Particuology 25 (2016) 93–103

Contents lists available at ScienceDirect

Particuology

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

Direct-forcing immersed boundary lattice Boltzmann simulation of particle/fluid interactions for spherical and non-spherical particles

A. Eshghinejadfard*, A. Abdelsamie, G. Janiga, D. Thévenin

Laboratory of Fluid Dynamics and Technical Flows, University of Magdeburg "Otto von Guericke", Universitätsplatz 2, 39106 Magdeburg, Germany

ARTICLE INFO

Article history:
Received 15 November 2014
Accepted 22 May 2015
Available online 20 August 2015

Keywords:
Immersed boundary method
Lattice Boltzmann
Particulate flow
Forcing method

ABSTRACT

The lattice Boltzmann method (LBM) is a useful technique for simulating multiphase flows and modeling complex physics. Specifically, we use LBM combined with a direct-forcing (DF) immersed boundary (IB) method to simulate fluid–particle interactions in two-phase particulate flows. Two grids are used in the simulation: a fixed uniform Eulerian grid for the fluid phase and a Lagrangian grid that is attached to and moves with the immersed particles. Forces are calculated at each Lagrangian point. To exchange numerical information between the two grids, discrete delta functions are used. The resulting DF IB-LBM approach is then successfully applied to a variety of reference flows, namely the sedimentation of one and two circular particles in a vertical channel, the sedimentation of one or two spheres in an enclosure, and a neutrally buoyant prolate spheroid in a Couette flow. This last application proves that the developed approach can be used also for non-spherical particles. The three forcing schemes and the different factors affecting the simulation (added mass effect, corrected radius) are also discussed.

Introduction

The transport of suspended solid particles in a fluid appears in many processes associated with filtration, crystallization, pollution control, blood clogging, microfluidic devices, medical applications, and the food industry (Ashraf Ali, Janiga, Temmel, Seidel-Morgenstern, & Thévenin, 2013). Therefore, an accurate and efficient modeling of a flow interacting with moving particles is of high importance. In the advanced applications of computational fluid dynamics (CFD), the mesoscopic approach has received particular attention in recent years, specifically in connection with the lattice Boltzmann method (LBM). This method has been used to analyze a variety of multiphase and multicomponent flows (Bogner & Rüde, 2013; Dietzel, Ernst, & Sommerfeld, 2011; Ernst, Dietzel, & Sommerfeld, 2013; Shan & Chen, 1993; Yu & Fan, 2010); it does not solve the macroscopic conservation equations directly, but rather models the interactions of some fictitious particles statistically using the Boltzmann equation. These particles, which are assigned properties capturing fluid attributes, are allowed to move between lattice nodes or stay at rest. The macroscopic flow properties such as density, velocity, and pressure can be retrieved from the collective behavior of the microscopic states of the particles including their location and velocity. Therefore, the LBM calculates the evolution of the particle distribution function. For several applications, it is a promising alternative to classical solvers based on finite differences, finite volumes, finite elements, or spectral methods (Sukop & Thorne, 2006; Succi, 2001). However, because it is based on regular Cartesian grids, it fails to directly simulate curved boundaries. The simplest way to represent a curved boundary is the bounce-back rule, which is however only of first-order accuracy. This method was applied to the motion of solid particles, see in particular Ladd (1994a, 1994b). To improve the accuracy of the bounce-back method, different schemes have been proposed. One approach, the immersed boundary method (IBM), is a non-boundary fitted method. The IBM was first introduced by Peskin (1972, 1977) to model the blood flow in the heart. In the IBM, the fluid equations are discretized on a fixed Eulerian grid over the entire domain and the immersed boundary is discretized on a moving Lagrangian mesh. As both IBM and LBM are based on a Cartesian grid, a combination can be readily applied to simulations of moving boundary problems. This combination is referred to as IB-LBM in the following. In the IBM, the force density is evaluated at each Lagrangian point using either the penalty method (Peskin, 1977), the momentum exchange method (Niu, Shu, Chew, & Peng, 2006), or the direct forcing (DF) method (Mohd-Yusof, 1997; Uhlmann, 2005). Feng and Michaelides (2004) first developed an IB-LBM coupled model and applied it to simulate the sedimentation of a large number of particles in an enclosure. In

* Corresponding author. Tel.: +49 3916712324.
E-mail address: amir.eshghinejadfard@ovgu.de (A. Eshghinejadfard).

http://dx.doi.org/10.1016/j.partic.2015.05.004
1674-2001/

their approach (penalty method), the particle boundary was treated as a deformable medium with high stiffness. This method has a drawback in that it requires the *a priori* selection of the stiffness parameter, based on the specific problem to be solved. They developed later the DF IB-LBM model (Feng & Michaelides, 2005) based on an earlier proposal by Mohd-Yusof (1997) for fixed complex boundaries. In the DF method, the force density term is naturally determined in the calculation process and there is therefore no need to consider the free parameter determining the stiffness coefficient, making this method much more efficient. Niu et al. (2006) proposed an IB-LBM called the momentum-exchange-based IB-LBM. Dupuis, Chatelain, and Koumoutsakos (2008) proposed a DF IB-LBM without solving the Navier–Stokes equations (NSE) for the evaluation of boundary force density. Their method is considered as a pure DF IB-LBM.

IB-LBM has been applied in recent years to an increasing variety of flow conditions. Kang and Hassan (2011) used a DF IB-LBM based on the split-forcing LBM with various interface schemes for flow problems with stationary complex boundaries. They suggested the IB-LBM with a sharp interface scheme for complex stationary boundary problems but stated that for moving boundary problems, the sharp interface scheme can introduce spurious oscillations. Therefore, for moving boundary problems, the diffuse interface scheme is more suitable because it produces a smooth evolution of forcing points. In the diffuse interface scheme, forcing points are located on the immersed object boundary.

Suzuki and Inamuro (2011) investigated the internal mass effect for various particle Reynolds numbers through the IB-LBM simulations of a moving body in a fluid. They found that the internal mass effect is fairly small for Reynolds numbers about 1, but grows as the Reynolds number increases. The effect becomes distinct for a Reynolds number over 10. Recently, the authors proposed a LBM combined with a higher-order IBM using a smooth velocity field near boundaries (Suzuki & Inamuro, 2013), to expand the velocity field smoothly into the body domain across the boundary.

The literature on particulate flow deals mostly with circular (2D) and spherical (3D) particles. Because real particles of interest are very often non-spherical, particle shape is an interesting aspect to study in regard to the flow behavior of particulate suspensions. Although some researchers have previously investigated non-spherical particles (Aidun, Lu, & Ding, 1998; Ding & Aidun, 2000; Huang, Yang, Krafczyk, & Lu, 2012; Mao & Alexeev, 2014; Rosén, Lundell, & Aidun, 2014), the application of IB-LBM to such particle geometries poses many challenges.

In this paper, the development of a DF IB-LBM method suitable for particulate flows is described. The comparisons involve circular and elliptical particles in 2D flows, and spherical and spheroidal particles in 3D flows. The paper is organized as follows. First, the governing equations of LBM and DF IBM together with the discretized equations are presented. Simulation results of moving particles for different 2D and 3D flows of increasing complexity are then discussed. After this, we provide concluding remarks.

Model formulation

Lattice Boltzmann method

The LBM focuses on the evolution of a discretized particle distribution function, $f(\mathbf{x}, t, \xi)$, which represents the probability of finding a particle in a certain location $\mathbf{x}$ with a certain velocity ξ at a certain time t. In contrast to the mass conservation and NSE, which are formulated in terms of macroscale variables (velocity, pressure), the LBM operates at a mesoscopic level via the distribution functions f, which are simply summed up to obtain the macroscopic dynamics. A discretization of the Boltzmann equation in time and space, and the restriction of velocity space $\{\xi\}$ into a finite set of velocities $\{\mathbf{c}_i\}$ with which particles move around in the lattice, leads to the lattice Boltzmann equation:

$$f_i(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t) - f_i(\mathbf{x}, t) = -\frac{1}{\tau}\left[f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)\right], \quad (1)$$

where f_i is the distribution function of particles moving with velocity $\mathbf{c}_i$ and the right-hand side accounts for the single relaxation time (SRT) Bhatnagar–Gross–Krook collision term (Bhatnagar, Gross, & Krook, 1954), which represents the variation induced in the distribution function because of collisions between the fictitious particles. Here, τ is the dimensionless mean relaxation time and Δt the time step. The equilibrium distribution function $f_i^{eq}(\boldsymbol{x}, t)$ is obtained using the Taylor series expansion of the Maxwell–Boltzmann distribution function with velocity $\mathbf{u}$ up to second order. It can be expressed as:

$$f_i^{eq} = \omega_i\rho\left[1 + \frac{\mathbf{c}_i\cdot\mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i\cdot\mathbf{u})^2}{2c_s^4} - \frac{u^2}{2c_s^2}\right], \quad (2)$$

where u is velocity magnitude, ω_i the weight associated with the velocity $\mathbf{c}_i$, and the speed of sound c_s is model-dependent. The macroscopic velocity $\mathbf{u}$ in Eq. (2) must satisfy the requirement for low Mach number M, i.e., $|\mathbf{u}|/c_s = M \ll 1$. This holds as the equivalent of the CFL condition for the classical Navier–Stokes solvers.

In the present work, for the two-dimensional (2D) and three-dimensional (3D) flows, the nine-velocity model (D2Q9) and the nineteen-velocity model (D3Q19) are applied, respectively. The speed of sound in Eq. (2) is given by $c_S = 1/\sqrt{3}$. The weight coefficients are $\omega_1 = 4/9$, $\omega_2 = \cdots = \omega_5 = 1/9$, and $\omega_6 = \cdots = \omega_9 = 1/36$ for the D2Q9 model, and $\omega_1 = 1/3$, $\omega_2 = \cdots = \omega_7 = 1/18$, $\omega_8 = \cdots = \omega_{19} = 1/36$ for the D3Q19 model. The speeds with which the corresponding distribution functions f_i propagate in the D2Q9 model are $\mathbf{c}_i = \{\mathbf{c}_{ix}, \mathbf{c}_{iy}\}$ with $\mathbf{c}_{ix} = (0, 1, 0, -1, 0, 1, -1, -1, 1)$ and $\mathbf{c}_{iy} = (0, 0, 1, 0, -1, 1, 1, -1, -1)$. The D3Q19 model has velocity vectors $\mathbf{c}_i = \{\mathbf{c}_{ix}, \mathbf{c}_{iy}, \mathbf{c}_{iz}\}$, where $\mathbf{c}_{ix} = (0, 1, -1, 0, 0, 0, 0, 1, -1, 1, -1, 1, -1, 1, -1, 0, 0, 0, 0)$, $\mathbf{c}_{iy} = (0, 0, 0, 1, -1, 0, 0, 1, 1, -1, -1, 0, 0, 0, 0, 1, -1, 1, -1)$, and $\mathbf{c}_{iz} = (0, 0, 0, 0, 0, 1, -1, 0, 0, 0, 0, 1, 1, -1, -1, 1, 1, -1, -1)$ (Guo & Shu, 2013).

Applying the Chapman–Enskog multi-scale analysis, the SRT-LBM recovers the NSE with first-order accuracy in time and second-order accuracy in space, when density and velocity are defined by the 0th and 1st moments of the probability distribution function, respectively:

$$\rho = \sum_i f_i, \quad (3)$$

$$\rho\mathbf{u} = \sum_i \mathbf{c}_i f_i. \quad (4)$$

Conceptually, the SRT-LBM algorithm is implemented in two stages: the first establishing the collision of particles, which controls the relaxation toward equilibrium; and the second developing the streaming of particles in which the distribution functions are shifted to neighboring lattice cells.

$$\text{Collision}: f_i'(\mathbf{x}, t) = f_i(\mathbf{x}, t) - \frac{\Delta t}{\tau}\left[f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)\right], \quad (5)$$

$$\text{Streaming}: f_i(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t) = f_i'(\mathbf{x}, t). \quad (6)$$

The pressure, p, is determined from the equation of state:

$$p = \rho c_s^2, \quad (7)$$

and the kinematic lattice viscosity, ν, is determined using:

$$\nu = \left(\tau - \frac{1}{2}\right)c_s^2\Delta t. \quad (8)$$