Commun. Comput. Phys.
doi: 10.4208/cicp.2009.09.054

Vol. **7**, No. 4, pp. 793-812
April 2010

Particulate Flow Simulation via a Boundary Condition-Enforced Immersed Boundary-Lattice Boltzmann Scheme

J. Wu and C. Shu*

Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260.

Received 18 March 2009; Accepted (in revised version) 20 August 2009

Available online 5 November 2009

Abstract. A boundary condition-enforced immersed boundary-lattice Boltzmann method (IB-LBM) for the simulation of particulate flows is presented in this paper. In general, the immersed boundary method (IBM) utilizes a discrete set of force density to represent the effect of boundary. In the conventional IB-LBM, such force density is pre-determined, which cannot guarantee exact satisfaction of non-slip boundary condition. In this study, the force density is transferred to the unknown velocity correction which is determined by enforcing the non-slip boundary condition. For the particulate flows, accurate calculation of hydrodynamic force exerted on the boundary of particles is of great importance as it controls the motion of particles. The capability of present method for particulate flows is depicted by simulating migration of one particle in a simple shear flow and sedimentation of one particle in a box and two particles in a channel. The expected phenomena and numerical results are achieved. In addition, particle suspension in a 2D symmetric stenotic artery is also simulated.

PACS: 45.50.-j, 47.11.-j

Key words: Lattice Boltzmann method, immersed boundary method, non-slip boundary condition, particulate flow, two-dimensional.

1 Introduction

Particulate flows have wide applications in engineering such as in river sediment resuspension and transport, cell transport in arteries and veins, fluidized bed reactors. The numerical methods for particulate flows can be broadly classified into two categories: moving mesh method and fixed mesh method. In the moving mesh method, the body-fitted mesh is updated with particle motion. The boundary condition is imposed on the

*Corresponding author. *Email addresses:* mpewuj@nus.edu.sg (J. Wu), mpeshuc@nus.edu.sg (C. Shu)

particle. The greatly used moving mesh method is perhaps the arbitrary Lagrangian-Eulerian (ALE) scheme [1–3]. Using this method, various flows with different particle shapes in Newtonian and non-Newtonian fluids have been successfully investigated. On the other hand, ALE approach would consume large computational effort due to expensive regeneration of geometrically adapted mesh. In contrast, the computational mesh remains unchanged in the fixed mesh method. No re-meshing procedure in accordance to the particle motion is required.

One popular approach in the fixed mesh category is the class of distributed Lagrange multiplier/fictitious domain (DLM/FD) method which was proposed by Glowinski et al [4, 5]. The basic idea of DLM/FD method is to view the particle region as a fictitious domain. A distributed Lagrange multiplier is imposed to enforce the constraints of rigid-body motion to the fictitious fluid inside the particle. Using this method, the fluid flow containing many rigid particles was successfully simulated by Glowinski et al [4,5]. Recently, the DLM/FD method has also been extended to simulate particulate flows in non-Newtonian fluids [6, 7]. Usually, the generalized Galerkin finite element scheme is incorporated into DLM/FD method.

The immersed boundary method (IBM) may be the simplest approach in the fixed mesh category. It was first introduced by Peskin [8] to model the blood flow in the human heart. This method uses a fixed Cartesian mesh to represent fluid phase, which is composed of Eulerian points. For the boundary immersed in the fluid, a set of Lagrangian points are used to represent it. The basic idea of IBM is to treat the physical boundary as deformable with high stiffness. A small distortion of the boundary will yield the force which tends to restore the boundary into its original shape. Hence, the effect of immersed boundary is depicted by the restoring force. The balance of restoring force is distributed into the Eulerian points through discrete delta function. The Navier-Stokes (N-S) equations with a body force are solved over the whole fluid-boundary domain. This approach can be efficiently applied to simulate flows with complex geometry. Recently, Fogelson and Peskin [9] indicated that IBM could also be applied to simulate flows involving suspended particles. Since then, many variants of IBM for particulate flow simulation have been presented. Based on the idea in [9], Hfler and Schwarzer [10] proposed a finite-difference method for particle-laden flows by adding a constraint force into the N-S equations to enforce rigid particle motion. The constraint force is determined by the penalty method. Using the direct forcing scheme, which was introduced by Fadlun et al. [11], Uhlmann [12] presented an improved immersed boundary method, which greatly suppresses the force oscillations, to simulate flow around suspended rigid particles. An enhanced version of direct forcing scheme was proposed by Luo et al. [13] recently to simulate spherical particle sedimentation. A nonlinear weighted technique and boundary point classification strategy at the immersed boundary are introduced to modify the velocity near the body. Li and Lai [14] applied the immersed interface method (IIM), which introduces the jump conditions for the velocity and pressure across the interface, to simulate the fixed and moving interface problems. Compared to IBM with the first-order accurate delta function, the second-order accuracy can be obtained in IIM.

Employing the IIM, Le et al. [15] successfully simulated flows around both rigid and elastic moving objects. Recently, an extended IBM named immersed finite element method (IFEM) has been proposed [16, 17]. Applying finite element technique to both fluid and object domains, the immersed body can be handled more appropriately and accurately. One advantage of this method is to effectively model deformable objects with motion.

In most of IBM solvers, the flow field is obtained by solving the N-S equations. As an alternative computational technique to N-S solvers, the lattice Boltzmann method (LBM) has achieved a great success in simulating various fluid flows in recent years [18,19]. LBM is a particle-based numerical scheme, which studies the dynamics of fictitious particles. Basically, it has two processes: streaming and collision. The major advantage of LBM is its simplicity, easy for implementation, algebraic operation and intrinsic parallel nature. No differential equation and resultant algebraic equation system are involved in the LBM. This approach has also been adopted to study particulate flows. Such work can be found in Ladd [20, 21], Behrend [22], Aidun et al. [23], and Qi [24]. The interactions between fluid and particle are implemented through the bounce back rule in LBM.

Although the LBM utilizes a fixed Cartesian mesh in particulate flow simulation, the implementation of boundary condition is still coupled with solution of flow domain, which is similar to the moving mesh method. To ease the coupling between the boundary and the flow domain, the idea of IBM is worthwhile to be incorporated into LBM. The first attempt was made by Feng and Michaelides [25,26]. The simulation of 2D and 3D particle sedimentation was carried out. Niu et al. [27] put forward a momentum exchange-based IB-LBM to simulate flows around fixed and moving particles.

It is well known that the restoring force calculation is an important task in IB-LBM. In general, there are three ways to finish this job: the penalty method [25], the direct forcing scheme [26] and the momentum exchange method [27]. The common feature of these approaches is to compute restoring force in advance. As indicated by Shu et al. [28] recently, pre-determination of restoring force would break down the satisfaction of non-slip boundary condition on the immersed object. As a result, the mass exchange across the boundary would be induced. Eventually, the momentum exchange due to mass exchange may lead to a force error, which would influence the solution of force on the object.

To satisfy the non-slip boundary condition, we developed a boundary condition-enforced IB-LBM [29] recently. Following the idea of Shu et al. [28], the restoring force is set as unknown rather than pre-calculated. By enforcing the non-slip boundary condition, the unknown restoring force is determined. As a consequence, the flow penetration to the solid body, which is often appeared in the conventional IBM results, is avoided, and more accurate numerical results can be achieved. For the particulate flows, the motion of particles is controlled by the force and torque exerted on them. Therefore, it is necessary to accurately compute the hydrodynamic force acting on the boundary of particle. Due to the accurately resolved flow field, it was shown in [29] that the boundary condition-enforced IB-LBM provides a simple and accurate way to compute lift and drag forces exerted on the immersed object. In this paper, this new IB-LBM is applied to simulate the

migration of a neutrally buoyant particle in a simple shear flow and the sedimentation of one particle in a box and two particles in a channel. The particle motion in a 2D symmetric stenotic artery is also simulated. All the obtained results compare very well with experimental and numerical data in the literature.

2 Numerical method

2.1 Boundary condition-enforced immersed boundary-lattice Boltzmann method

For the immersed boundary method (IBM), the effect of boundary is represented by a set of force density which acts on the fluid field surrounding the boundary. For the viscous and incompressible flows, the governing equations in IBM are

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)+\nabla p=\mu\Delta\mathbf{u}+\mathbf{f}, \tag{2.1}$$

$$\nabla\cdot\mathbf{u}=0, \tag{2.2}$$

$$\mathbf{f}(\mathbf{x},t)=\int_{\Gamma}\mathbf{F}(s,t)\delta(\mathbf{x}-\mathbf{X}(s,t))ds, \tag{2.3}$$

$$\frac{\partial \mathbf{X}(s,t)}{\partial t}=\mathbf{u}(\mathbf{X}(s,t),t)=\int_{\Omega}\mathbf{u}(\mathbf{x},t)\delta(\mathbf{x}-\mathbf{X}(s,t))d\mathbf{x}. \tag{2.4}$$

Here $\mathbf{X}$, $\mathbf{u}$, p and $\mathbf{f}$ are the Eulerian coordinates, fluid velocity, fluid pressure and the force density acting on the fluid field, respectively. $\mathbf{X}$ and $\mathbf{F}$ are the Lagrangian coordinates and boundary force density. And $\delta(\mathbf{x}-\mathbf{X}(s,t))$ is a Dirac delta function. Eqs. (2.1)-(2.2) are the N-S equations with force density. Eqs. (2.3)-(2.4) describe the interaction between the immersed boundary and the fluid by distributing the boundary force at the Lagrangian points to Eulerian points and interpolating the velocity at the Eulerian points to Lagrangian points.

In the present study, the flow field governed by Eqs. (2.1) and (2.2) is obtained by the lattice Boltzmann equation. In order to correctly recover Eqs. (2.1) and (2.2) in the lattice Boltzmann frame, the contribution of the force density to both momentum $\rho\mathbf{u}$ and momentum flux $\rho\mathbf{uu}$ should be considered [30, 31]. Here, the LBE proposed by Guo et al. [31] is employed

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta t,t+\delta t)-f_\alpha(\mathbf{x},t)=-\frac{1}{\tau}\left(f_\alpha(\mathbf{x},t)-f_\alpha^{eq}(\mathbf{x},t)\right)+F_\alpha\delta t, \tag{2.5}$$

$$F_\alpha=\left(1-\frac{1}{2\tau}\right)w_\alpha\left(\frac{\mathbf{e}_\alpha-\mathbf{u}}{c_s^2}+\frac{\mathbf{e}_\alpha\cdot\mathbf{u}}{c_s^4}\mathbf{e}_\alpha\right)\cdot\mathbf{f}, \tag{2.6}$$

$$\rho\mathbf{u}=\sum_\alpha\mathbf{e}_\alpha f_\alpha+\frac{1}{2}\mathbf{f}\delta t, \tag{2.7}$$

where f_α is the distribution function; f_α^{eq} is its corresponding equilibrium state; τ is the single relaxation time; $\mathbf{e}_\alpha$ is the particle velocity; w_α are coefficients which depend on the selected lattice; $\mathbf{f}$ is the force density acting on the fluid field. For the popular D2Q9 model [32], the lattice velocity set is given by

$$\mathbf{e}_\alpha = \begin{cases} 0 & \alpha = 0, \\ (\cos[(\alpha-1)\pi/2], \sin[(\alpha-1)\pi/2]) & \alpha = 1,2,3,4, \\ \sqrt{2}(\cos[(\alpha-5)\pi/2+\pi/4], \sin[(\alpha-5)\pi/2+\pi/4]) & \alpha = 5,6,7,8. \end{cases} \tag{2.8}$$

And the corresponding equilibrium distribution function is

$$f_\alpha^{eq}(\mathbf{x},t) = \rho w_\alpha \left[1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2 - (c_s|\mathbf{u}|)^2}{2c_s^4} \right]. \tag{2.9}$$

Here, $w_0 = 4/9$, $w_\alpha = 1/9$, ($\alpha = 1,\cdots,4$) and $w_\alpha = 1/36$, ($\alpha = 5,\cdots,8$). $c_s = 1/\sqrt{3}$ is the sound speed of this model. The relationship between the relaxation time and the kinematic viscosity of fluid is $v = \left(\tau - \frac{1}{2}\right) c_s^2 \delta t$.

In IBM, at every evolution time step, if the velocity of fluid at the boundary point is equal to the velocity of boundary at the same position, the non-slip boundary condition would be strictly guaranteed. In our proposed boundary condition-enforced IB-LBM [29], the fluid velocities at the boundary points are enforced to be equal to the boundary velocities, which would produce a set of velocity corrections at the boundary points. A brief description of this method is shown below. By defining the intermediate fluid velocity $\mathbf{u}^*$ as

$$\mathbf{u}^* = \frac{1}{\rho} \sum_\alpha \mathbf{e}_\alpha f_\alpha \tag{2.10}$$

and the velocity correction of fluid field $\mathbf{u}^*$ as

$$\delta\mathbf{u} = \frac{1}{2\rho} \mathbf{f} \delta t, \tag{2.11}$$

then, Eq. (2.7) can be expressed as

$$\mathbf{u} = \mathbf{u}^* + \delta\mathbf{u}. \tag{2.12}$$

In this work, the velocity correction $\delta\mathbf{u}$ is set as unknown, which is determined in such a way that the velocity at the boundary point interpolated from the corrected fluid velocity (i.e. Eq. (2.12)) satisfies the non-slip boundary condition. As shown in Fig. 1, the velocity correction $\delta\mathbf{u}$ at Eulerian points is distributed from the velocity correction at the boundary (Lagrangian) points. According to the idea of IBM, the immersed boundary can be represented by a set of Lagrangian points $\mathbf{X}_B(s_l,t)$, $l = 1,2,\cdots,m$. Setting an unknown velocity correction vector $\delta\mathbf{u}_B^l$ at every boundary point, the fluid velocity correction $\delta\mathbf{u}$ can be calculated using the Dirac delta function

$$\delta\mathbf{u}(\mathbf{x},t) = \int_\Gamma \delta\mathbf{u}_B(\mathbf{X}_B,t)\delta(\mathbf{x} - \mathbf{X}_B(s,t))ds. \tag{2.13}$$

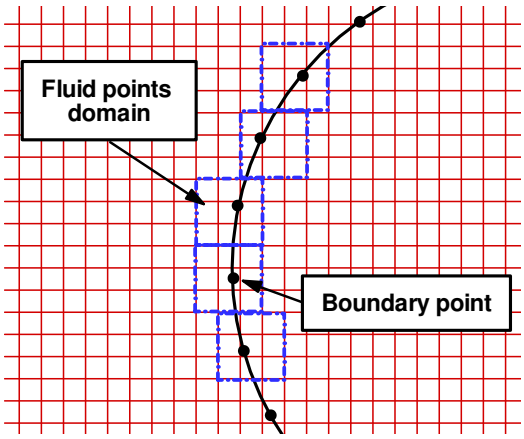

Figure 1: Relationship between boundary points and fluid points.

In the actual implementation, $\delta(\mathbf{x}-\mathbf{X}_B(s,t))$ is smoothly approximated by a continuous kernel distribution

$$\delta(\mathbf{x}-\mathbf{X}_B(s,t))=D_{ij}\left(\mathbf{x}_{ij}-\mathbf{X}_B^l\right)=\delta\left(x_{ij}-X_B^l\right)\delta\left(y_{ij}-Y_B^l\right), \tag{2.14}$$

where $\delta(r)$ is proposed by Peskin [33]

$$\delta(r)=\begin{cases}\dfrac{1}{4h}\left(1+\cos\left(\dfrac{\pi r}{2}\right)\right), & |r|\leqslant 2,\\ 0, & |r|>2;\end{cases} \tag{2.15}$$

h is the mesh spacing of Eulerian mesh around the boundary. Using Eq. (2.14), the velocity correction at Eulerian points can be expressed as

$$\delta\mathbf{u}\left(\mathbf{x}_{ij},t\right)=\sum_l \delta\mathbf{u}_B^l\left(\mathbf{X}_B^l,t\right)D_{ij}\left(\mathbf{x}_{ij}-\mathbf{X}_B^l\right)\Delta s_l \qquad (l=1,2,\cdots,m), \tag{2.16}$$

where Δs_l is the arc length of the boundary element.

In order to satisfy the non-slip boundary condition, the fluid velocity at the boundary point must be equal to the boundary velocity at the same position

$$\mathbf{U}_B^l\left(\mathbf{X}_B^l,t\right)=\sum_{i,j}\mathbf{u}\left(\mathbf{x}_{ij},t\right)D_{ij}\left(\mathbf{x}_{ij}-\mathbf{X}_B^l\right)\Delta x\Delta y. \tag{2.17}$$

Here, $\mathbf{U}_B^l$ is the boundary velocity; $\mathbf{u}$ is the fluid velocity, which is corrected by the velocity correction $\delta\mathbf{u}$

$$\mathbf{u}\left(\mathbf{x}_{ij},t\right)=\mathbf{u}^*\left(\mathbf{x}_{ij},t\right)+\delta\mathbf{u}\left(\mathbf{x}_{ij},t\right), \tag{2.18}$$

where $\mathbf{u}^*$ is the intermediate fluid velocity given from Eq. (2.10).

In Eq. (2.17), the unknown variables are the velocity corrections $\delta\mathbf{u}_B^l$ at the boundary points. After substituting Eqs. (2.16) and (2.18) into Eq. (2.17), the following equation system can be obtained

$$\mathbf{U}_B^l\left(\mathbf{X}_B^l,t\right)=\sum_{i,j}\mathbf{u}^*\left(\mathbf{x}_{ij},t\right)D_{ij}\left(\mathbf{x}_{ij}-\mathbf{X}_B^l\right)\Delta x\Delta y + \sum_{i,j}\sum_{l}\delta\mathbf{u}_B^l\left(\mathbf{X}_B^l,t\right)D_{ij}\left(\mathbf{X}_B^l-\mathbf{x}_{ij}\right)\Delta s_l D_{ij}\left(\mathbf{x}_{ij}-\mathbf{X}_B^l\right)\Delta x\Delta y. \tag{2.19}$$

Setting $\delta_{ij}^B=D_{ij}(\mathbf{x}_{ij}-\mathbf{X}_B^l)\Delta s_l$ and $\delta_{ij}=D_{ij}(\mathbf{x}_{ij}-\mathbf{X}_B^l)\Delta x\Delta y$, then system (2.19) can be rewritten as a matrix form

$$\mathbf{AX}=\mathbf{B}, \tag{2.20}$$

where

$$\mathbf{X}=\{\delta\mathbf{u}_B^1,\delta\mathbf{u}_B^2,\cdots,\delta\mathbf{u}_B^m\}^T,$$

$$\mathbf{A}=\begin{pmatrix}\delta_{11}&\delta_{12}&\cdots&\delta_{1n}\\ \delta_{21}&\delta_{22}&\cdots&\delta_{2n}\\ \vdots&\vdots&\ddots&\vdots\\ \delta_{m1}&\delta_{m2}&\cdots&\delta_{mn}\end{pmatrix}\begin{pmatrix}\delta_{11}^B&\delta_{12}^B&\cdots&\delta_{1m}^B\\ \delta_{21}^B&\delta_{22}^B&\cdots&\delta_{2m}^B\\ \vdots&\vdots&\ddots&\vdots\\ \delta_{n1}^B&\delta_{n2}^B&\cdots&\delta_{nm}^B\end{pmatrix},$$

$$\mathbf{B}=\begin{pmatrix}\mathbf{U}_B^1\\ \mathbf{U}_B^2\\ \vdots\\ \mathbf{U}_B^m\end{pmatrix}-\begin{pmatrix}\delta_{11}&\delta_{12}&\cdots&\delta_{1n}\\ \delta_{21}&\delta_{22}&\cdots&\delta_{2n}\\ \vdots&\vdots&\ddots&\vdots\\ \delta_{m1}&\delta_{m2}&\cdots&\delta_{mn}\end{pmatrix}\begin{pmatrix}\mathbf{u}_1^*\\ \mathbf{u}_2^*\\ \vdots\\ \mathbf{u}_n^*\end{pmatrix},$$

where m is the number of Lagrangian (boundary) points, and n is the number of surrounding Eulerian points used in the delta function interpolation. By solving equation system (2.20), the unknown variables $\delta\mathbf{u}_B^l$ can be obtained. It is shown that the elements of matrix $\mathbf{A}$ are only related to the boundary points and their surrounding Eulerian points. After obtaining the velocity correction at the boundary point, the velocity correction $\delta\mathbf{u}$ and corrected fluid velocity can then be computed by using Eqs. (2.16) and (2.18). In the LBM computation, the macroscopic density and pressure are calculated by

$$\rho=\sum_{\alpha}f_\alpha,\qquad P=c_s^2\rho. \tag{2.21}$$

For the particulate flow problems, the motion of particles is controlled by the force and torque exerted on them, which may vary with the positions of particles. Therefore, it is necessary to compute the force and torque accurately and efficiently. Indeed, the present IB-LBM provides an effective way to compute the force and torque. According to Eq. (2.11), the boundary force density can be calculated by

$$\mathbf{F}\left(\mathbf{X}_B^l\right)=2\rho\delta\mathbf{U}_B^l/\delta t. \tag{2.22}$$

This is the force exerted on the fluid, and the hydrodynamic force exerted on the boundary can be easily obtained from the Newton's third law.

The total force on a particle includes the gravity/buoyancy force, hydrodynamic force and the particle collision force $\mathbf{F}_i^{\text{col}}$. Mathematically, it can be written as

$$\mathbf{F}_i = \left(1-\frac{\rho_f}{\rho_p}\right) M_i \mathbf{g} - \sum_l \mathbf{F}\left(\mathbf{X}_B^l\right) \Delta s_l + \mathbf{F}_i^{\text{col}}, \tag{2.23}$$

where M_i is the mass of the particle; ρ_f and ρ_p are the density of fluid and particle, respectively. The torque acting on the particle is expressed as

$$\mathbf{T}_i = -\sum_l \left(\mathbf{X}_B^l - \mathbf{X}_R\right) \times \mathbf{F}\left(\mathbf{X}_B^l\right) \Delta s_l, \tag{2.24}$$

where $\mathbf{X}_R$ is the center of mass of the particle.

After obtaining the force and torque exerted on the particle, the motion of the particle can then be determined by using the following equations

$$M_i \frac{d\mathbf{U}_R}{dt} = \mathbf{F}_i, \tag{2.25}$$

$$I_i \frac{d\mathbf{\Omega}}{dt} = \mathbf{T}_i, \tag{2.26}$$

where $\mathbf{U}_R$ and $\mathbf{\Omega}$ are the translational and angular velocities of the particle, respectively; I_i is the moment inertia of the particle. In summary, the numerical implementation of present scheme can be outlined as follows

---

Step 1: Set initial values, compute the matrix $\mathbf{A}$ and get its inverse matrix $\mathbf{A}^{-1}$;

Step 2: Use Eq. (2.5) to get the density distribution function at time level $t = t_n$ (initially setting $F_\alpha = 0$) and compute the macroscopic variables using Eqs. (2.10) and (2.21);

Step 3: Solve equation system (2.20) to get the velocity corrections at the boundary points and use Eq. (2.16) to get the fluid velocity corrections;

Step 4: Modify the fluid velocity by using Eq. (2.18);

Step 5: Compute the equilibrium distribution function using Eq. (2.9);

Step 6: Compute the boundary force density by using Eq. (2.22);

Step 7: Compute the force and torque exerted on the particle by using Eqs. (2.23) and (2.24);

Step 8: Update the boundary position and velocities of the particle by Eqs. (2.25) and (2.26).

Step 9: Repeat Step 2 to 8 until convergence is reached.

2.2 Particle collision rule

In the particulate flows, collisions between particles or particle and wall are unavoidable. It is important to handle these collisions for the study of all particulate problems. Usually, in order to implement the repulsive force for the collision process, an artificial mechanism is introduced in the numerical scheme. Here, we select the model which is used by Niu et al. [27]

$$\mathbf{F}_i^{p-p}=\begin{cases}0 & X_R^{i,j}>r_i+r_j+\zeta,\\ 2.4\varepsilon\sum_{j=1,j\neq i}^{N}\left[2\left(\frac{r_i+r_j}{X_R^{i,j}}\right)^{14}-\left(\frac{r_i+r_j}{X_R^{i,j}}\right)^{8}\right]\frac{\mathbf{X}_R^i-\mathbf{X}_R^j}{(r_i+r_j)^2} & X_R^{i,j}\leqslant r_i+r_j+\zeta,\end{cases} \tag{2.27}$$

$$\mathbf{F}_i^{p-w}=\begin{cases}0 & X_R^{i,w}>2r_i+\zeta,\\ 2.4\varepsilon\sum_{j=1}^{N}\left[2\left(\frac{r_i}{X_R^{i,w}}\right)^{14}-\left(\frac{r_i}{X_R^{i,w}}\right)^{8}\right]\frac{\mathbf{X}_R^i-\mathbf{X}_w}{(r_i)^2} & X_R^{i,w}\leqslant 2r_i+\zeta,\end{cases} \tag{2.28}$$

where $\varepsilon=(\frac{2r_ir_j}{r_i+r_j})^2$, r_i is radius of the particle, $X_R^{i,j}=|\mathbf{X}_R^i-\mathbf{X}_R^j|$, $X_R^{i,w}=|\mathbf{X}_R^i-\mathbf{X}_w|$, $\mathbf{X}_w$ represents the wall position and ζ is the threshold and is set to one lattice unit in our simulation. Therefore, the particle collision force is $\mathbf{F}_i^{\text{col}}=\mathbf{F}_i^{p-p}+\mathbf{F}_i^{p-w}$.

3 Numerical results and discussion

3.1 A moving neutrally buoyant particle in linear shear flow

As indicated in [29], the present method could calculate the forces exerted on the immersed boundary accurately due to the exact satisfaction of non-slip boundary condition. Therefore, it could be applied to simulate the particulate flows efficiently. To assess this capability, the motion of a single neutrally buoyant circular particle in linear shear flow is investigated. This problem was studied by Feng et al. [34] using finite element method. They found that the particle always transfers to the center of channel, independent of its initial position and velocity. Recently, this problem was also studied by Feng and Michaelides [25] and Niu et al. [27] based on the IB-LBM.

Fig. 2 shows the schematic diagram of this problem. The parameters used in present simulation are as follows: the distance between two plates is $H=1$; the length of plates is $L=25$; the diameter of the circular particle is $D=0.25H$; the upper and lower plates are moving at constant velocity of $U_w/2=3/80$ in the opposite directions; the particle density is equal to the fluid density, i.e. $\rho_f=\rho_p=1.0$; the Reynolds number is $\text{Re}=U_wH/\upsilon=40$. A uniform mesh for the computational domain is used and the mesh size is 2001×81. The periodic boundary condition is used at the inlet and outlet of channel. The initial position of particle is located at $y_0=0.25H$ above the lower plate and is initially at rest.

Fig. 3 demonstrates the lateral migration of the particle using present method together with the results of others [27, 34]. From this figure, it can be found that the particle migrates to the centerline, just the same as observed by Feng et al. [34]. At the same time, it

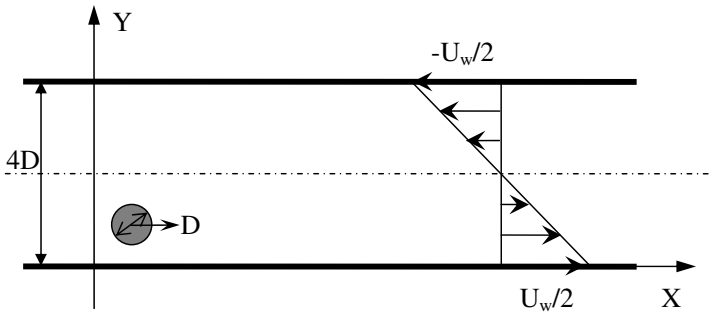

Figure 2: Schematic diagram of a neutrally buoyant particle in a linear shear flow.

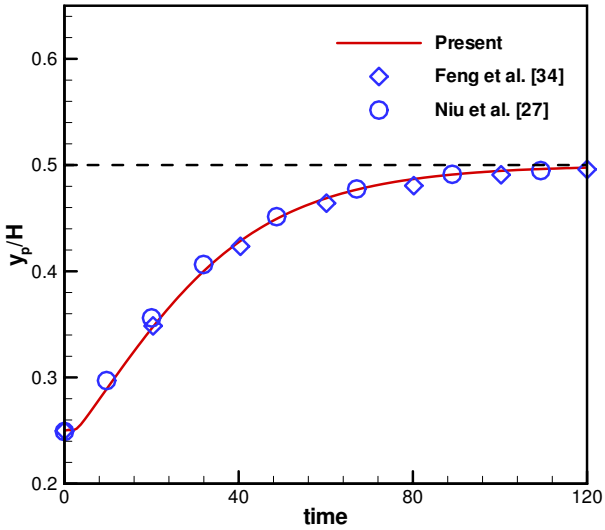

Figure 3: Comparison of lateral migration of particle with previous data.

is shown that a very good agreement is obtained between the present results and those of others [27,34]. The two components of the particle translational velocity obtained using present method and conventional IB-LBM [25,27] are plotted in Fig. 4. Again, the preset results agree well with previous numerical results.

In current simulation, the particle surface is represented by 50 Lagrangian points with uniform distribution. To investigate the effects of the number of Lagrangian points on the simulation accuracy, the simulation of present problem with different number of surface points is carried out. Here, four numbers of surface points are used: 30, 40, 50, and 60. Table 1 shows the steady angular velocity of particle for the four cases. From the table, it is clear that the number of surface points affects the simulation accuracy very little.

From the above simulation, it was found that our proposed IB-LBM [29] can efficiently and accurately simulate the flows with particles. As compared to conventional IB-LBM [25,27], the obtained flow field is more accurate and thus, it could produce more accurate

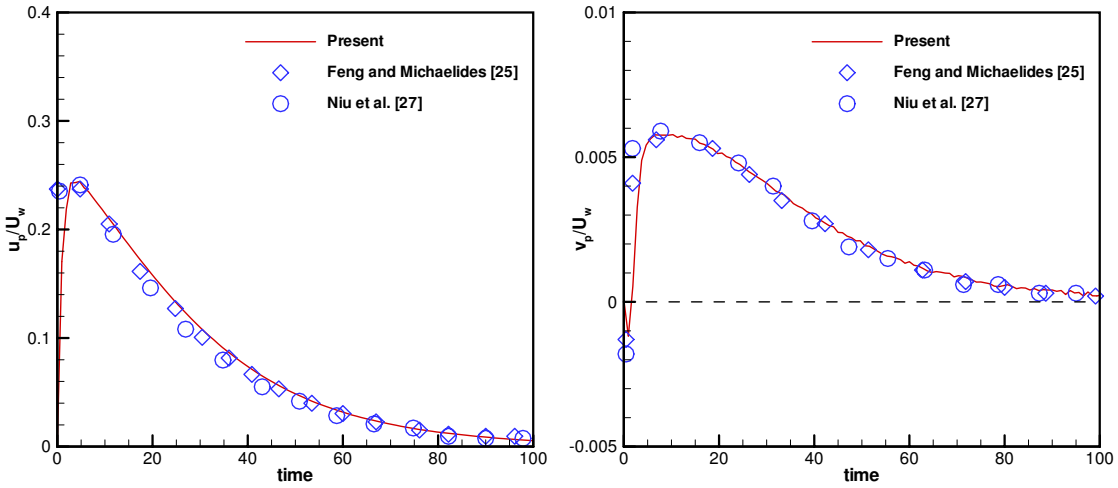

Figure 4: Comparison of particle translational velocities with previous data.

Table 1: Effects of the number of Lagrangian points.

Number of surface points	Steady angular velocity of particle
30	0.035242
40	0.035231
50	0.035234
60	0.035214

calculation of forces exerted on the particle. In addition, the boundary force density can be simply determined from the velocity correction. In contrast, the boundary force density is calculated by solving the momentum equations in the direct forcing method [26] or interpolating the distribution functions from the Eulerian points to Lagrangian points in the momentum exchange method [27]. Clearly, the force calculation by present method is simpler. On the other hand, we have to indicate that it is required to calculate the elements of matrix **A** and its inversion at every time step due to the motion of particle. This will take additional computational effort. However, as indicated in [29], the increase of computation effort is very little since the number of boundary points is much less than the total number of Eulerian points.

3.2 Particle sedimentation in viscous fluid

The particle sedimentation is an important aspect in the area of engineering application. In order to further validate the present method, the sedimentation of one particle and two particles in the viscous fluid are simulated.

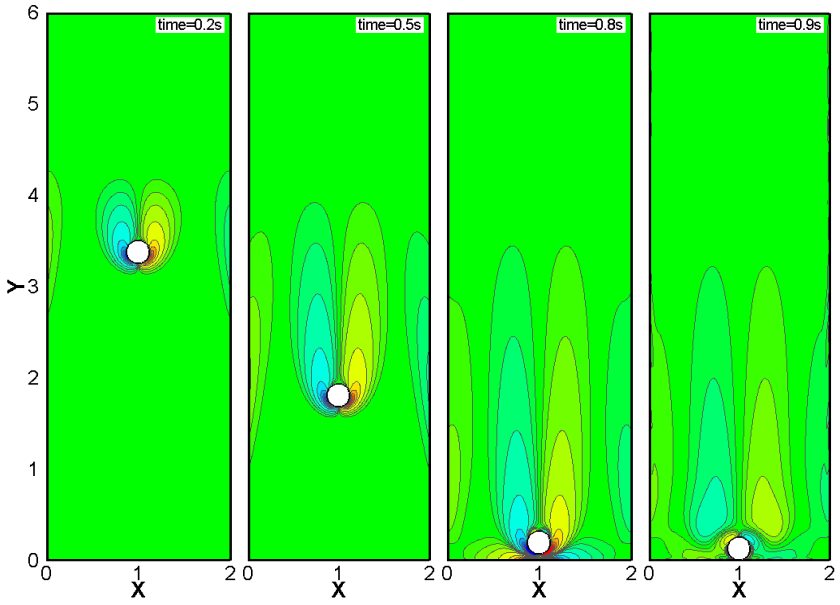

Figure 5: Instantaneous vorticity contours at different time stages.

3.2.1 One particle sedimentation

As a validation of numerical method, this problem has been extensively studied. In present simulation, a box with $2cm$ width (x-direction) and $6cm$ height (y-direction) is selected. The viscosity μ and density ρ_f of fluid in the box are $0.1g/(cm\cdot s)$ and $1.0g/cm^3$, respectively. The density of rigid circular particle is $\rho_p = 1.25g/cm^3$, and its radius is $0.125cm$. Initially, the particle is set at ($1cm$, $4cm$) with the static (same as the fluid) state. Due to gravity force, the particle will be falling down. A uniform mesh of 201×601 is used in the current simulation. The particle surface is represented by 50 Lagrangian points with uniform distribution.

Fig. 5 displays the instantaneous vorticity contours at different time stages: $0.2s$, $0.5s$, $0.8s$ and $0.9s$. The clear variation of flow structure can be observed. The evolutions of some quantities with respect to time are plotted in Fig. 6. They are longitudinal coordinate of particle center y_p, longitudinal velocity of particle center v_p, Reynolds number of particle Re_p and translational kinetic energy E_t. Here, the Reynolds number and energy are defined as

$$\mathrm{Re}_p = \frac{\rho_p d_p \sqrt{u_p^2+v_p^2}}{\mu}, \tag{3.1}$$

and

$$E_t = 0.5M\left(u_p^2+v_p^2\right), \tag{3.2}$$

where u_p and v_p are two components of velocity of particle center; d_p is the diameter

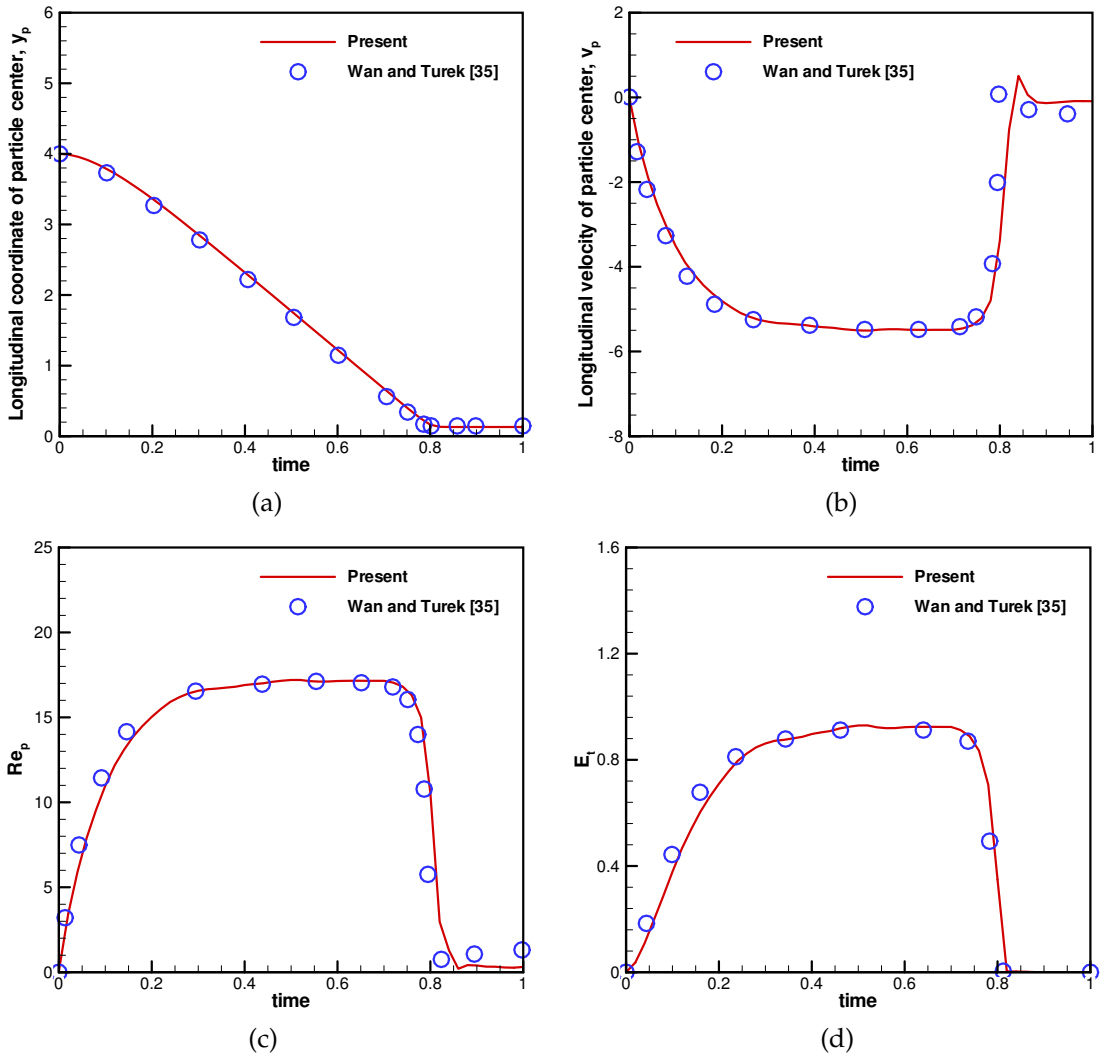

Figure 6: Time histories of some quantities (a) longitudinal coordinate; (b) longitudinal velocity; (c) Reynolds number; (d) translational kinetic energy.

of particle; M is the mass of particle. For comparison, the results of Wan and Turek [35], who applied multigrid finite element and fictitious domain method to solve this problem, are also included in Fig. 6. It is clear from the figure that the present results compare well with those of Wan and Turek [35]. The small differences between present and previous results after the particle reaches the bottom are attributed to the different use of particle-wall collision rule. The maximum Reynolds number of particle in the present computation is 17.08, which is very close to 17.15 provided by Wan and Turek [35].

3.2.2 Two particles sedimentation

Similar to the case of one particle sedimentation, this problem has also been widely studied. Here, a channel of width $2cm$ and height $8cm$ is considered. The properties of fluid are $\mu = 0.001g/(cm\cdot s)$ and $\rho_f = 1.0g/cm^3$. The two particles have the same properties. The density is $\rho_p = 1.01g/cm^3$ and the radius is $0.1cm$. Initially, both fluid and particles are at rest. The first particle (P1) is set at ($0.999cm$, $7.2cm$) and the other (P2) at ($1cm$, $6.8cm$). The two particles start dropping under the gravity force. In present simulation, a uniform mesh is used with mesh size of 201×801. The particle surface is represented by 50 Lagrangian points with uniform distribution.

It is known that the two particles settled close to each other would go through drafting, kissing and tumbling or DKT motion [36]. In Fig. 7, the results of present computation show the positions of two particles at several different time stages. From the figure, the clear DKT motion is demonstrated.

The instantaneous vorticity contours at time stages of $1.5s$, $2.5s$, $3.5s$ and $4.5s$ are given in Fig. 8. The interesting phenomena due to the DKT can be observed. Figs. 9 and 10 depict the instantaneous transverse and longitudinal coordinates of two particle centers, together with the results of Feng et al. [25] and Niu et al. [27]. It can be found that these results are in good agreement until kissing and tumbling begin. As pointed out by Fortes et al. [36], the tumbling is essentially a breakup of an unstable configuration of the particle positions. Thus an exact agreement after kissing may not be expected.

From Figs. 7-10, it can be found that P1 initially trails P2 by keeping a steady distance (about $0.4cm$). At about time stage of $0.8s$, two particles start to approach closer. Then about $0.6s$ later, two particles almost touch each other, which means that they enter the "kissing" part of motion. At the same time, the centers of two particles start to slowly deviate from the channel center and to move closer. At about time stage of $2.4s$, the particles completely tumble. The same phenomena at these three time stages were also observed in [25] and [27].

3.3 Particle suspension in a 2D symmetric stenotic artery

In order to examine the present method for the particulate flows with complex geometries, the particle suspension in a two-dimensional stenotic artery is simulated. This kind of flow is of great interest in medical science. There are numerous papers on the study of the pulsatile flow in a mildly or severely stenotic artery. Recently, such problem was studied by Li et al. [37] employing LBM.

As shown in Fig. 11, this system is a two-dimensional rigid planar channel with length $L = 32d$ and width $W = 8d$, where d is the diameter of the rigid circular particles. The stenosis is created by adding two symmetric protuberances inside the vessel. The protuberance is a semicircle with radius determined by the width of the stenosis throat b. In our simulations $d < b < 2d$. The densities of both fluid and particles are $1.0g/cm^3$ and the viscosity of fluid is $0.01g/(cm\cdot s)$. The pressure difference between the inlet and outlet is 541 Pa. Hence, a pressure boundary condition which was proposed by Zou and He [38]

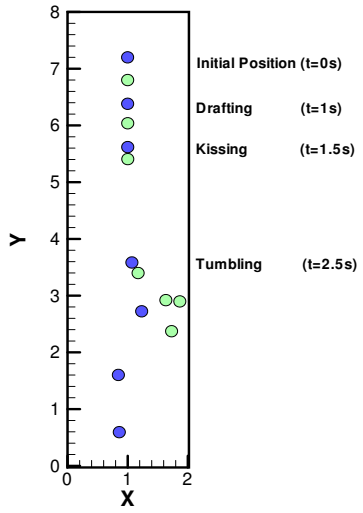

Figure 7: Sedimentation of two particles in a channel at different time stages.

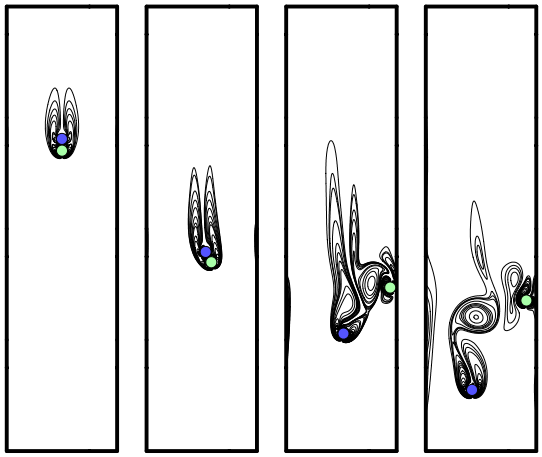

Figure 8: Instantaneous vorticity contours at different time stages.

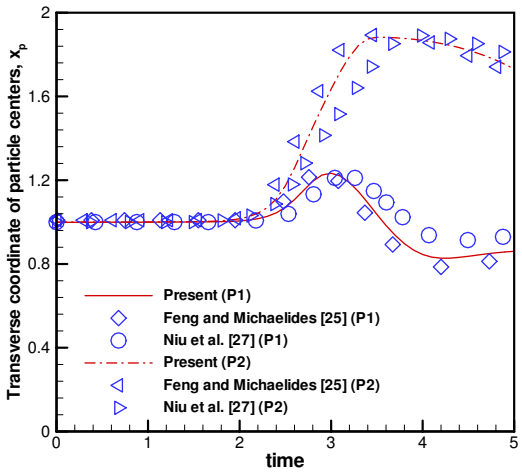

Figure 9: Transverse coordinates of two particle centers.

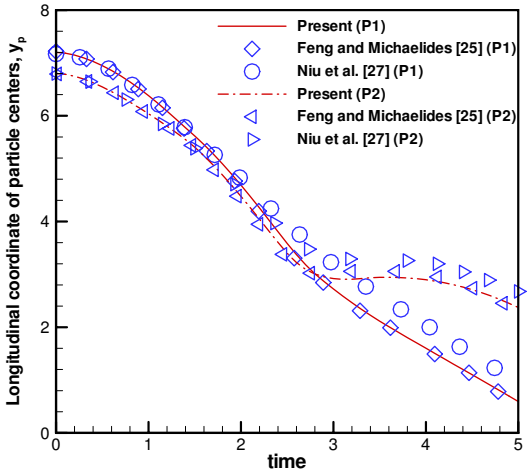

Figure 10: Longitudinal coordinates of two particle centers.

is applied. The radius of the particle is 8 lattice units. The particle surface is represented by 50 Lagrangian points with uniform distribution. The particles are placed 8d left to the stenosis throat and keep fixed in the first 5000 time steps. After that time, the particles are set to move freely due to the forces and torques exerting on them.

3.3.1 One particle passes the stenosis throat with $b = 1.75d$

When the particle gets across the stenosis throat, the velocity is larger than that in the flat tube. Fig. 12 provides the snapshots for the initial position of particle $2d$ above the

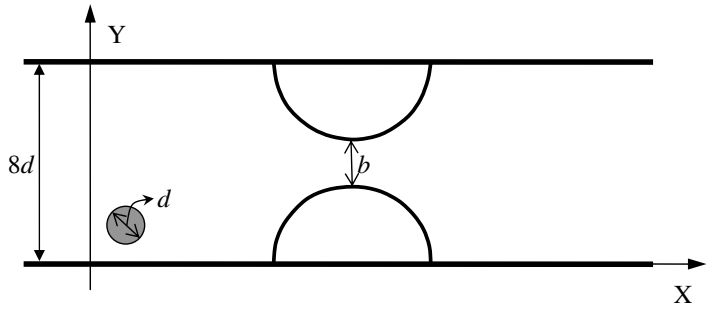

Figure 11: Schematic diagram of the planar channel with stenosis.

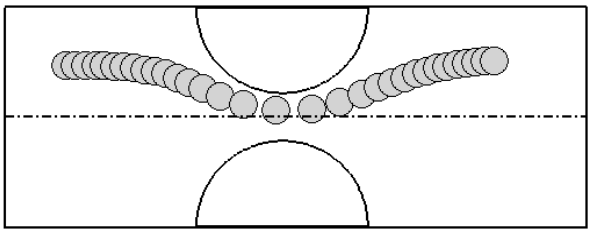

Figure 12: Snapshots of the positions of a particle.

centerline. From Fig. 12, it is clear that the particle moves toward the centerline when it gets near and across the protuberance. After passing the stenosis, the particle migrates up to the same height of the initial position. This observation is the same as that in [37]. The non-dimensional velocity at the center of particle is shown in Fig. 13. The x-direction velocity at stenosis throat is about five times of that in the flat tube. The y-direction velocity is almost equal to zero far away the protuberance and displays a wave shape when the particle passes stenosis throat.

3.3.2 Two particles pass the stenosis throat with $b=1.75d$

Since the gap between two protuberances is only $1.75d$ and the particles are rigid, two particles can not pass the throat side by side. The trajectories of two particles, which are symmetric to the centerline initially, are given in Fig. 14. The particles stop before the throat and the velocities of fluid become zero after the particles block the throat completely. However, as indicated in [37], there is space for the fluid to flow even the throat is blocked by the particles in three-dimensional simulation.

When two particles are positioned asymmetric to the centerline initially, the different phenomena appear. Fig. 15 shows the snapshots. Initially the upper particle is positioned $2d+S$ above the centerline and the lower particle is positioned $2d$ below the centerline, where $S=d/4000$. Because of this very small asymmetry, the particles can pass the throat. Before the particles reach the throat, they go forward side by side. However, when they are going to get across the throat, the effect of small asymmetry is exhibited. The lower

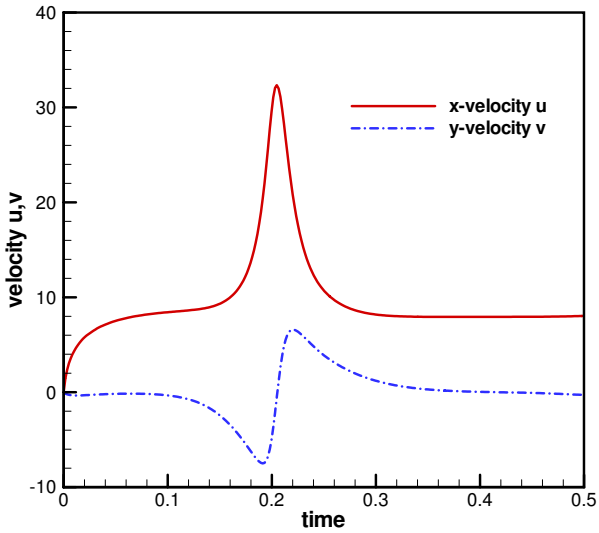

Figure 13: Velocities of the particle with respect to time.

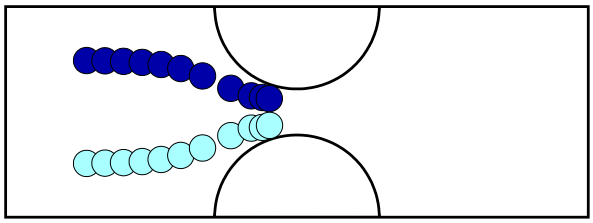

Figure 14: Trajectories of two particles symmetric to the centerline initially.

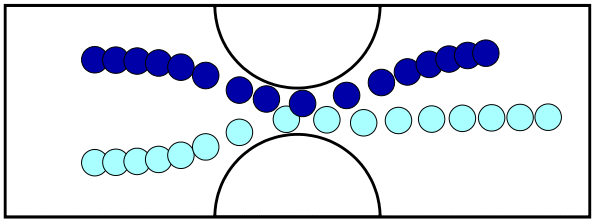

Figure 15: Trajectories of two particles with the initial position asymmetric to the centerline with asymmetry $d/4000$.

particle keeps moving while the upper particle stops and then turns back, leaving space to let the lower particle pass the throat. Once the lower particle passes the throat, the upper one turns around again and follows the lower particle. The enlarged part of the x-direction coordinates of two particles with respect to the time is shown in Fig. 16. This figure clearly shows the process.

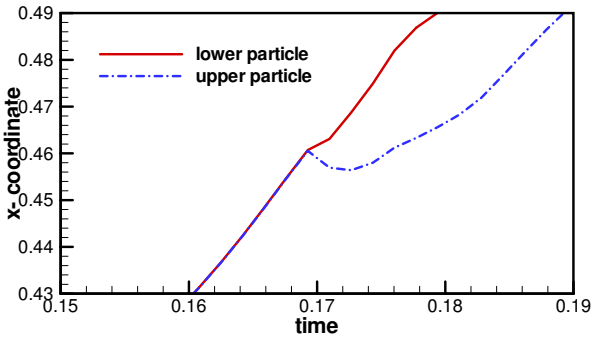

Figure 16: Enlarged part of x-direction coordinates with respect to the time when two particles pass the throat.

4 Conclusions

A developed immersed boundary-lattice Boltzmann method for simulating the particulate flows is provided in this paper. The basic idea of IBM is that the effect of boundary can be replaced by the restoring force acting on the fluid field. Therefore, the key issue is to obtain this restoring force. In the conventional IB-LBM, such force density is determined in advance. As a result, the non-slip boundary condition is not exactly satisfied. To overcome this drawback in conventional IB-LBM, in this study, the force density is regarded as unknown which is determined by enforcing the non-slip boundary condition. For simulating the particulate flows, accurate computation of the hydrodynamic force which controls the motion of particles is very important. Since the non-slip boundary condition is enforced, the force calculation is accurate in the present IB-LBM.

In order to validate the method for handling particulate flow problems, the simulation of migration of a neutrally buoyant particle in a simple shear flow is carried out. Then, the sedimentation of one particle in a box and two particles in a channel is modeled. Finally, the cases of particle suspension in a 2D symmetric stenotic artery are studied. All the obtained numerical results compare well with previous experimental and numerical results. It seems that the boundary condition-enforced IB-LBM is a suitable scheme to handle the particulate flow problems.

References

[1] H. H. Hu, Direct simulation of flows of solid-liquid mixtures, Int. J. Multiphase Flow, 22, 335-352, 1996.
[2] B. Maury, Direct simulations of 2D fluid-particle flows in biperiodic domains, J. Comput. Phys., 156, 325-351, 1999.
[3] H. H. Hu, N. A. Patankar, and M. Y. Zhu, Direct numerical simulations of fluid-solid systems using the arbitrary Lagrangian-Eulerian technique, J. Comput. Phys., 169, 427-462, 2001.

[4] R. Glowinski, T. W. Pan, T. I. Hesla, and D. D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow, 25, 755-794, 1999.
[5] R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, and J. Priaux, A distributed Lagrange multiplier/fictitious domain method for flows around moving rigid bodies: application to particulate flow, Int. J. Numer. Meth. Fluids, 30, 1043-1066, 1999.
[6] P. Singh, D. D. Joseph, T. I. Hesla, R. Glowinski, and T. W. Pan, A distributed Lagrange multiplier/fictitious domain method for viscoelastic particulate flows, J. Non-Newtonian Fluid Mech., 91, 165-188, 2000.
[7] Z. Yu, N. Phan-Thien, Y. Fan, and R. I. Tanner, Viscoelastic mobility of a system of particles, J. Non-Newtonian Fluid Mech., 104, 87-124, 2002.
[8] C. S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys., 25, 220-252, 1977.
[9] A. L. Fogelson and C. S. Peskin, A fast numerical method for solving the three-dimensional Stokes equation in the presence of suspended particles, J. Comput. Phys., 79, 50-69, 1988.
[10] K. Hfler and S. Schwarzer, Navier-Stokes simulation with constraint forces: finite-difference method for particle-laden flows and complex geometries, Phys. Rev. E, 61, 7146-7160, 2000.
[11] E. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys., 161, 35-60, 2000.
[12] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys., 209, 448-476, 2005.
[13] K. Luo, Z. Wang, and J. Fan, A modified immersed boundary method for simulations of fluid-particle interactions, Comput. Meth. Appl. Mech. Eng., 197, 36-46, 2007.
[14] Z. Li and M. C. Lai, The immersed interface method for the Navier-Stokes equations with singular forces, J. Comput. Phys., 171, 822-842, 2001.
[15] D. V. Le, B. C. Khoo, and J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, J. Comput. Phys., 220, 109-138, 2006.
[16] X. Wang and W. K. Liu, Extended immersed boundary method using FEM and RKPM, Comput. Meth. Appl. Mech. Eng., 193, 1305-1321, 2004.
[17] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu, Immersed finite element method, Comput. Meth. Appl. Mech. Eng., 193, 2051-2067, 2004.
[18] S. Chen and G. D. Doolen, Lattice Boltzmann method for fluid flows, Annu. Rev. Fluid Mech., 30, 329-364, 1998.
[19] S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Oxford Univ. Press, New York, 2001.
[20] A. J. C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. I. Theoretical foundation, J. Fluid Mech., 271, 285-310, 1994.
[21] A. J. C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. II. Numerical results, J. Fluid Mech., 271, 311-339, 1994.
[22] O. Behrend, Solid-fluid boundaries in particle suspension simulations via the lattice Boltzmann method, Phys. Rev. E, 52, 1164-1175, 1995.
[23] C. K. Aidun, Y. Lu, and E. Ding, Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation, J. Fluid Mech., 373, 287-311, 1998.
[24] D. Qi, Lattice-Boltzmann simulations of particles in non-zero-Reynolds-number flows, J. Fluid Mech., 385, 41-62, 1999.
[25] Z. Feng and E. Michaelides, The immersed boundary-lattice Boltzmann method for solving

fluid-particles interaction problem, J. Comput. Phys., 195, 602-628, 2004.
[26] Z. Feng and E. Michaelides, Proteus: A direct forcing method in the simulations of particulate flows, J. Comput. Phys., 202, 20-51, 2005.
[27] X. D. Niu, C. Shu, Y. T. Chew, and Y. Peng, A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows, Phys. Lett. A, 354, 173-182, 2006.
[28] C. Shu, N. Y. Liu, and Y. T. Chew, A novel immersed boundary velocity correction-lattice Boltzmann method and its application to simulate flow past a circular cylinder, J. Comput. Phys., 226, 1607-1622, 2007.
[29] J. Wu and C. Shu, Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications, J. Comput. Phys., 228, 1963-1979, 2009.
[30] A. J. C. Ladd and R. Verberg, Lattice-Boltzmann simulations of particle-fluid suspensions, J. Stat. Phys., 104, 1191-1251, 2001.
[31] Z. Guo, C. Zheng, and B. Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, Phys. Rev. E, 65, 046308, 2002.
[32] Y. H. Qian, D. d'Humieres, and P. Lallemand, Lattice BGK model for Navier-Stokes equation, Europhys. Lett., 17, 479-484, 1992.
[33] C. S. Peskin, The immersed boundary method, Acta Numer., 11, 479-517, 2002.
[34] J. Feng, H. H. Hu, and D. D. Joseph, Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 2, Couette and Poiseuille flows, J. Fluid Mech., 277, 271-301, 1994.
[35] D. Wan and S. Turek, Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method, Int. J. Numer. Meth. Fluids, 51, 531-566, 2006.
[36] A. Fortes, D. D. Joseph, and T. S. Lundgren, Nonlinear mechanics of fluidization of beds of spherical particles, J. Fluid Mech., 177, 467-483, 1987.
[37] H. Li, H. Fang, Z. Lin, S. Xu, and S. Chen, Lattice Boltzmann simulation on particle suspensions in a two-dimensional symmetric stenotic artery, Phys. Rev. E, 69, 031919, 2004.
[38] Q. Zou and X. He, On pressure and velocity boundary conditions for the lattice Boltzmann BGK model, Phys. Fluids, 9, 1591-1598, 1997.