-dimensional Navier–Stokes flow around cylinders, J. Comput. Phys. 187 (2003) 371–390.
[21] Z. Zhang, A. Prosperetti, A method for particle simulation, J. Appl. Mech. Trans. ASME 70 (2003) 64–74.
[22] Z. Zhang, A. Prosperetti, A second-order method for three-dimensional particle simulation, J. Comput. Phys. 210 (2005) 292–324.
[23] 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.
[24] A.J.C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation, J. Fluid Mech. 271 (1994) 285–309;
Part 2. Numerical results, J. Fluid Mech. 271 (1994) 311–339.
[25] 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.
[26] 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.
[27] Z.G. Feng, E.E. Michaelides, Proteus: a direct forcing method in the simulations of particulate flows, J. Comput. Phys. 202 (2005) 20–51.
[28] T.M. Burton, J.K. Eaton, Fully resolved simulations of particle-turbulence interaction, J. Fluid Mech. 545 (2005) 67–111.
[29] 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.
[30] F. Lucci, A. Ferrante, S. Elghobashi, Modulation of isotropic turbulence by particles of Taylor length-scale size, J. Fluid Mech. 650 (2010) 5–55.
[31] A. Naso, A. Prosperetti, The interaction between a solid particle and turbulent flow, New J. Phys. 12 (2010) 033040.
[32] H. Homann, J. Bec, Finite-size effects in the dynamics of Neutrally buoyant particles in turbulent flow, J. Fluid Mech. 651 (2010) 81–91.
[33] N.Q. Nguyen, A.J.C. Ladd, Lubrication corrections for lattice Boltzmann simulations of particle suspensions, Phys. Rev. E 66 (2002) 046708.
[34] D. Qi, Lattice-Boltzmann simulations of particles in non-zero-Reynolds-number flows, J. Fluid Mech. 385 (1999) 41–62.
[35] E.-J. Ding, C.K. Aidun, The dynamics and scaling law for particles suspended in shear flow with inertia, J. Fluid Mech. 423 (2000) 317–344.
[36] E. Climent, M.R. Maxey, Numerical simulations of random suspensions at finite Reynolds numbers, Int. J. Multiphase Flow 29 (2003) 579–601.
[37] D. d'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, L.-S. Luo, Multiple-relaxation-time lattice Boltzmann models in three-dimensions, Philos. Trans. R. Soc. Lond. A Math. Phys. Eng. Sci. 360 (2002) 437–451.
[38] P. Lallemand, L.-S. Luo, Lattice Boltzmann method for moving boundaries, J. Comput. Phys. 184 (2003) 406–421.
[39] A. Caiazzo, Analysis of lattice Boltzmann nodes initialisation in moving boundary problems, Prog. Comput. Fluid Dyn. 8 (2008) 3–10.
[40] N. Mordant, J. Pinton, Velocity measurement of a settling sphere, Eur. Phys. J. B 18 (2000) 343–352.
[41] P. Bagchi, M.Y. Ha, S. Balachandar, Direct numerical simulation of flow and heat transfer from a sphere in a uniform cross-flow, J. Fluids Eng. 123 (2001) 347–358.
[42] Y. Peng, W. Liao, L.-S. Luo, L.-P. Wang, Comparison of the lattice Boltzmann and pseudo-spectral methods for decaying turbulence. Part I. Low-order statistics, Comput. Fluids 39 (2010) 568–591.
[43] R. Mei, L.-S. Luo, P. Lallemand, D. d'Humires, Consistent initial conditions for lattice Boltzmann simulations, Computers & Fluids 35 (2006) 855–862.
[44] K. Stratford, I. Pagonabarraga, Parallel simulation of particle suspensions with the lattice Boltzmann method, Comp. Math. with Applications 55 (2008) 1585–1593.