Available online at www.sciencedirect.com
SCIENCE DIRECT
Physics Letters A 354 (2006) 173–182
PHYSICS LETTERS A
www.elsevier.com/locate/pla

A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows

X.D. Niu, C. Shu *, Y.T. Chew, Y. Peng

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

Received 21 January 2006; accepted 24 January 2006

Available online 31 January 2006

Communicated by R. Wu

Abstract

A momentum exchange-based immersed boundary-lattice Boltzmann method is presented in this Letter for simulating incompressible viscous flows. This method combines the good features of the lattice Boltzmann method (LBM) and the immersed boundary method (IBM) by using two unrelated computational meshes, an Eulerian mesh for the flow domain and a Lagrangian mesh for the solid boundaries in the flow. In this method, the non-slip boundary condition is enforced by introducing a forcing term into the lattice Boltzmann equation (LBE). Unlike the conventional IBM using the penalty method with a user-defined parameter or the direct forcing scheme based on the Navier–Stokes (NS) equations, the forcing term is simply calculated by the momentum exchange of the boundary particle density distribution functions, which are interpolated by the Lagrangian polynomials from the underlying Eulerian mesh. Numerical examples show that the present method can provide very accurate numerical results.

Keywords: Lattice Boltzmann method; Immersed boundary method; Momentum exchange; Moving boundaries

1. Introduction

As an alternative computational technique to the Navier–Stokes (NS) solvers, the lattice Boltzmann method (LBM) has achieved a great success in simulating complex fluid flows in the past decades [1–13]. LBM is a particle-based numerical technique, 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.

Due to requirement of streaming process from one mesh point to its neighboring point, the standard LBM requires the computational mesh to be a uniform one. For example, the most popular D2Q9 model is based on a square Cartesian mesh. This greatly limits its application to solve practical problems. As we know, most flow problems involve complex geometry and need to use non-uniform mesh to capture the thin boundary layer. To remove the drawbacks of standard LBM, some efforts have been attempted. One way is to use the interpolation-supplemented technique [14,15].

The additional interpolation process greatly increases the computational effort. The other way is to use the Taylor series expansion and least square-based LBM (TLLBM) [16–18]. The interpolation process is actually build-in in the TLLBM to reduce the computational effort. The trade-off of TLLBM is that additional memory is needed to store the coefficients. Other effort includes the change of standard LBE into a set of differential equations, which can be solved by conventional CFD techniques such as finite-difference (FD), finite-element (FE) and finite-volume (FV) methods. This way loses the primary advantages of LBM. It seems that

* Corresponding author.
E-mail address: mpeshuc@nus.edu.sg (C. Shu).

0375-9601/$ – see front matter
doi:10.1016/j.physleta.2006.01.060

among all the versions, the standard LBM is the most efficient one in terms of computational effort and memory requirement. It could become a very efficient method if it can be combined with some other techniques to solve flows with complex geometry.

On the other hand, we notice that the immersed boundary method (IBM) has received a great attention in recent years to simulate flows with complex and moving boundaries [19–22]. IBM was developed by Peskin [19] in 1970s for simulation of the blood flow in the heart. It uses a fixed Eulerian mesh for the fluid, and a set of Lagrangian points to represent the physical boundaries. The basic idea of this method is to treat the boundary as deformable with high stiffness. A small distortion of the particle boundary will yield a force that tends to restore the boundary into its original position. The balances of such forces are distributed into the Eulerian nodes of the grid and the Navier–Stokes equations with a body force are solved over the whole fluid domain. In the IBM, the Eulerian mesh is usually the Cartesian mesh. The common feature of using Cartesian mesh in the standard LBM and IBM motivates the researchers to combine them into an efficient one. The first such attempt was made by Feng and Michaelides [23,24]. In the work of Feng and Michaelides [23,24], the restoration force due to deformation is computed by the penalty method [23] or the direct forcing scheme [24]. The penalty method introduces a user-defined spring parameter which may have a significant effect on the computational efficiency and accuracy. The direct forcing scheme requires solving the NS equations by FD method, which spoils the merits of LBM. On the other hand, the single relaxation collision model (BGK model) is used in their work. As shown by d'Humières [25], the multi-relaxation time collision model has a better computational stability than the BGK model.

In order to improve the stability of immersed boundary-lattice Boltzmann method (IB-LBM) of Feng and Michaelides [23,24], in this work, the multi-relaxation collision model [25] is used in the LBM computation. The most important contribution of this Letter is to propose a simple and efficient way to compute the force at the boundary point. Unlike the IB-LBM of Feng and Michaelides [23,24] using the penalty method with a user-defined parameter or the direct forcing scheme based on the Navier–Stokes (NS) equations, the forcing term is simply calculated by the momentum exchange method proposed by Ladd [13]. Different to the work of Ladd [13], who evaluated the momentum-exchange of the particles along the link of two neighboring mesh points crossing the boundaries, the present method implements the momentum-exchange on the boundary nodes where the particle density distribution functions are interpolated by the Lagrangian polynomials from the underlying Eulerian mesh. Compared to the work of Ladd [13], the present force computation is simpler and more convenient because one does not need to care the details of the boundary position and mesh points. The present method is validated by its application to simulate flows over a line of circular cylinders placed at the middle of a straight channel, a steady flow past a circular cylinder, the migration of a neutrally buoyant particle in a simple shear flow and the flow of two interacting circular particles settling in a channel. The present results are compared well with those available in the literature [15,18,23,26–30].

2. Momentum exchange-based immersed boundary-lattice Boltzmann method

2.1. Lattice Boltzmann model with multi-relaxation time collision

Consider the two-dimensional lattice Boltzmann model on a square lattice with lattice spacing δ_x and nine lattice velocities (D2Q9 model). The velocity set is given by

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

where $c = \delta_x/\delta_t$ and δ_t is assumed to be equal to δ_x. The general lattice Boltzmann model with multi-relaxation time collision can be written as the following equation for the density distribution function f_α moving with velocity $\boldsymbol{e}_\alpha$

$$f_\alpha(\boldsymbol{r}+\boldsymbol{e}_\alpha\delta_t, t+\delta_t) = f_\alpha(\boldsymbol{r},t) - A\big[f_\alpha(\boldsymbol{r},t) - f_\alpha^{\mathrm{eq}}(\boldsymbol{r},t)\big], \tag{2}$$

where f_α^{eq} is the equilibrium density distribution function, $\boldsymbol{r}(x, y)$ is the space position vector, t denotes time and A is the diagonal relaxation matrix with

$$A = \mathrm{diag}(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9). \tag{3}$$

This general model reduces to the usual single-relaxation lattice BGK model [1–3] if all relaxation parameters are set to be a single value τ, i.e., $a_\alpha = 1/\tau$. In this Letter, we adopt the generalized lattice Boltzmann (GLB) model of d'Humières [25] because the coding implementation of this model is almost as simple as that of the BGK model. Most importantly, this model has been shown to have a superior stability over the single-relaxation lattice BGK model as its relaxations are optimized by the linear numerical stability analysis [31]. In the GLB model, a new set of moment variables

$$R = (\rho, e, \varepsilon, j_x, q_x, j_y, q_y, p_{xx}, p_{xy})^{\mathrm{T}}$$

(T is the transpose operator) is introduced by relating it to the set of

$$F = (f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9)^{\mathrm{T}}$$

as follows

$$R = MF. \tag{4}$$

From the kinetic theory of gases and making use of the symmetry feature of the discrete velocity set, M is constructed as

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{pmatrix}. \tag{5}$$

In the vector R, ρ is the fluid density, e is the energy, ε is related to the square of e, j_x and j_y are the components of the moment density $\rho \boldsymbol{u}(u_x, u_y)$ and $\boldsymbol{u}$ is the velocity, q_x and q_y correspond to the energy flux, and p_{xx} and p_{xy} are proportional to the diagonal and off-diagonal components of the viscous stress tensor. The evolution equation of the GLB model can be written as follows

$$R(\boldsymbol{r} + \boldsymbol{e}_\alpha \delta_t, t + \delta_t) - R(\boldsymbol{r}, t) = -S\big\lfloor R(\boldsymbol{r}, t) - R^{\mathrm{eq}}(\boldsymbol{r}, t)\big\rfloor, \tag{6a}$$

$$f_\alpha(\boldsymbol{r} + \boldsymbol{e}_\alpha \delta_t, t + \delta_t) = -M^{-1} R(\boldsymbol{r} + \boldsymbol{e}_\alpha \delta_t, t + \delta_t) \tag{6b}$$

with

$$R^{\mathrm{eq}} = \big(0, e^{\mathrm{eq}}, \varepsilon^{\mathrm{eq}}, 0, q_x^{\mathrm{eq}}, 0, q_y^{\mathrm{eq}}, p_{xx}^{\mathrm{eq}}, p_{xy}^{\mathrm{eq}}\big)^{\mathrm{T}}$$

and $S = M \cdot A \cdot M^{-1} = \mathrm{diag}(0, s_2, s_3, 0, s_5, 0, s_7, s_8, s_9)$. It should be stressed that M^{-1} can be easily computed even by hand using the fact that M is an orthogonal matrix and thus $M \cdot M^{\mathrm{T}}$ is a diagonal matrix. Therefore

$$\begin{aligned} M^{-1} &= M^{\mathrm{T}}\big(M \cdot M^{\mathrm{T}}\big)^{-1} \\ &= \begin{pmatrix} 1/9 & -1/9 & 1/9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/9 & -1/36 & -1/18 & 1/6 & -1/6 & 0 & 0 & 1/4 & 0 \\ 1/9 & -1/36 & -1/18 & 0 & 0 & 1/6 & -1/6 & -1/4 & 0 \\ 1/9 & -1/36 & -1/18 & -1/6 & 1/6 & 0 & 0 & 1/4 & 0 \\ 1/9 & -1/36 & -1/18 & 0 & 0 & -1/6 & 1/6 & -1/4 & 0 \\ 1/9 & 1/18 & 1/36 & 1/6 & 1/12 & 1/6 & 1/12 & 0 & 1/4 \\ 1/9 & 1/18 & 1/36 & -1/6 & -1/12 & 1/6 & 1/12 & 0 & -1/4 \\ 1/9 & 1/18 & 1/36 & -1/6 & -1/12 & -1/6 & -1/12 & 0 & 1/4 \\ 1/9 & 1/18 & 1/36 & 1/6 & 1/12 & -1/6 & -1/12 & 0 & -1/4 \end{pmatrix}. \end{aligned} \tag{7}$$

The equilibrium values are chosen to be

$$e^{\mathrm{eq}} = -2\rho + 3\big(j_x^2 + j_y^2\big)/\rho, \tag{8a}$$

$$\varepsilon^{\mathrm{eq}} = \rho - 3\big(j_x^2 + j_y^2\big)/\rho, \tag{8b}$$

$$q_x^{\mathrm{eq}} = -j_x, \tag{8c}$$

$$q_y^{\mathrm{eq}} = -j_y, \tag{8d}$$

$$p_{xx}^{\mathrm{eq}} = \big(j_x^2 - j_y^2\big)/\rho, \tag{8e}$$

$$p_{xy}^{\mathrm{eq}} = j_x j_y/\rho \tag{8f}$$

and the fluid density and moment flux $\boldsymbol{J}(j_x, j_y)$ are obtained by

$$\rho = \sum_\alpha f_\alpha, \tag{9}$$

$$\boldsymbol{J}(j_x, j_y) = \sum_\alpha f_\alpha \boldsymbol{e}_\alpha. \tag{10}$$

The relaxation parameters s_α can be determined by a linear stability analysis [31], and in the present D2Q9 discrete velocity model, they are given as $s_2 = 1.64$, $s_3 = 1.54$, $s_5 = s_7 = 1.9$ and $s_8 = s_9 = 2/(6\upsilon/\delta_t + 1)$ and υ is the kinetic viscosity of the fluid.

The GLB model can include external forces, such as gravity. In the single-relaxation lattice BGK model, it is known that the whole force could be added to the lattice Boltzmann equation at once and half of the force is added in the velocity calculation

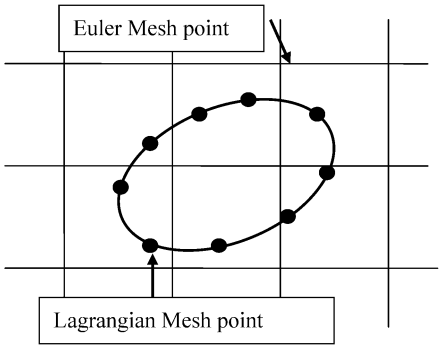


Fig. 1. Schematic diagram of the immersed boundary method. The thick line is the boundary with solid circle symbol representing the boundary points.

to consider the force effect on the momentum flux [32–34]. However, in the present GLB model, the particle evolution is based on the momentum. Therefore, to add a force $\boldsymbol{F}(\boldsymbol{r},t)$, it is preferable to execute it in two steps, adding one-half ($\boldsymbol{F}(\boldsymbol{r},t)\delta_t/2$) of the force before the relaxations and another half after the relaxations, in order to conserve mass up to the second order in the Chapman–Enskog analysis [35].

2.2. Immersed boundary method

The immersed boundary method (IBM) involves specifying the body force term in the flow solver to represent the presence of solid boundaries of an object in the flow domain without altering the underlying computational mesh. In the IBM, the boundary configuration is denoted by a set of the Lagrangian points, and the flow field covers both inside and outside the object and is represented by the Eulerian mesh points (see Fig. 1). The interaction between the immersed boundary and the fluid flow is modeled by injecting the force generated by the boundaries onto the fluid, and can be expressed by

$$\boldsymbol{F}(\boldsymbol{r},t) = \int_{\Omega} \boldsymbol{g}(\boldsymbol{X},t)\delta(\boldsymbol{r}-\boldsymbol{X})\,ds, \tag{11}$$

where $\delta(\boldsymbol{r}-\boldsymbol{X})$ is the Dirac delta function, $\boldsymbol{X}(X,Y)$ is the coordinate vector of the Lagrangian boundary points, and $\boldsymbol{g}(\boldsymbol{X},t)$ is the Lagrangian force density, which will be defined in the following section.

To smooth the Dirac delta function $\delta(\boldsymbol{r}-\boldsymbol{X})$, we adopt the distribution function D_{ij} (i, j are the indexes of the Eulerian mesh points in x- and y-directions, respectively) proposed by Peskin [19,36],

$$D_{ij}(\boldsymbol{r}_{ij}-\boldsymbol{X}_l) = \frac{1}{h^2}\delta_h\left(\frac{x_{ij}-X_l}{h}\right)\delta_h\left(\frac{y_{ij}-Y_l}{h}\right) \tag{12}$$

with

$$\delta_h(a) = \begin{cases} \frac{1}{4}(1+\cos(\frac{\pi a}{2})) & |a| \leqslant 2, \\ 0 & \text{otherwise}, \end{cases} \tag{13}$$

where $h=\delta_x$ is the mesh spacing. An important property of the function $\delta_h(a)$ is that its integral is equal to one. Using Eq. (12), Eq. (11) can be approximated by

$$\boldsymbol{F}(\boldsymbol{r}_{ij},t) = \sum_l \boldsymbol{g}(\boldsymbol{X}_l,t)D_{ij}(\boldsymbol{r}_{ij}-\boldsymbol{X}_l)\Delta s_l, \tag{14}$$

where Δs_l is the arc length of the boundary element.

2.3. Evaluation of force by momentum exchange of particles

To obtain the distribution functions on the boundary points, the two-dimensional Lagrangian interpolated polynomials formed by the Eulerian (Cartesian) mesh points are used. Thus, the distribution function on the Lagrangian boundary point can be approximated by

$$f_\alpha(\boldsymbol{X}_l,t) = \sum_{ij}\left(\prod_{k=1,k\neq i}^{i\max}\frac{X_l-x_{kj}}{x_{ij}-x_{kj}}\right)\left(\prod_{m=1,m\neq j}^{j\max}\frac{Y_l-y_{im}}{y_{ij}-y_{im}}\right)f_\alpha(\boldsymbol{r}_{ij},t), \tag{15}$$

where i max and j max are the maximum numbers of the mesh points in the x- and y-directions, respectively. It should be stressed that the above polynomials can be either centered around the boundary (interpolation) or ended at boundary (extrapolation). From

a physical point of view, for the solid boundary, it is more reasonable to use mesh points inside the fluid domain to do interpolation/extrapolation. Likewise, if the boundary is a fluid interface, one would extrapolate to the boundary from both sides. In the cases considered in the present work, extrapolation using the third order polynomials is used.

Note that Eq. (15) is applied to all lattice velocity directions. By using the bounce back rules for the distribution functions of all directions, the non-slip boundary condition is satisfied and we have a new set of density distribution functions on the boundary points [12,37–39]

$$f_\beta(\boldsymbol{X}_l, t+\delta_t) = f_\alpha(\boldsymbol{X}_l, t) - 2w_\alpha \rho \frac{\boldsymbol{e}_\alpha \cdot \boldsymbol{U}_{BC}}{c_s^2}, \tag{16}$$

where β denotes the opposite direction of α, i.e., $\boldsymbol{e}_\beta = -\boldsymbol{e}_\alpha$; $\boldsymbol{U}_{BC}(= \boldsymbol{U}_p + \boldsymbol{\Omega}_p \times (\boldsymbol{X}_l - \boldsymbol{X}_p))$ is the velocity of the boundary with $\boldsymbol{U}_p$ and $\boldsymbol{\Omega}_p$ representing the translational and angular velocities of the object, respectively, and $\boldsymbol{X}_p$ is the mass center of the object; $w_\alpha = 1/9$ for $\alpha \in \{1, \ldots, 4\}$ and $w_\alpha = 1/36$ for $\alpha \in \{5, \ldots, 8\}$ and $c_s = 1/\sqrt{3}$ for the D2Q9 model.

The force density at the boundary point can be calculated by the momentum exchange of particles, that is,

$$\boldsymbol{g}(\boldsymbol{X}_l, t) = \sum_\beta e_\beta \big[f_\beta(\boldsymbol{X}_l, t) - f_\alpha(\boldsymbol{X}_l, t) \big]. \tag{17}$$

The hydrodynamic force exerted on the boundary point is the balance force of $\boldsymbol{g}(\boldsymbol{X}_l, t)$:

$$\boldsymbol{F}(\boldsymbol{X}_l, t) = -\boldsymbol{g}(\boldsymbol{X}_l, t) \tag{18}$$

and the total force $\boldsymbol{F}_T$ and the torque $\boldsymbol{T}_T$ acting on the object are obtained as

$$\boldsymbol{F}_T = \sum_l \boldsymbol{F}(\boldsymbol{X}_l, t) \Delta s_l, \tag{19}$$

$$\boldsymbol{T}_T = \sum_l (\boldsymbol{X}_l - \boldsymbol{X}_p) \times \boldsymbol{F}(\boldsymbol{X}_l, t) \Delta s_l. \tag{20}$$

Having the force and torque on the object, the motion of the object can be determined by:

$$M_p \frac{d\boldsymbol{U}_p}{dt} = \boldsymbol{F}_T, \tag{21}$$

$$I_p \frac{d\boldsymbol{\Omega}_p}{dt} = \boldsymbol{T}_T, \tag{22}$$

where M_p and I_p are the mass and the moment of inertia of the object, respectively.

An important advantage of this momentum exchange-based immersed boundary method is that, it allows the object to move arbitrarily on the computational domain with a simple consideration of interactions between the boundary and the fluid domain. It should be stressed that the Lagrangian boundary points must be denser than the underlying Cartesian mesh point in order to avoid leakage. The numerical implementation of the GLB model with the momentum exchange-based IBM is summarized below:

- Step 1: Compute moments $\boldsymbol{R}(\boldsymbol{r}, t)$ at all fluid points from the fluid density and velocity (initially setting $\boldsymbol{R} = \boldsymbol{R}^{\mathrm{eq}}$) and modify the moment flux at all fluid points by $\boldsymbol{J}' = \boldsymbol{J} + \frac{1}{2}\boldsymbol{F}\delta_t$.
- Step 2: Make relaxations to the moments $\boldsymbol{R}$ by Eq. (6a) and modify the moment flux by $\boldsymbol{J}'' = \boldsymbol{J}' + \frac{1}{2}\boldsymbol{F}\delta_t$ again.
- Step 3: Compute the post-collision density distribution functions $f_\alpha(\boldsymbol{r}, t)$ by Eq. (6b) at all fluid points.
- Step 4: Stream $f_\alpha(\boldsymbol{r}, t)$.
- Step 5: Compute $f_\alpha(\boldsymbol{r}, t)$ by Eq. (15) at all boundary points and make bounce-back according to Eq. (16). The Lagrangian force $\boldsymbol{g}(\boldsymbol{X}, t)$ is obtained by Eq. (17) and then redistributed into the flow field by Eq. (14).
- Step 6: Update the boundary positions of the objects by Eqs. (21) and (22).
- Step 7: Repeat Steps 1 to 7 until convergence is reached.

3. Numerical results and discussion

3.1. The flow over a line of circular cylinders placed at the middle of a straight channel

To examine the accuracy of the present method, we first simulate the flow over a line of circular cylinders placed at the middle of a straight channel. This case has been investigated by Aidun and Lu [26], and Inamuro et al. [27] using both the FEM and the LBM. We took the case where the cylinder is fixed and the channel walls move tangentially with the same flow condition as used by Aidun and Lu [26], and Inamuro et al. [27]. Specifically, the velocity of the channel walls is set as $U_w = -0.01$–(-0.04), the mesh size of 128×128 is used and the Reynolds number is $Re = |U_w| D_c/\upsilon = 1$ (D_c is the diameter of the cylinder). The periodic boundary condition is used at the inlet and outlet of the channel.

Table 1
Comparison of present results with those of Aidun and Lu [26], and Inamuro et al. [27] for the flow over a line of circular cylinders in a straight channel. Error is evaluated based on the result of the FEM

Case	Method	$\tau(s_8, s_9)$	$U_w(U_p)$	f_c	Error (%)
I $(D_c = 10.8\delta_x)$	FEM [26]	–	−0.004	0.966	–
	LBM [26]	1.796		1.022	5.8
	LBM [27]	1.796		1.053	9.0
	Present	1.796		1.039	7.6
II $(D_c = 20.8\delta_x)$	FEM [26]	–	−0.002	1.158	–
	LBM [26]	1.748		1.229	6.1
	LBM [27]	1.748		1.251	8.0
	Present	1.748		1.242	7.2
III $(D_c = 60.8\delta_x)$	FEM [26]	–	−0.001	2.067	–
	LBM [26]	2.324		2.054	−0.6
	LBM [27]	2.324		2.093	1.3
	Present	2.324		2.072	0.24
	LBM [27][a]	2.324	0.01	2.094	1.3
	Present[a]	2.324		2.072	0.24

[a] The results of a moving circular cylinder between the fixed channel walls.

Table 2
Comparison of the drag coefficient, the recirculation length obtained by the present method with previous studies based on the FDM [28] and the LBM [15,18]. Error is evaluated based on the result of the FDM

Case	Method	C_D	Error (%)	L_W	Error (%)	θ_s (degree)	Error (%)
$Re = 20$	FDM [28]	2.045	–	1.88	–	43.7	–
	LBM [15]	2.152	5.2	1.84	−2.1	42.96	−1.7
	LBM [18]	2.111	3.2	1.92	2.1	42.79	−2.1
	Present	2.144	4.8	1.89	0.5	42.95	−1.7
$Re = 40$	FEM [28]	1.522	–	4.69	–	53.8	–
	LBM [15]	1.499	−1.5	4.49	−4.3	52.84	−1.8
	LBM [18]	1.574	3.4	4.51	−3.8	53.8	0
	Present	1.589	4.4	4.52	−3.8	53.86	0.1

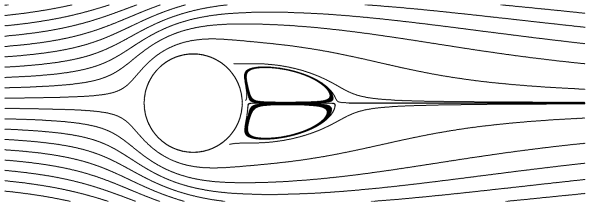
We compared the results of the dimensionless force $f_c = |\boldsymbol{F}_T|/(\pi\rho U_w^2 D_c)$ acting on the cylinder with those of Aidun and Lu [26] and Inamuro et al. [27] in Table 1. It should be mentioned that, in the LBM used by Aidun and Lu [26] and Inamuro et al. [27], the cylinder boundary surface is approximated by the zig-zag staircase grids. From Table 1, it is found that the present results agree well with those of Aidun and Lu [26] and Inamuro et al. [27], and for $D_c = 60.8\delta_x$, the relative error to the FEM result [26] is within 0.24%. To show the accuracy of the present method for the moving boundary problem, like Inamuro et al. [27], we also calculated the dimensionless f_c of a moving cylinder with $D_c = 60.8\delta_x$ and the velocity $U_p = -U_w$ between the fixed channel walls. The present result, together with that of Inamuro et al. [27], was shown in the bottom two rows of Table 1. Obviously, our result agrees very well with that of Inamuro et al. [27] and is consistent with those of a fixed cylinder between moving channel walls.

3.2. *Steady flows past a circular cylinder at* $Re = 20$ *and* 40

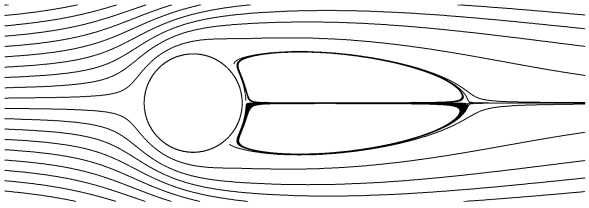
The second numerical validation of the present method is the flow past a circular cylinder. This flow has been studied extensively and a number of numerical and experimental results exist in the literature. The problem is very attractive because the flow behavior depends on the Reynolds number ($Re = U_\infty D_c/\upsilon$) and is not easy to simulate accurately using Cartesian grids.

Present simulations were performed at $Re = 20$ and 40 and results were compared with established numerical data [28,29]. The flow domain is set by $L \times W = 40D_c \times 40D_c$ square domain with a uniform mesh size of 800×800. To minimize the effect of the outer boundary on the numerical results, the circular cylinder is placed at $(2L/5, W/2)$ with diameter of $D_c = 20\delta_x$. The free steam velocity U_∞ is set to be 0.1. Fig. 2 shows the streamlines for the cases of $Re = 20$ and 40, respectively. As shown in Fig. 2, a pair of symmetric vortices develops behind the cylinder at these two Reynolds numbers. The drag coefficients ($C_D = F_x/(1/2)\rho U_\infty^2 D_c$) and the length of the recirculation zone ($L_W = 2L/D_c$) were compared with the established results of FDM [28] and LBM [15,18] in Table 2. Basically, the present results agree well with those in the literature. It should be stressed that the present results can be improved by using the finer meshes. An alternative way to obtain accurate results can be sought for the mesh-refinement technique [10]. Since the present work focuses on the way to evaluate the force at the boundary point, related mesh-refinement work is not included.

Fig. 3 shows the vorticity distributions over the circular cylinder surface at $Re = 20$ and 40, respectively. The numerical results



(a) Re = 20



(b) Re = 40

Fig. 2. Streamlines of the flow around a circular cylinder at $Re = 20$ and 40.

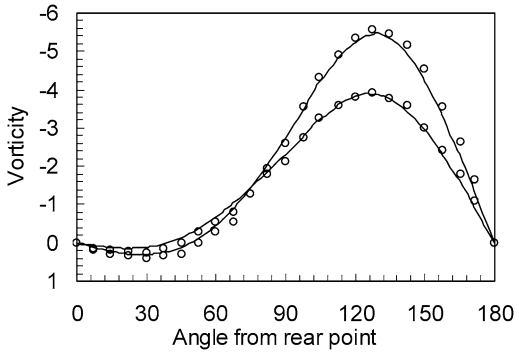


Fig. 3. Vorticity distribution over the cylinder surface for the flow around a circular cylinder at $Re = 20$ and 40 (○-results of the FDM [29]; — -present results).

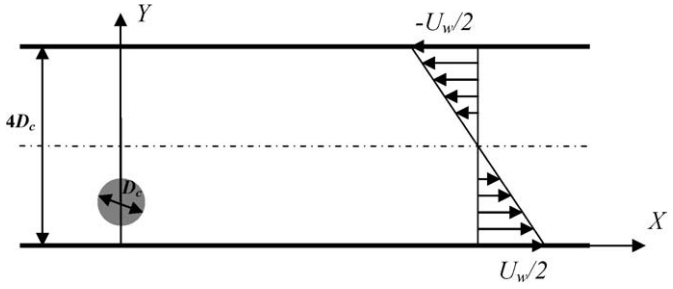


Fig. 4. Schematic diagram of a neutrally buoyant particle in a simple shear flow between walls.

of the FDM [29] were also included in this figure for comparison. The accuracy of the present method is well demonstrated in the figure.

3.3. The motion of a neutrally buoyant particle in a linear shear flow

The motion of a neutrally buoyant two-dimensional circular particle in a viscous fluid has been extensively studied by many researchers using the LBM [23] and the FEM [30]. Therefore, the comparison of present results with those from other numerical methods will serve as a further validation to the present method for moving boundary problems. A schematic diagram of the neutrally buoyant particle in a linear shear flow is depicted in Fig. 4, and the parameters used in the simulation are as follows: the height and length of the channel are $H = 1$ and $L = 25$, respectively; the diameter of the particle is $D = 0.25H$; the upper and lower walls of the channel are moving at velocities of $U_w/2 = 1/20$ in opposite directions; the particle density is equal to that of the fluid ($\rho_s = \rho = 1$) and the fluid bulk Reynolds number is $Re = U_w H/\upsilon = 40$. Numerical simulation of this problem was based

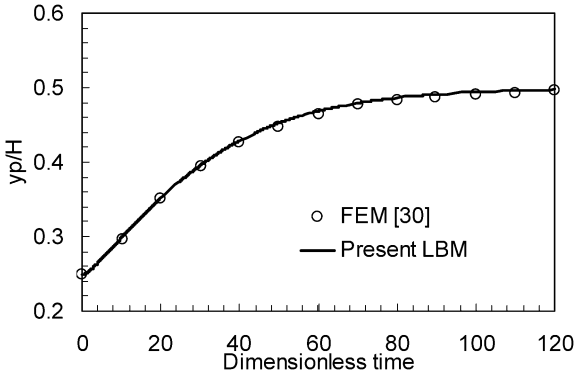


Fig. 5. The lateral migration of the particle with dimensionless time $t^* = tU_w/H$.

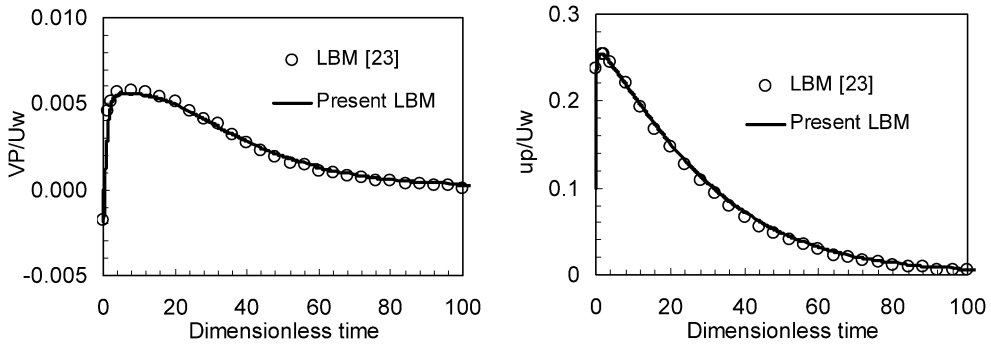


Fig. 6. Particle translational velocities vary with dimensionless time $t^* = tU_w/H$ time.

on a Cartesian mesh of 2000×80 with the periodic boundary condition used at the inlet and outlet of the channel. Initially, the particle is set at the position of $y_{t=0} = 0.25H$ above the bottom wall of the channel and at rest.

The lateral migration of the particle using the present method was compared with the results from the FEM [30] in Fig. 5. It is found from Fig. 5 that the cylinder migrates to the centerline where it is at equilibrium state. As shown in Fig. 5, a very good agreement is achieved between the present results and those of the FEM [30]. Fig. 6 shows the two components of the particle translational velocity obtained by the present method and the LBM of Feng and Michaelides [23]. A perfect agreement is observed with the results obtained by the two methods.

3.4. Sedimentation of two circular particles in a viscous fluid

The case of two circular particles falling in a viscous fluid was considered to further validate our method. An important mechanism in this case is the well-known "drafting, kissing and tumbling" or DKT motion [40]. The leading particle creates a wake of low pressure, and the trailing particle is caught in its wake and falls faster than the leading one. This phenomenon is called drafting. The increased speed of the trailing particle impels a kissing contact with the leading one. In kissing contact, the two particles form a long body with the line of center along the stream. This state is unstable, and as a result, the particles tumble under the influence of a couple which tries to bring them in broadside stable state.

For the solution of this problem, we chose a 2 cm (x-direction) $\times$ 8 cm (y-direction) channel filled with a viscous fluid of density $\rho_f = 1$ g/cm^3 and viscosity 0.1 g/cm s. Inside the channel, two circular particles of density $\rho_s = 1.01$ g/cm^3 and radius $a = 0.01$ cm initially located at (0.999 cm, 7.2 cm) (particle 1) and (1 cm, 6.8 cm) (particle 2) fall under gravity. This case is chosen to be identical to that presented by Feng and Michaelides [23]. In our simulation, the grid used is 201×801. In this case, the force on a particle includes the gravity/buoyancy force, the particle–particle/wall interaction forces as well as the hydrodynamic force mentioned in Section 2.3. The particle–particle/wall interaction forces are added to keep the physical behaviors of the particles and prevent particles from penetrating into each other's boundaries and the fluid container. For the simulation of the sedimentation problem presented here, we choose the Lennard-Jones potential ($E_{pp} = 0.4\varepsilon\lfloor(2a/R_{ip,jp})^{12} - (2a/R_{ip,jp})^6\rfloor$, where ip and jp index the particles) to model the particle–particle/wall interactions, and we have

$$F_{ip}^{p-p} = \begin{cases} 0 & R_{ip,jp} > 2a + \zeta, \\ 2.4\varepsilon \sum_{jp=1,\, jp\neq ip}^{N} \left[2\left(\frac{2a}{R_{ip,jp}}\right)^{14} - \left(\frac{2a}{R_{ip,jp}}\right)^{8}\right] \frac{\boldsymbol{R}_{ip} - \boldsymbol{R}_{jp}}{(2a)^2} & R_{ip,jp} \leqslant 2a + \zeta, \end{cases} \tag{23}$$

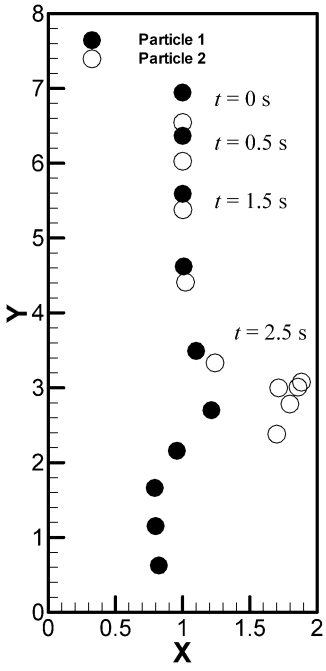


Fig. 7. Sedimentation pf two circular particles in a channel at different time stages.

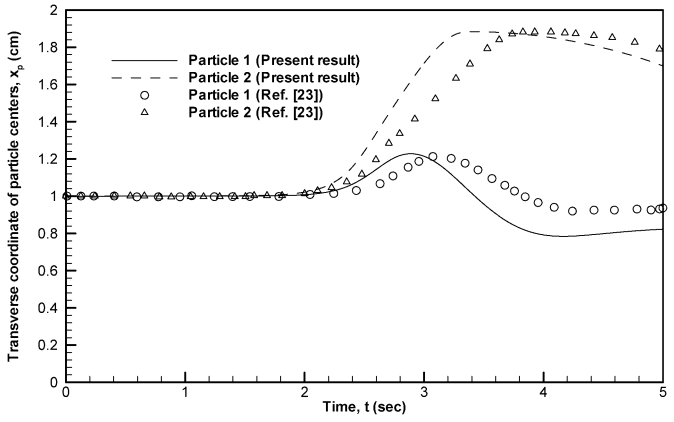


Fig. 8. Transverse coordinates of the centers of the two particles.

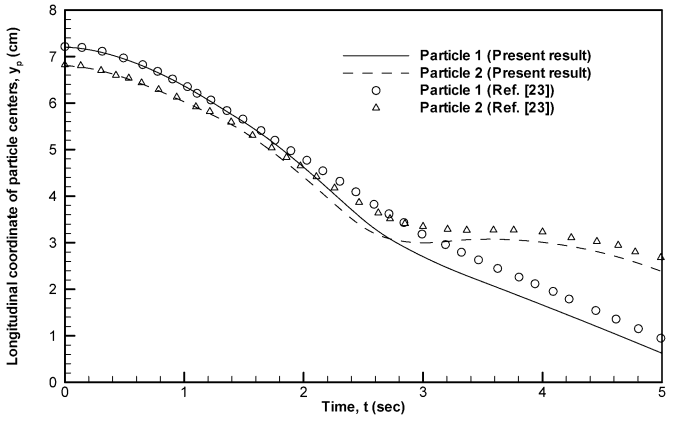


Fig. 9. Longitudinal coordinates of the centers of the two particles.

$$F_{ip}^{p-w} = \begin{cases} 0 & R_{ip,j} > 2a + \zeta, \\ 2.4\varepsilon \sum_{j=1}^{j\max} \left[2\left(\frac{a}{R_{ip,j}}\right)^{14} - \left(\frac{a}{R_{ip,j}}\right)^{8}\right] \frac{\boldsymbol{R}_{ip} - \boldsymbol{r}_{wj}}{(a)^2} & R_{ip,j} \leqslant 2a + \zeta, \end{cases} \tag{24}$$

where $\varepsilon = a^2$, $R_{ip,jp} = |\boldsymbol{R}_{ip} - \boldsymbol{R}_{jp}|$, $R_{ip,w} = |\boldsymbol{R}_{ip} - \boldsymbol{r}_{wj}|$, r_w represents the wall positions and ζ is the threshold and is set to one lattice unit in the present simulations. Hence, the total force exerted on the ipth particle can be written as:

$$\boldsymbol{F}_T = \left(1 - \frac{\rho_f}{\rho_s}\right) M_{ip}\boldsymbol{g} + \boldsymbol{F}_{ip}^{p-p/w} + \sum_l \boldsymbol{F}_{ip}(\boldsymbol{X}_l, t)\Delta s_l. \tag{25}$$

Fig. 7 shows the positions of two particles at different times and the DKT motion is obviously observed in this figure. Figs. 8 and 9 compared the instantaneous transverse and longitudinal coordinates of the centers of the two particles with the results of Feng and Michaelides [23]. As expected, the comparison is found to be good up to the kissing of particles. It is known that the tumbling phenomenon is the realization of instability and different modes of tumbling have been reported in the literature [41]. Thus an exact agreement after kissing may not be expected. As shown in Figs. 7–9, one can observe that particle 1 trails the particle 2 initially and these two particles keep a steady distance up to about $t = 0.8$ s. After this time, the particles move closer, and at $t \approx 1.4$ s, the particles enter the "kissing" motion. At the same time, the centers of the two particles start to slowly deviate from the channel center and then, the particles move closer and tumble up to $t = 2.4$ s.

4. Conclusions

A momentum exchange-based immersed boundary-lattice Boltzmann method is developed to simulate the incompressible viscous flows with moving and complex boundaries. The present method preserves the merits of the LBM and the IBM by using two unrelated computational meshes, an Eulerian mesh for the flow domain and a Lagrangian mesh for the moving boundaries in the flow. The simplicity of the Eulerian mesh facilitates the numerical implementation of the LBM, and the generality of the Lagrangian mesh makes it easy to handle complex boundaries. The non-slip boundary condition is satisfied directly by the bounce-back condition of the particle density distribution functions, which are interpolated by the Lagrangian polynomials from the underlying Eulerian mesh. The force generated by the boundaries is calculated by the momentum exchange of the particles.

The present method has been validated by simulating four different cases (a flow over a line of circular cylinders, a steady flow past a circular cylinder, a neutrally buoyant particle migration in a linear shear flow and the flow of two interacting circular particles settling in a channel) and comparing the results with those of other methods. The numerical evidences showed the good capability of the present method for simulation of incompressible viscous flows. However, we have to indicate that although the present scheme greatly improves the flexibility of the LBM in simulating the moving boundary problems, there is no improvement on the accuracy. In other words, it keeps the same order of accuracy as the conventional LBM.

References

[1] G.R. McNamara, G. Zanetti, Phys. Rev. Lett. 61 (1988) 2332.
[2] Y.H. Qian, D. d'Humières, P. Lallemand, Europhys. Lett. 17 (1992) 479.
[3] S. Chen, H. Chen, D. Martinez, W.H. Matthaeus, Phys. Rev. Lett. 67 (1991) 3776.
[4] S. Chen, S.P. Dawson, G.D. Doolen, D.R. Janecky, A. Lawniczak, Comput. Chem. Eng. 19 (1995) 617.
[5] S. Chen, G.D. Doolen, Annu. Rev. Fluid Mech. 30 (1998) 329.
[6] S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Oxford Univ. Press, New York, 2001.
[7] H. Yu, L.-S. Luo, S.S. Girimaji, J. Comput. Eng. Sci. 3 (2002) 73.
[8] X. He, Q. Zou, L.-S. Luo, M. Dembo, J. Stat. Phys. 87 (1997) 115.
[9] R. Cornubert, D. d'Humières, D. Levermore, Physica D 47 (1991) 241.
[10] O. Filippova, D. Hänel, J. Comput. Phys. 147 (1998) 219.
[11] R. Mei, L.-S. Luo, W. Shyy, J. Comput. Phys. 155 (1999) 307.
[12] M. Bouzidi, M. Firdaouss, P. Lallemand, Phys. Fluids 13 (2001) 3452.
[13] A.J.C. Ladd, J. Fluid Mech. 271 (1994) 285.
[14] X. He, G.D. Doolen, Phys. Rev. E 56 (1997) 434.
[15] X. He, G.D. Doolen, J. Comput. Phys. 134 (1997) 306.
[16] C. Shu, Y.T. Chew, X.D. Niu, Phys. Rev. E 64 (2001) 045701.
[17] C. Shu, X.D. Niu, Y.T. Chew, Phys. Rev. E 65 (2002) 036708.
[18] X.D. Niu, Y.T. Chew, C. Shu, J. Comput. Phys. 188 (2003) 176.
[19] C.S. Peskin, J. Comput. Phys. 25 (1977) 220.
[20] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, J. Periaux, J. Comput. Phys. 169 (2001) 363.
[21] K. Höfler, S. Schwarzer, Phys. Rev. E 61 (2000) 7146.
[22] N.A. Patankar, P. Singh, D.D. Joseph, R. Glowinski, T.-W. Pan, Int. J. Multiphase Flow 26 (2000) 1509.
[23] Z.G. Feng, E.E. Michaelides, J. Comput. Phys. 195 (2004) 602.
[24] Z.G. Feng, E.E. Michaelides, J. Comput. Phys. 202 (2005) 20.
[25] D. d'Humières, Generalized lattice Boltzmann equations, in: B.D. Shizgal, D.P. Weaver (Eds.), Rarefied Gas Dynamics: Theory and Simulations, Progress in Astronautics and Aeronautics, vol. 159, AIAA, Washington, DC, 1992, p. 450.
[26] C.K. Aidun, Y. Lu, J. Stat. Phys. 81 (1995) 49.
[27] T. Inamuro, K. Maeba, F. Ogino, Int. J. Multiphase Flow 26 (2000) 1981.
[28] S.C.R. Dennis, G.Z. Chang, J. Fluid Mech. 42 (1980) 471.
[29] B. Fornberg, J. Fluid Mech. 98 (1980) 819.
[30] J. Feng, H.H. Hu, D.D. Joseph, J. Fluid Mech. 277 (1994) 271.
[31] P. Lallemand, L.-S. Luo, Phys. Rev. E 61 (2000) 6546.
[32] I. Ginzburg, P.M. Adler, J. Phys. II 4 (1994) 191.
[33] Z. Guo, C. Zheng, B. Shi, Phys. Rev. E 65 (2002) 046308.
[34] A.J.C. Ladd, R. Verberg, J. Stat. Phys. 104 (2001) 1191.
[35] P. Lallemand, L.-S. Luo, Phys. Rev. E 68 (3) (2003) 036706.
[36] C.S. Peskin, Acta Numer. 11 (2002) 479.
[37] L.S. Luo, Phys. Rev. Lett. 81 (1998) 1618.
[38] L.S. Luo, Phys. Rev. E 62 (2000) 4982.
[39] A.J.C. Ladd, D. Frenkel, Phys. Fluids A 2 (1990) 1921.
[40] A. Fortes, D.D. Joseph, T.S. Lundgren, J. Fluid Mech. 177 (1987) 467.
[41] N.A. Patankar, A formulation for fast computations of rigid particulate flows, Annual Research Briefs-2001, Center for Turbulent Research, Stanford University, 2001.