A comparative study of immersed-boundary and interpolated bounce-back methods in LBE

Yan Peng and Li-Shi Luo*

Department of Mathematics and Statistics,
Center for Computational Sciences,
Old Dominion University,
Norfolk, Virginia 23529, USA
E-mail: ypeng@odu.edu E-mail: lluo@odu.edu
*Corresponding author

Abstract: The Interpolated Bounce-Back (IBB) method and Immersed-Boundary (IB) method are compared for fluid-solid boundary conditions in the Lattice Boltzmann Equation (LBE) in terms of their numerical accuracy and computational efficiency. We carry out simulations for the flow past a circular cylinder asymmetrically placed in the channel in two dimensions with the Reynolds number Re = 20 and 100, corresponding to steady and unsteady flows, respectively. The results obtained by the LBE method are compared with the existing data. We observe that the LBE with either the IBB or IB methods for the no-slip boundary conditions exhibits a second-order rate of convergence. While the computational cost for both methods are comparable, the interpolated bounce-back method is more accurate than the immersed-boundary method with the same mesh size. Consequently the IBB method is more efficient computationally, while the IB method is easier to implement.

Keywords: Lattice Boltzmann equation; LBE; immersed boundary method; interpolated bounce-back.

Reference to this paper should be made as follows: Peng, Y. and Luo, L-S. (2008) 'A comparative study of immersed-boundary and inerpolated bounce-back methods in LBE', *Progress in Computational Fluid Dynamics*, Vol. 8, Nos. 1–4, pp. 156–167.

Biographical notes: Y. Peng is a Research Assistant Professor in the Dept. of Math. and Stat. at Old Dominion University. She completed her PhD Degree from the National University of Singapore in 2005. Since 1999, her research focus has been the lattice Boltzmann equation and its applications.
L-S. Luo is currently the Richard F. Barry Jr. Distinguished Endowed Professor at Department of Mathematics and Statistics, Old Dominion University. He obtained his PhD Degree in Physics from Georgia Institute of Technology in 1993. Before joining ODU in 2004, he worked in Los Alamos National Laboratory, Worcester Polytechnic Institute, ICASE/NASA Langley Research Center, and National Institute of Aerospace. His main research interests include kinetic theory, nonequilibrium flows, complex fluids, kinetic/mesoscopic methods for computational fluid dynamics, and scientific computing.

1 Introduction

In recent years, the Lattice Boltzmann Equation (LBE) has become an efficient alternative for Direct Numerical Simulations (DNS) of complex fluids, such as colloidal suspensions and interfacial flows (cf. Yu et al. (2003) and references therein). The LBE has been proved to be valid for solving the incompressible Navier-Stokes equations (Junk et al., 2005), it has also been shown to be efficient and competitive in some circumstances such as complex fluid flows through porous media (Ginzburg and d'Humières, 2003; Pan et al., 2006) and non-spherical particulate suspensions in fluid flows (Ladd, 1994a, 1994b; Ladd and Verberg, 2001; Qi and Luo, 2002, 2003; Qi et al., 2002). In the LBE simulations of complex flows, accurate treatment of fluid-boundary conditions is imperative. Recent studies show that interpolated bounce-back method is accurate and effective (Bouzidi et al., 2001; Ginzburg and d'Humières, 2003; Pan et al., 2006). However, it should be stressed that the second-order accuracy of the bounce-back type boundary conditions can only be fully realised through the LBE with Multiple-Relaxation-Time (MRT) collision model (Ginzbourg and Adler, 1994; Ginzburg and d'Humières, 1996, 2003; Bouzidi et al., 2001; Pan et al., 2006).

The idea of the IBB method (Bouzidi et al., 2001; Ginzburg and d'Humières, 2003; Pan et al., 2006) to treat fluid-solid boundary conditions is relatively simple. The LBE method usually employs uniform Cartesian meshes in two or three dimensions. With a simple bounce-back method, arbitrary shape of a solid object is crudely approximated by those grid points closest to the boundary geometry. Thus, a smooth boundary in 2D is approximated by zig-zag staircase. While the zig-zag approximation is very easy to implement, it is inaccurate in preserving the boundary geometric integrity. By using interpolations (Bouzidi et al., 2001; Ginzburg and d'Humières, 2003; Pan et al., 2006), the accuracy of geometric representation is greatly improved. However, the interpolations also increase the complexity of the LBE method in terms of programming, especially for moving boundary problems.

The IB method due to Peskin (1977, 2002) and Lai and Peskin (2000) is a simple method to treat boundary conditions in bio-fluid systems. It has undergone various modifications and improvements to become a useful method for problems of fluid-structure interactions (cf. Peskin, 2002; Lai and Peskin, 2000; Mittal and Iaccarino, 2005; Li and Ito, 2006). The essence of the immersed boundary method is to use Eulerian Cartesian mesh for fluid equations and a set of Lagrangian marker points for boundaries. The set of marker points is completely independent of the Eulerian mesh, and the interaction between the marker points and the fluid flow are realised through a smoothed approximation to the Dirac δ-function (Peskin, 1977, 2002). The independence of a geometric boundary defined by the Lagrangian markers to the underlying Eulerian Cartesian mesh for fluid flow makes the IB method very easy to implement, and this is the main attraction and advantage of the IB method.

Since the LBE method is an Eulerian method with uniform Cartesian meshes, it is a natural candidate of fluid solver for the IB method. Only very recently the IB method has been incorporated into the lattice Boltzmann equation to simulate rigid particular suspensions in fluid flow (Feng and Michaelides, 2004, 2005). Due to its infancy, much is still untested about the IB-LBE method. In this work we intend to investigate the IB-LBE method through a comparative study between the IB-LBE method and the IBB-LBE method. We will compare these two methods in the simulations of a laminar flow past a cylinder asymmetrically placed in a channel in two-dimensions with the Reynolds number Re $= 20$ and 100 (Schäfer and Turek, 1996).

The remainder of this paper is organised as follows. In Section 2 we provide succinct descriptions of the LBE method and the IB method. We provide brief descriptions for the LBE with MRT collision model (d'Humières, 1992; Lallemand and Luo, 2000; d'Humières et al., 2002; Lallemand and Luo, 2003), which is superior over the lattice Bhatnagar-Gross-Krook (BGK) equation (Chen et al., 1992; Qian et al., 1992) in terms of numerical stability, computational efficiency, (Lallemand and Luo, 2000; d'Humières et al., 2002; Lallemand and Luo, 2003), and accuracy of boundary conditions (Bouzidi et al., 2001; Ginzburg and d'Humières, 2003; Pan et al., 2006). We also discuss in sufficient details the interpolated bounce-back boundary conditions (Bouzidi et al., 2001). We next describe the IB method and its implementation in the LBE through a body force. We also discuss the use of stretched grids in the LBE to enhance computational efficiency (He et al., 1996, 1997; He and Luo, 1997). In Section 2 we discuss in details the results obtained by using the IB-LBE and IBB-LBE methods for the benchmark flow past a cylinder in a 2D channel. We observe that in general the IBB-LBE method produces more accurate results than the IB-LBE method, while both methods have a comparable computational efficiency and exhibit a second-order rate of convergence. Our results are particularly interesting because previous results (Feng and Michaelides, 2004, 2005) show that the IB-LBE with the BGK collision model has a first-order rate of convergence. Finally, we summarise our results and conclude the paper in Section 4.

2 Numerical methods

2.1 Lattice Boltzmann Equation

The LBE with multiple-relaxation-time collision model is written as (d'Humières, 1992; Lallemand and Luo, 2000; d'Humières et al., 2002; Lallemand and Luo, 2003):

$$\begin{aligned}&\mathbf{f}(\boldsymbol{x}_j + \boldsymbol{c}\delta_t, t_n + \delta_t) - \mathbf{f}(\boldsymbol{x}_j, t_n)\\ &\quad = -\mathsf{M}^{-1} \cdot \widehat{\mathsf{S}} \cdot [\mathbf{m}(\boldsymbol{x}_j, t_n) - \mathbf{m}^{(\text{eq})}(\boldsymbol{x}_j, t_n)],\end{aligned} \tag{1}$$

where the bold-font symbols denote Q-tuple vectors, and Q is the total number of discrete velocities:

$$\begin{aligned}&\mathbf{f}(\boldsymbol{x}_j + \boldsymbol{c}\delta_t, t_n + \delta_t) = (f_0(\boldsymbol{x}_j, t_n + \delta_t),\\ &\quad f_1(\boldsymbol{x}_j + \boldsymbol{c}_1\delta_t, t_n + \delta_t) \ldots, f_b(\boldsymbol{x}_j + \boldsymbol{c}_b\delta_t, t_n + \delta_t))^{\mathsf{T}},\\ &\mathbf{f}(\boldsymbol{x}_j, t_n) = (f_0(\boldsymbol{x}_j, t_n), f_1(\boldsymbol{x}_j, t_n), \ldots, f_b(\boldsymbol{x}_j, t_n))^{\mathsf{T}},\end{aligned}$$

$b = (Q - 1)$ is the number of non-zero discrete velocities, and $\mathbf{f}$, $\mathbf{m}$ and $\mathbf{m}^{(\text{eq})}$ represent the vectors whose components are the distribution functions, the velocity moments, and the equilibrium moments, respectively. The $Q \times Q$ matrix M maps $\mathbf{f}$ to $\mathbf{m}$, i.e.,

$$\mathbf{m} = \mathsf{M} \cdot \mathbf{f}, \quad \mathbf{f} = \mathsf{M}^{-1} \cdot \mathbf{m},$$

and $\widehat{\mathsf{S}}$ is a $Q \times Q$ diagonal matrix of the relaxation rates $\{s_i | 0 < s_i < 2\}$, i.e.,

$$\widehat{\mathsf{S}} = \text{diag}(s_0, s_1, \cdots, s_b).$$

In this work, we use the nine-velocity model in two dimensions (D2Q9 model). The discrete velocities are: $\boldsymbol{c}_0 = (0,0)$, $\boldsymbol{c}_1 = (1,0)c$, $\boldsymbol{c}_2 = (0,1)c$, $\boldsymbol{c}_3 = (-1,0)c$, $\boldsymbol{c}_4 = (0,-1)c$, $\boldsymbol{c}_5 = (1,1)c$, $\boldsymbol{c}_6 = (-1,1)c$, $\boldsymbol{c}_7 = (-1,-1)c$, and $\boldsymbol{c}_8 = (1,-1)c$, where $c := \delta_x/\delta_t$. Corresponding to the nine discrete velocities $\{c_i\}$, there are nine moments:

$$\mathbf{m} = (\delta\rho, e, \varepsilon, j_x, q_x, j_y, q_y, p_{xx}, p_{xy})^{\mathsf{T}}. \tag{2}$$

Among these nine moments, the density fluctuation $\delta\rho$ and the momentum $\boldsymbol{j} := (j_x, j_y)$ are conserved moments for the athermal LBE model considered here, the other six non-conserved moments, e, ε, $\boldsymbol{q} = (q_x, q_y)$, p_{xx}, and p_{xy} are related to the energy, the energy square, the heat flux, and the diagonal and off-diagonal components of the stress tensor, respectively. The equilibria of the conversed moments are themselves, and that of the non-conserved moments are functions of the conserved ones. With the low Mach number approximation, we use the following equilibria for the non-conserved moments:

$$e^{(\mathrm{eq})} = -2\delta\rho + 3\boldsymbol{j}\cdot\boldsymbol{j}, \quad \varepsilon^{(\mathrm{eq})} = \delta\rho - 3\boldsymbol{j}\cdot\boldsymbol{j}, \tag{3a}$$

$$q_x^{(\mathrm{eq})} = -j_x, \quad q_y^{(\mathrm{eq})} = -j_y, \tag{3b}$$

$$p_{xx}^{(\mathrm{eq})} = j_x^2 - j_y^2, \quad p_{xy}^{(\mathrm{eq})} = j_x j_y. \tag{3c}$$

With the orderings of the discrete velocities $\{\boldsymbol{c}_i\}$ and the corresponding moments $\{m_i\}$ given above, the transform matrix M in Equation (1) is:

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

In the equilibria of Equation (3), we have used the incompressibility approximation, i.e., the density $\rho = \rho_0 + \delta\rho$, where $\rho_0 = 1$ is the mean density, and $\boldsymbol{j} \approx \rho_0\boldsymbol{u}$. We only deal with the density fluctuation $\delta\rho$ in simulation. This approach reduces the effect due to round-off error (Skordos, 1993; Mei et al., 2006).

For the D2Q9 model above, the speed of sound is

$$c_s = \frac{1}{\sqrt{3}}c, \quad c := \delta_x/\delta_t, \tag{4}$$

and the shear viscosity ν and the bulk viscosity ζ are:

$$\nu = \frac{1}{3}\left(\frac{1}{s_7} - \frac{1}{2}\right)c\delta_x, \quad \zeta = \frac{1}{6}\left(\frac{1}{s_1} - \frac{1}{2}\right)c\delta_x. \tag{5}$$

For the D2Q9 model, it is required that $s_7 = s_8$ and $s_4 = s_6$. Obviously, the relaxation rates s_0, s_3 and s_5 for the conserved moments ($\delta\rho$, j_x and j_y) have no effect for the model. The other relaxation rates, s_2 (for ε) and $s_4 = s_6$ (for q_x and q_y) do not affect the hydrodynamics in the lowest order approximation and only affect the small scale behaviour of the model. Also, the relaxation rates $s_4 = s_6$ can affect the accuracy of the boundary conditions (Ginzbourg and Adler, 1994; Ginzburg and d'Humières, 1996, 2003; Pan et al., 2006). Usually, the values of s_2 and $s_4 = s_6$ are determined by the linear stability analysis. We will use $s_2 = 1.54$ and $s_4 = s_6 = 1.9$ unless otherwise stated.

2.1.1 *Interpolated bounce-back boundary conditions*

The Bounce-Back (BB) Boundary Conditions (BCs) are used to model the no-slip fluid-solid boundary conditions and are based on the intuitive picture that a particle reverses its momentum when colliding with a solid wall. Based on this intuitive picture, as illustrated by Figure 1(a), if the wall is located one half grid spacing beyond the last fluid node, $\boldsymbol{r}_\mathrm{A}$, then a particle with velocity $\boldsymbol{c}_1$ at node $\boldsymbol{r}_\mathrm{A}$ and time t_n collides with the wall at $\boldsymbol{r}_\mathrm{w}$, reverses its momentum, and returns to $\boldsymbol{r}_\mathrm{A}$. Therefore, when the wall location is not precisely located at $\delta_x/2$ beyond the last fluid node, interpolations can be used to compute the distribution function at a desirable location. Let

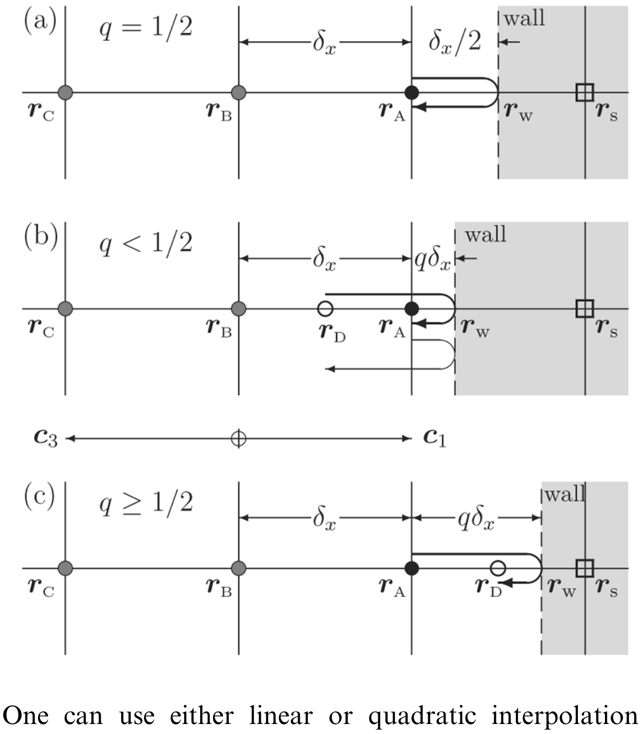
$$q := \frac{\|\boldsymbol{r}_\mathrm{A} - \boldsymbol{r}_\mathrm{w}\|}{\delta_x}, \tag{6}$$

as depicted in Figure 1, then

- when $q < \frac{1}{2}$, $f_i(\boldsymbol{r}_\mathrm{D})$ can be constructed from the f_i's in the nearby nodes *before* the bounce-back collision, so that $f_{\bar{\imath}}(\boldsymbol{r}_\mathrm{A})$ is obtained after the bounce-back collision with the wall located at $\boldsymbol{r}_\mathrm{w}$
- when $q \geq \frac{1}{2}$, $f_{\bar{\imath}}(\boldsymbol{r}_\mathrm{A})$ is obtained with $f_{\bar{\imath}}(\boldsymbol{r}_\mathrm{D})$ *after* the bounce-back collision and $f_{\bar{\imath}}$ at other nearby nodes.

where $\boldsymbol{c}_{\bar{\imath}} := -\boldsymbol{c}_i$.

Figure 1 Illustration of the bounce-back (BB) boundary conditions (BCs): (a) $q = 1/2$, the 'perfect' BB-BCs without interpolation; (b) $q < 1/2$, the BB-BCs with interpolations *before* the collision with the wall located at $\boldsymbol{r}_\mathrm{w}$ and (c) $q \geq 1/2$, the BB-BCs with interpolations *after* the collision with the wall


(a)
$q = 1/2$
δ_x
$\delta_x/2$
wall
$\boldsymbol{r}_\mathrm{C}$
$\boldsymbol{r}_\mathrm{B}$
$\boldsymbol{r}_\mathrm{A}$
$\boldsymbol{r}_\mathrm{w}$
$\boldsymbol{r}_\mathrm{s}$
(b)
$q < 1/2$
δ_x
$q\delta_x$
wall
$\boldsymbol{r}_\mathrm{C}$
$\boldsymbol{r}_\mathrm{B}$
$\boldsymbol{r}_\mathrm{D}$
$\boldsymbol{r}_\mathrm{A}$
$\boldsymbol{r}_\mathrm{w}$
$\boldsymbol{r}_\mathrm{s}$
$\boldsymbol{c}_3$
$\boldsymbol{c}_1$
(c)
$q \geq 1/2$
δ_x
$q\delta_x$
wall
$\boldsymbol{r}_\mathrm{C}$
$\boldsymbol{r}_\mathrm{B}$
$\boldsymbol{r}_\mathrm{A}$
$\boldsymbol{r}_\mathrm{D}$
$\boldsymbol{r}_\mathrm{w}$
$\boldsymbol{r}_\mathrm{s}$

One can use either linear or quadratic interpolations to compute the unknown distribution function $f_{\bar{\imath}}(\boldsymbol{r}_\mathrm{A})$ (Bouzidi et al., 2001). We will use the following quadratic interpolations:

$$\begin{aligned} f_{\bar{\imath}}(\boldsymbol{r}_\mathrm{A}, t_n + \delta_t) &= q(1+2q)f_i^*(\boldsymbol{r}_\mathrm{A}, t_n) \\ &\quad + (1-4q^2)f_i^*(\boldsymbol{r}_\mathrm{A} - \boldsymbol{c}_i\delta_t, t_n) \\ &\quad + q(2q-1)f_i^*(\boldsymbol{r}_\mathrm{A} - 2\boldsymbol{c}_i\delta_t, t_n), \quad 0 < q < 1/2, \end{aligned} \tag{7a}$$

$$
\begin{aligned}
f_{\bar{\imath}}(\boldsymbol{r}_{\mathrm{A}}, t_n + \delta_t) &= \frac{1}{q(1+2q)} f_i^*(\boldsymbol{r}_{\mathrm{A}}, t_n) \\
&+ \frac{(2q-1)}{q} f_{\bar{\imath}}^*(\boldsymbol{r}_{\mathrm{A}}, t_n) \\
&+ \frac{(1-2q)}{(1+2q)} f_{\bar{\imath}}^*(\boldsymbol{r}_{\mathrm{A}} - \boldsymbol{c}_i \delta_t, t_n), \quad 1/2 \le q < 1, \qquad \text{(7b)}
\end{aligned}
$$

where f_i^* denotes the post-collision distribution function. It must be stressed that $q = 0$ and $q = 1$ are singular cases which reduce to the case of $q = 1/2$ based on the basic picture of the bounce-back collision. Therefore it is simply futile to enforce the Dirichlet boundary conditions for velocity $\boldsymbol{u}$ to be satisfied on a grid point in the LBE (Ginzbourg and Adler, 1994; Ginzburg and d'Humières, 2003; Ginzburg, 2005a, 2005b, 2006, 2007).

While the bounce-back boundary conditions are the most often used in the LBE method, they are also misunderstood most often. By no means the intuitive picture illustrated in Figure 1 should not be taken for granted or literally. The precise location where the no-slip boundary conditions are satisfied is model dependent. It can be shown analytically that, for the Poiseuille flow, the no-slip boundary location is precisely one half lattice spacing beyond the last fluid node if and only if the following relation is satisfied (Ginzbourg and Adler, 1994; Ginzburg and d'Humières, 2003):

$$
s_q = 8\frac{(2 - s_\nu)}{(8 - s_\nu)}, \qquad (8)
$$

where $s_\nu = s_7 = s_8$ is the relaxation rate for p_{xx} and p_{xy} which also determines the shear viscosity ν given by Equation (5) and $s_q = s_4 = s_6$ is the relaxation rate for $\boldsymbol{q} = (q_x, q_y)$. Obviously, such a relationship is impossible to be satisfied with the single-relaxation-time model (or the BGK model), therefore the boundary location depends on the relaxation parameter τ in lattice BGK models.

2.2 Immersed Boundary Method

The Immersed Boundary Method (IBM) has become a popular method to treat boundary conditions in computational fluid dynamics. In IBM, various boundary conditions are re-casted as body forces due to boundary condition constraints. The immersed boundary method uses a set of discrete marker points (markers) to present boundary geometry. These markers are completely independent of the mesh used in a flow solver and are governed by Lagrangian dynamics. The coupling between the marker points and the flow are realised by the force the flow exerts on the markers and the reaction force the markers exert on the flow. The distribution of the forces is accomplished by an approximated Dirac δ-function.

For the two-dimensional domain Ω containing a one-dimensional closed boundary Γ, the configuration of Γ can be represented in the parametric form of $\boldsymbol{X}(\varsigma, t)$ for $0 \le \varsigma \le 1$, and $\boldsymbol{X}(0, t) = \boldsymbol{X}(N_\varsigma, t)$, where the parameter ς tracks a material point of the boundary. The equation of motion for a Lagrangian point on the boundary is:

$$
\begin{aligned}
\partial_t \boldsymbol{X}(\varsigma, t) &= \boldsymbol{u}(\boldsymbol{X}(\varsigma, t)) \\
&= \int_\Omega \boldsymbol{u}(\boldsymbol{x}, t)\delta(\boldsymbol{x} - \boldsymbol{X}(\varsigma, t))\, \mathrm{d}\boldsymbol{x},
\end{aligned} \qquad (9)
$$

where $\boldsymbol{u}$ is the velocity of the marker at position $\boldsymbol{X}(\varsigma, t)$. By virtue of the no-slip boundary conditions, the velocity $\boldsymbol{u}$ must also satisfy the incompressible Navier-Stokes equations:

$$
\rho\partial_t \boldsymbol{u} + \rho\boldsymbol{u} \cdot \boldsymbol{\nabla}\boldsymbol{u} = -\boldsymbol{\nabla}p + \rho\nu\nabla^2\boldsymbol{u} + \boldsymbol{f}, \qquad (10)
$$

where the flow velocity $\boldsymbol{u}$, the pressure p, and the body force $\boldsymbol{f}$ evolve on the Eulerian coordinate system $\boldsymbol{x}$, while the boundary force $\boldsymbol{F}$ evolves on the Lagrangian coordinate system $\boldsymbol{X}$ defined by the boundary Γ. The body force $\boldsymbol{f}$ and the boundary force $\boldsymbol{F}$ are related as the following:

$$
\boldsymbol{f}(\boldsymbol{x}, t) = \int_\Gamma \boldsymbol{F}(\varsigma, t)\delta(\boldsymbol{x} - \boldsymbol{X}(\varsigma, t))\, \mathrm{d}\varsigma, \qquad (11a)
$$

$$
\boldsymbol{F}(\varsigma, t) = \boldsymbol{F}(\boldsymbol{X}(\varsigma, t), t). \qquad (11b)
$$

The essence of the IBM is represented by Equations (9), (11a) and (11b). The Lagrangian markers are evolved by the velocity $\boldsymbol{u}$ and the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ must be interpolated from the Eulerian coordinate $\boldsymbol{x}$ in the flow domain Ω to the Lagrangian coordinate $\boldsymbol{X}(\varsigma, t)$ at the boundary Γ, as indicated by the integral over the flow domain Ω in Equation (9). The boundary force $\boldsymbol{F}$ has to be distributed from the Lagrangian coordinate $\boldsymbol{X}$ to the Eulerian one $\boldsymbol{x}$ as indicated by the integral over the boundary Γ in Equation (11a). Obviously, the connection between the Eulerian mesh for flows and the Lagrangian markers for boundaries relies on the approximation of the Dirac δ-function. We use the following smooth approximation of the Dirac δ-function in d dimensions:

$$
\delta_h(\boldsymbol{x}) = \delta_h(x_1)\delta_h(x_2)\cdots\delta_h(x_d), \quad \boldsymbol{x} \in \mathbb{R}^d, \qquad (12a)
$$

$$
\delta_h(x) = \begin{cases} \dfrac{1}{4h}\left[1 + \cos\left(\dfrac{\pi x}{2h}\right)\right] & |x| \le 2h, \\ 0 & |x| > 2h. \end{cases} \qquad (12b)
$$

Obviously, the approximated δ-function satisfies the following criteria:

$$
\lim_{h\to 0} \delta_h(x) = \delta(x), \quad \sum_{x_i} \delta_h(x_i)h = 1,
$$

and other higher order moment constraints (Peskin, 2002).

2.2.1 Fluid-boundary interaction in IBM

The hydrodynamic force due to flow at any point $\boldsymbol{x}$ is

$$
\boldsymbol{f}'(\boldsymbol{x}) = \rho(\partial_t\boldsymbol{u} + \boldsymbol{u} \cdot \boldsymbol{\nabla}\boldsymbol{u}) + \boldsymbol{\nabla}p - \rho\nu\nabla^2\boldsymbol{u}, \qquad (13)
$$

which can be approximated in discrete time as the following:

$$
\begin{aligned}
\boldsymbol{f}'(\boldsymbol{x}) &= \rho^{(n)}\left(\frac{\boldsymbol{u}^{(n+1)} - \boldsymbol{u}^{(n)}}{\delta_t} + \boldsymbol{u}^{(n)} \cdot \boldsymbol{\nabla}\boldsymbol{u}^{(n)}\right) \\
&\quad + \boldsymbol{\nabla}p^{(n)} - \rho^{(n)}\nu\nabla^2\boldsymbol{u}^{(n)},
\end{aligned} \qquad (14)
$$

where the superscript (n) denotes the discrete time $t = t_n = n\delta_t$. If the velocity $\boldsymbol{u} = \boldsymbol{U}$ at the boundary $\boldsymbol{x} = \boldsymbol{X}$, then the no-slip boundary conditions lead to:

$$\begin{aligned}\boldsymbol{f}'(\boldsymbol{X}) = \rho^{(n)}\left(\frac{\boldsymbol{U}-\boldsymbol{u}^{(n)}}{\delta_t} + \boldsymbol{u}^{(n)}\cdot\boldsymbol{\nabla}\boldsymbol{u}^{(n)}\right)\\ + \boldsymbol{\nabla}p^{(n)} - \rho^{(n)}\nu\nabla^2\boldsymbol{u}^{(n)}.\end{aligned} \tag{15}$$

Because the flow fields $\boldsymbol{u}$ and p are computed on the Eulerian grids $\{\boldsymbol{x}_j\}$, so is the force $\boldsymbol{f}'$. To obtain the value of $\boldsymbol{f}'$ at a given boundary point $\boldsymbol{X}$, we use the following interpolation formula for $\boldsymbol{u}$ and p:

$$\begin{aligned}\boldsymbol{u}(\boldsymbol{X}) &= \int_\Omega \boldsymbol{u}(\boldsymbol{x})\delta(\boldsymbol{x}-\boldsymbol{X})\mathrm{d}\boldsymbol{x}\\ &\approx \int_\Omega \boldsymbol{u}(\boldsymbol{x})\delta_h(\boldsymbol{x}-\boldsymbol{X})\mathrm{d}\boldsymbol{x}\\ &\approx \sum_{\boldsymbol{x}_j\in\Omega_h(\boldsymbol{X})} \boldsymbol{u}(\boldsymbol{x}_j)\delta_h(\boldsymbol{x}_j-\boldsymbol{X})h^d,\end{aligned} \tag{16}$$

where $\Omega_h(\boldsymbol{X})$ denotes the support of $\delta_h(\boldsymbol{x}-\boldsymbol{X})$.

The boundary Γ is represented by L discrete points $\{\boldsymbol{X}_k := \boldsymbol{X}(\varsigma_k)|k = 1, 2, \ldots, L\}$. The boundary force $\boldsymbol{F}$ is evaluated over a linear segment ℓ_k of the boundary:

$$\begin{aligned}\int_{\Gamma_k}\boldsymbol{F}(\boldsymbol{X})\mathrm{d}\boldsymbol{X} &= \int_{\Gamma_k}\mathrm{d}\boldsymbol{X}\int_\Gamma \mathrm{d}\boldsymbol{X}'\,\boldsymbol{F}(\boldsymbol{X}')\delta(\boldsymbol{X}'-\boldsymbol{X})\\ &= \int_{\Gamma_k}\boldsymbol{f}'(\boldsymbol{X})\,\mathrm{d}\boldsymbol{X} = \int_{\Omega(\Gamma_k)}\boldsymbol{f}'(\boldsymbol{x})\,\delta(\boldsymbol{x}-\boldsymbol{X})\,\mathrm{d}\boldsymbol{x},\end{aligned} \tag{17}$$

where Γ_k denotes the linear segment between $\boldsymbol{X}_{k-1/2}$ and $\boldsymbol{X}_{k+1/2}$, $\boldsymbol{X}_{k\pm1/2} := (\boldsymbol{X}_k + \boldsymbol{X}_{k\pm1})/2$, and $\Omega(\Gamma_k)$ is a covering of Γ_k such that $\Gamma_k = \Gamma\cap\Omega(\Gamma_k)$. The first integral of $\boldsymbol{F}(\boldsymbol{X})$ in Equation (17) is approximated by

$$\int_{\Gamma_k}\boldsymbol{F}(\boldsymbol{X})\mathrm{d}\boldsymbol{X} \approx \boldsymbol{F}(\boldsymbol{X}_k)\ell_k,$$

where $\ell_k := \|\boldsymbol{X}_{k+1/2} - \boldsymbol{X}_{k-1/2}\|$, and the last integral of $\boldsymbol{f}(\boldsymbol{x})$ is approximated by

$$\int_{\Omega(\Gamma_k)}\boldsymbol{f}'(\boldsymbol{x})\,\delta(\boldsymbol{x}-\boldsymbol{X})\,\mathrm{d}\boldsymbol{x} \approx \boldsymbol{f}'(\boldsymbol{x}=\boldsymbol{X}_k)V_k,$$

where $V_k := V(\Omega(\Gamma_k))$ is the volume (or area) of $\Omega(\Gamma_k)$. The choice of $\Omega(\Gamma_k)$ can be somewhat arbitrary, and we use what proposed in Feng and Michaelides (2004):

$$V_k = \ell_k\delta_x,$$

as illustrated in Figure 2. Therefore we have:

$$\boldsymbol{F}(\boldsymbol{X}_k)\ell_k \approx \boldsymbol{f}'(\boldsymbol{X}_k)\ell_k\delta_x. \tag{18}$$

For IBM, although it is not necessary for the test case studied in this paper, we ensure that $\ell_k \le \delta_x/2$ in our simulations.

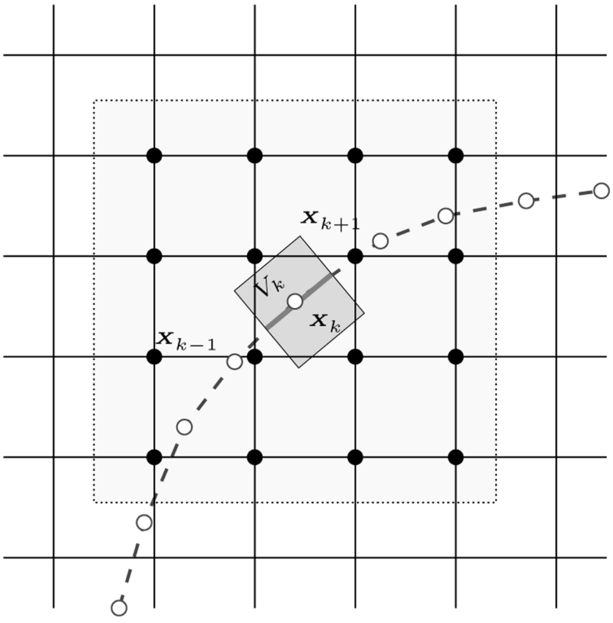
With the relationship between the surface force $\boldsymbol{F}$ and the body force $\boldsymbol{f}$ given by Equation (18), the fluid-structure interaction between the flow evolved on the Eulerian mesh $\{\boldsymbol{x}_j\}$ and the interface on the Lagrangian grid $\{\boldsymbol{X}_k\}$ is carried out as follows:

- At each time step $t = t_n$, flow fields $\boldsymbol{u}^{(n)}$ and $p^{(n)}$ are interpolated to a Lagrangian grid $\boldsymbol{X}_k$ from nearby Eulerian grid points $\boldsymbol{x}_j \in \Omega_h(\boldsymbol{X}_k)$ according to Equation (16), the hydrodynamic force is given by $\boldsymbol{f}'(\boldsymbol{X}_k)$ of Equation (14).
- The surface force $\boldsymbol{F}(\boldsymbol{X}_k)$ is computed from the hydrodynamic body force $\boldsymbol{f}'(\boldsymbol{X}_k)$ according to Equation (18). If applicable, the integration of the surface force $\boldsymbol{F}(\boldsymbol{X}_k)$ is used to evolve the body closed by the interface $\Gamma = \{\boldsymbol{X}_k\}$.
- According to Equation (11a), the surface force $\boldsymbol{F}(\boldsymbol{X}_k)$ is re-distributed to the Eulerian grids as the body force in the Navier-Stokes Equation (10):

$$\boldsymbol{f}(\boldsymbol{x}_j) = \sum_{\boldsymbol{X}_k\in\Omega_h(\boldsymbol{x}_j)}\boldsymbol{F}(\boldsymbol{X}_k)\delta_h(\boldsymbol{x}_j-\boldsymbol{X}_k)h^{(d-1)}. \tag{19}$$

- With the body force $\boldsymbol{f}$ due to fluid-boundary interaction given, the flow variables $\boldsymbol{u}$ and p can be evolved to the next time step $t = t_{n+1}$.

Figure 2 Calculation of local forces at the boundary. The horizontal and vertical lines are the Eulerian mesh. The solid discs denote the grid points within Ω_k with respect to $\boldsymbol{X}_k$. The dashed curve is the boundary Γ and the circles denote the Lagrangian points $\{\boldsymbol{X}_k\}$. The thick line denotes the linear segment $\ell_k = \|\boldsymbol{X}_{k+1/2} - \boldsymbol{X}_{k-1/2}\|$, and the shaded rectangle denotes the area $V_k = \ell_k\delta_x$



2.2.2 The body force in the LBE

To ensure the mass conservation up to the second-order Chapman-Enskog expansion, the procedure to include the body force $\boldsymbol{f}$ in the LBE is the following.

- compute moments $\mathbf{m} = \mathsf{M}\cdot\mathbf{f}$ at $t = t_n$

- compute $\boldsymbol{j}' = \boldsymbol{j} + \boldsymbol{f}\delta_t/2$, then $\boldsymbol{j}'$ is used to compute the hydrodynamic force at boundary $\boldsymbol{f}'(\boldsymbol{X}_k)$ and as the output of the flow momentum
- use $\boldsymbol{j}'$ to compute the equilibria $\mathbf{m}^{(\text{eq})}$ of Equations (3), then the collision (or relaxation) step is carried out for the non-conserved moments using $\mathbf{m}^{(\text{eq})}(\rho, \boldsymbol{j}')$
- compute $\boldsymbol{j}'' = \boldsymbol{j}' + \boldsymbol{f}\delta_t/2$
- compute the post-collision distribution $\mathbf{f}^*$ with $\boldsymbol{j}''$ as the momentum
- advect $\mathbf{f}^*(t_n)$ to $\mathbf{f}(t_{n+1})$.

2.3 LBE with non-uniformly stretched grids

The lattice Boltzmann equation employs a uniform Cartesian mesh with δ_x. To use non-uniform mesh, one can use either local grid refinement (Filippova and Hänel, 1998; Crouse et al., 2003; Tölke et al., 2006) or grid stretching (He et al., 1996, 1997; He and Luo, 1997) and the latter approach is used in this study. In the grid-stretching approach, one can use an arbitrary non-uniform mesh with the finest grid spacing of δ_x. On an arbitrary non-uniform mesh, the advection step in the LBE does not transport data from one grid point to another, thus interpolations are required where the grid spacings are larger than δ_x.

In this work we will use a very simple grid stretching strategy for a two dimensional Cartesian mesh. About a rectangular uniform fine mesh of δ_x, the grid spacings are stretched exponentially along both x and y directions beyond the four boundaries of the fine mesh, as illustrated by Figure 3. The stretched grids are given by:

$$x_k = A_x e^{\pi \xi_k}, \quad \xi_k = \frac{(k-1)}{(N_\xi - 1)}\xi_\infty = (k-1)\delta_\xi, \tag{20a}$$

$$y_k = A_y e^{\pi \eta_k}, \quad \eta_k = \frac{(k-1)}{(N_\eta - 1)}\eta_\infty = (k-1)\delta_\eta, \tag{20b}$$

where index k is only used for the stretched grids, A_x and A_y are parameters which determine the first stretched grid spacing in x- and y-axis, respectively, N_ξ and N_η are the numbers of non-uniform grids in ξ and η coordinates, respectively, and $x_\infty := e^{\pi\xi_\infty}$ and $y_\infty := e^{\pi\eta_\infty}$ are the boundaries in the corresponding directions. We define:

$$\mathrm{d}_i\xi_j = \xi(\boldsymbol{x}_j - \boldsymbol{c}_i\delta t) - \xi(\boldsymbol{x}_j), \tag{21a}$$

$$\mathrm{d}_i\eta_j = \eta(\boldsymbol{x}_j - \boldsymbol{c}_i\delta t) - \eta(\boldsymbol{x}_j). \tag{21b}$$

The values of the parameters pertinent to grid stretching will be given later.

On the non-uniform mesh, the distribution functions can be computed by using second-order upwind interpolations:

$$f_i(j, k, t_{n+1}) = \sum_{j',\,k'=0}^{2} a^i_{j,j'} b^i_{k,k'} f^*_i(j - i_x j', k - i_y k', t_n), \tag{22}$$

where $i_x = \text{sign}(c_{ix})$ and $i_y = \text{sign}(c_{iy})$ are the signs of discrete velocity components c_{ix} and c_{iy}, respectively, $\boldsymbol{c}_i = (c_{ix}, c_{iy})$. The interpolations are upwind with respect to the discrete velocity $\boldsymbol{c}_i$. The interpolation coefficients are

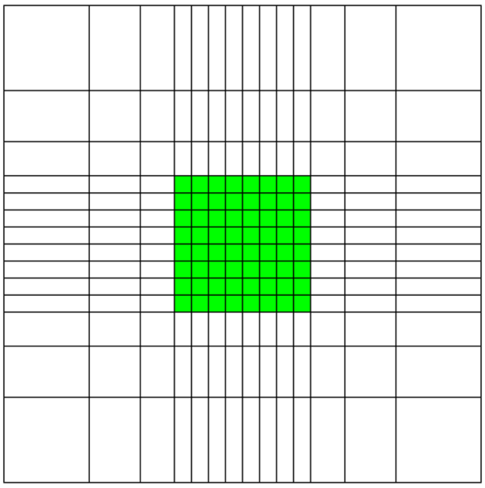
$$a^i_{j,0} = \frac{(|\mathrm{d}_i\xi_j| - \delta_\xi)(|\mathrm{d}_i\xi_j| - 2\delta_\xi)}{2{\delta_\xi}^2}, \tag{23a}$$

$$a^i_{j,1} = -\frac{|\mathrm{d}_i\xi_j|(|\mathrm{d}_i\xi_j| - 2\delta_\xi)}{{\delta_\xi}^2}, \tag{23b}$$

$$a^i_{j,2} = \frac{|\mathrm{d}_i\xi_j|(|\mathrm{d}_i\xi_j| - \delta_\xi)}{2{\delta_\xi}^2}, \tag{23c}$$

where $\delta_\xi := \xi_\infty/(N_\xi - 1)$ and $\delta_\eta := \eta_\infty/(N_\eta - 1)$ are the grid spacings in the new coordinate system $(\xi,\, \eta)$, $b^i_{k,k'}$ can be defined similarly by substituting $|\mathrm{d}_i\eta_k|$ and δ_η for $|\mathrm{d}_i\xi_j|$ and δ_ξ, respectively, in $a^i_{j,j'}$.

Figure 3 Illustration of stretched mesh. The shaded part is the fine mesh of the grid space $\delta_x = 1$



It should be noted that interpolations used in grid stretching increase numerical dissipation and anisotropic effects in small scales close to the grid spacing δ_x (Lallemand and Luo, 2003). However, the effects due to the second-order interpolations are beyond k^2, where k is the wave-number. Given the fact that the LBE method is second-order accurate in space, the interpolations would not affect the formal numerical accuracy of the LBE method and the results in simulations, as observed previously (He et al., 1996, 1997; He and Luo, 1997).

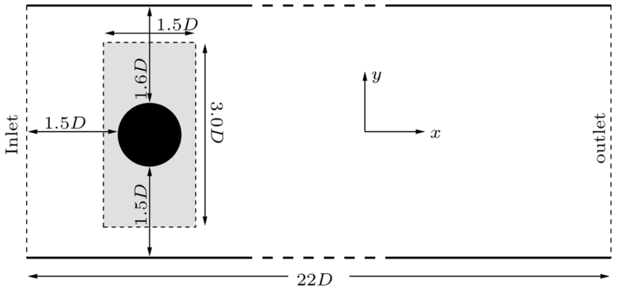
3 Results and discussion

In this study we intend to compare the immersed boundary method with the interpolated bounce-back boundary conditions in the lattice Boltzmann equation, so the numerical accuracy and computational efficiency of these two methods can be assessed. We will compare these two methods by using the test case of the flow past a cylinder asymmetrically placed in a channel in two dimensions.

3.1 Flow configuration, boundary conditions, and numerical mesh

Figure 4 illustrates the geometric configuration of the flow past a cylinder asymmetrically placed in a channel in two dimensions. The cylinder diameter is D. The computational domain is $L \times H = 22D \times 4.1D$, as depicted in Figure 4. A rectangle of $1.5D \times 3.0D$ co-centric with the cylinder is covered with fine grid of $\delta_x = 1$. The grid spacings beyond the rectangular area are stretched. Table 1 provides the values of the grid spacings in the unit of δ_x with minimum and maximum stretching in right ($+x$), left ($-x$), upper ($+y$) and lower ($-y$) part of mesh with respect to the cylinder centre for all the meshes used in our calculations. In the upper stream and spanwise directions, the grids are not stretched severely. However, grids near the outlet are stretched with a factor larger than 20.

Figure 4 The geometric configuration for the flow past a cylinder asymmetrically placed in the channel



At the channel walls and the cylinder surface, the no-slip boundary conditions are applied. At the channel walls, the bounce-back boundary conditions are used, while at the cylinder surface, either immersed-boundary method or the interpolated bounce-back method is used to realise the no-slip boundary conditions.

In the inlet at the left, the following velocity conditions are imposed:

$$u(0, y) = 4U_{\max} y(H - y)/H^2, \quad 0 \le y \le H, \tag{24a}$$

$$v = 0, \tag{24b}$$

where $U_{\max}$ is the maximum velocity at the center of the channel inlet. The inflow boundary conditions are realised by the bounce-back boundary conditions:

$$f_{\bar{\imath}} = f_i + 6w_i \boldsymbol{c}_{\bar{\imath}} \cdot \boldsymbol{u}(0, y) \tag{25}$$

where $w_0 = 4/9$, $w_i = 1/9$ for $i = 1$, 2, 3 and 4, and $w_i = 1/36$ for $i = 5$, 6, 7 and 8. At outlet, the following extrapolations are used (Chen et al., 1996)

$$f_i(N_x) = 2f_i(N_x - 1) - f_i(N_x - 2). \tag{26}$$

The cylinder diameter D and the maximum velocity at entrance $U_{\max}$ define the units in this problem. The unit of time is defined by $D/U_{\max}$. Therefore the grid spacing δ_x and the time step δ_t are given by $\delta_t = \delta_x = D/N_D$, with $D = 1$ and N_D is the number of grid points across the cylinder radius. The Reynolds number for this flow is defined by

$$\text{Re} = \frac{\overline{U}D}{\nu},$$

where $\overline{U} = 2U_{\max}/3$ is the mean velocity. We compute the drag and lift coefficients, C_D and C_L:

$$C_D = \frac{F_D}{\frac{1}{2}\rho_0 \overline{U}^2 D}, \quad C_L = \frac{F_L}{\frac{1}{2}\rho_0 \overline{U}^2 D}, \tag{27}$$

where F_D and F_L are the drag and lift forces, respectively. For the LBE with IBM, F_D and F_L are easily determined by:

$$F_D = -\int_\Omega f_x \mathrm{d}\boldsymbol{x} = -\oint_\Gamma F_x \mathrm{d}\boldsymbol{X}, \tag{28a}$$

$$F_L = -\int_\Omega f_y \mathrm{d}\boldsymbol{x} = -\oint_\Gamma F_y \mathrm{d}\boldsymbol{X}, \tag{28b}$$

where f_x and f_y (F_x and F_y) are the x- and y-components of the body force $\boldsymbol{f}$ (the surface $\boldsymbol{F}$), respectively, and Γ is the cylinder boundary. For unsteady flow, the Strouhal number is defined by

$$\text{St} = \frac{f_\omega D}{\overline{U}}, \tag{29}$$

where f_ω is the frequency in the lift force.

For the interpolated bounce-back method, the force exerted on the cylinder is computed from the momentum transfer at the boundary. Each boundary location $\boldsymbol{r}_j$ is

Table 1 The values of the grid spacings with minimum and maximum stretching in right ($+x$), left ($-x$), upper ($+y$) and lower ($-y$) part of mesh with respect to the cylinder for all the meshes used in our calculations. All the grid spacings are given in the unit of $\delta_x = 1$

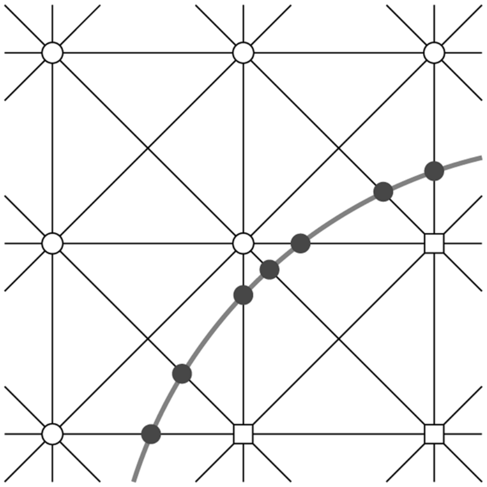
D/δ_x	10	20	25	40	50	80	100
$\Delta^{+x}_{\min}$	1.054	1.018	1.0325	1.0012	1.0	1.016	1.08
$\Delta^{+x}_{\max}$	23.289	25.424	24.525	25.84	24.455	26.712	28.35
$\Delta^{-x}_{\min}$	1.12	1.088	1.045	1.032	1.0	1.008	1.06
$\Delta^{-x}_{\max}$	2.454	2.706	2.482	2.66	2.446	2.6376	2.78
$\Delta^{+y}_{\max}$	1.044	1.0526	1.0	1.018	1.04	1.014	1.01
$\Delta^{+y}_{\max}$	1.367	1.3896	1.282	1.401	1.35	1.407	1.41
$\Delta^{-y}_{\min}$	1.119	1.098	1.005	1.024	1.057	1.019	1.03
$\Delta^{-y}_{\max}$	1.388	1.412	1.225	1.344	1.304	1.348	1.36

defined by the intersection between the cylinder boundary and a velocity vector connecting two grid points, as illustrated in Figure 5. At each intersection point $\boldsymbol{r}_j$, the hydrodynamic force exerted on the cylinder is

$$\boldsymbol{F} = \sum_{\boldsymbol{r}_j \in \Gamma} \boldsymbol{F}(\boldsymbol{r}_j), \quad \boldsymbol{F}(\boldsymbol{r}_j) := [f_i^*(t_n) + f_{\bar{\imath}}(t_n)]\boldsymbol{c}_i, \qquad (30)$$

where $\boldsymbol{c}_i$ connects a fluid node to a non-fluid node; f_i^* is the post-collision distribution of a particle heading to the collision with the boundary and $f_{\bar{\imath}}$ is the pre-collision distribution of a particle returned by the collision with the boundary. Either f_i^* or $f_{\bar{\imath}}$ has to be computed by using the interpolations of Equation (7), depending on the position of $\boldsymbol{r}_j$ with respect to fluid grids.

Figure 5 Illustration of the boundary points for the interpolated bounce-back boundary method in the LBE. The boundary points are marked by solid discs (●). The fluid and non-fluid nodes are marked by open circles (○) and squares (□), respectively. The curve represents a boundary



3.2 Steady flow at Re = 20

We first carry out the simulation for the steady flow at Re = 20. We use $U_{\max} = 0.03c$ ($\overline{U} = 0.02c$), and vary the values of both N_D and s_ν to maintain a constant Re according to: $1/s_\nu = 2\overline{U}N_D/\mathrm{Re} + 1/2$. We use the following criterion for the steady state:

$$\max_{\boldsymbol{x}_j} \frac{\|\boldsymbol{u}^{(n+1)}(\boldsymbol{x}_j) - \boldsymbol{u}^{(n)}(\boldsymbol{x}_j)\|}{\overline{U}} \leq 5.0 \times 10^{-6}. \qquad (31)$$

It is important to note that, even after the criterion of Equation (31) is satisfied, the C_D and C_L and other quantities are still fluctuating in time. Figure 6 shows the phase diagram of $C_L(t_n)$ vs. $C_D(t_n)$ for the case of $D = 40\delta_x$. The variations for both C_D and C_L are of 10^{-2} and 10^{-3}, respectively, after the convergence criterion (31) is attained. However, relative error in C_L is about 50% in this case, while that of C_D is less than 1%, because C_D is more than two orders of magnitude larger than C_L. Each cycle of C_L–C_D in Figure 6 takes about 300 iterations in time, and will take longer and longer as $t_n \to \infty$, because the LBE is an explicit scheme. After Equation (31) is satisfied, additional 2000 iterations are run to obtain the maximal and minimal values of C_D and C_L and then the mean values of C_D and C_L are computed. We also compute the difference of pressures at the front and back of the cylinder, Δp, and the recirculation length L_r in the same way we compute C_D and C_L. When the cylinder front and back are not grid points, we measure the pressures at the nearest grid points instead. These grid points are at most $\delta_x/2$ away from the true boundary locations, and because the pressure p is rather smooth at low Reynold number, we do not use extrapolations or other means to compute the pressure at the boundary of cylinder.

Figure 6 $C_L(t_n)$ vs. $C_D(t_n)$, Re = 20 and $D = 40\delta_x$. After the convergence criterion is met, 2000 iterations are plotted in this figure: (a) the interpolated bounce-back method and (b) the immersed-boundary method

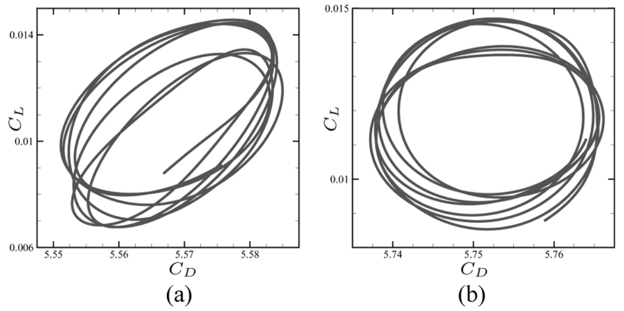


Table 2 compiles the results of the drag coefficient C_D, the lift coefficient C_L, the pressure difference Δp, and the recirculation length L_r for both the interpolated bounce-back method and IBM, with the resolutions of $D = 10\delta_x$, $20\delta_x$, $40\delta_x$, and $80\delta_x$. Table 2 also includes the number of grid points $N_x \times N_y$, and the number of iterations N_t and CPU time to attain the convergence criterion (31). Clearly, with the same resolution, the interpolated bounce-back method yields much more accurate results for C_D, C_L, Δp and L_r than the immersed-boundary method, while the computational time of these two methods are comparable. We also include the lower and upper bounds of C_D, C_L, Δp, and L_r given by Schäfer and Turek (1996). We use the averaged values of the lower and upper bounds and the values obtained with the largest mesh size to compute the relative errors, which are also included in Table 2. Clearly, the interpolated bounce-back method is much more accurate than the immersed-boundary method, the errors in the measurements obtained with the interpolated bounce-back method is almost one order of magnitude smaller than that obtained with the immersed-boundary method, except for C_L. In the case of C_L, the results are not well converged and it is only fortuitous that the IBM yields a better result.

To further investigate the difference between the immersed-boundary and interpolated bounce-back methods, we show in Figure 7 the normalised radial velocity $\tilde{u}_r := u_r/U_{\max}$ and normalised tangential velocity $\tilde{u}_\theta := u_\theta/U_{\max}$ on the cylinder surface. The angle θ

Table 2 Grid size dependence of C_D, C_L, Δp, L_r, N_t and the CPU time (s) for the steady-state flow at Re $= 20$. The relative errors are computed from the measurements obtained with the largest mesh size with respect to the corresponding mean values of lower and upper bounds given by Schäfer and Turek (1996)

D/δ_x	$N_x \times N_y$	C_D	C_L	Δp	L_r	N_t	*CPU* (s)
Interpolated bounce-back method							
10	50×40	5.6481	0.00848	0.1268	0.0815	6,400	26.0
20	95×79	5.5705	0.00920	0.1181	0.0833	12,500	152.0
25	123×101	5.5777	0.01036	0.1181	0.0845	18,400	394.3
40	190×158	5.5682	0.01062	0.1175	0.0849	23,400	1,135
50	248×199	5.5664	0.00992	0.1172	0.0850	29,200	2,777
80	375×315	5.5650	0.01046	0.1171	0.0851	36,000	7,027
Relative error		0.27%	2.2%	0.26%	0.57%		
Immersed-boundary method							
10	50×40	6.4376	0.02025	0.1458	0.0886	6,500	14.5
20	95×79	5.9339	0.01267	0.1323	0.0887	14,100	165.0
25	123×101	5.8572	0.01328	0.1287	0.0886	16,300	313.5
40	190×158	5.7512	0.01172	0.1243	0.0875	24,200	1,589
50	248×199	5.7163	0.01073	0.1228	0.0874	27,100	1,990
80	375×315	5.6688	0.01064	0.1206	0.0863	41,600	11,326
Relative error		1.6%	0.56%	2.7%	1.9%		
Lower and upper bounds given by Schäfer and Turek (1996)							
Lower bounds		5.5700	0.0104	0.1172	0.0842		
Upper bounds		5.5900	0.0110	0.1176	0.0852		

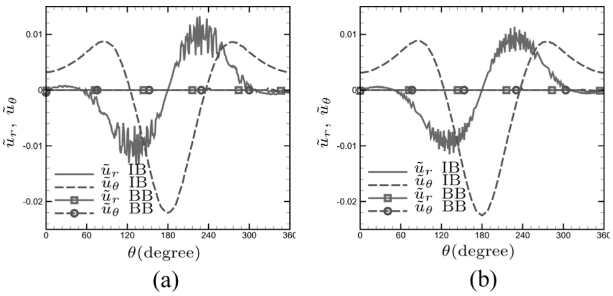
is measured from the back of the cylinder and varies counter-clockwise. The velocity on the cylinder boundary is obtained by using approximated δ-function of Equation (12) and the velocity values at grid points near the boundary. Clearly, the immersed-boundary method cannot satisfy the no-slip boundary conditions exactly on the cylinder surface and this has been observed previously (cf., e.g., Kim et al., 2001; Tyagi and Acharya, 2005). Furthermore, the magnitude of the velocity at the cylinder surface does not change as the grid spacing is refined, as shown in Figure 7 – the magnitude of the surface velocity remains almost the same with $D = 40\delta_x$ and $D = 80\delta_x$. In contrast, the magnitude of the surface velocity obtained by the interpolated bounce-back boundary conditions is about two orders of magnitude smaller. For $D = 40\delta_x$, the maximum values of $\tilde{u}_r$ and $\tilde{u}_\theta$ are 0.0134 and 0.0220, respectively, for the immersed-boundary method and 0.0003 and 0.0004 for the interpolated bounce-back method; and for $D = 80\delta_x$, the maximum values of $\tilde{u}_r$ and $\tilde{u}_\theta$ are 0.0115 and 0.0225, respectively, for the immersed-boundary method and 0.0003 and 0.0002 for the interpolated bounce-back method. Clearly, inaccurate drag and lift forces are a direct consequence of inaccurate no-slip boundary conditions in the immersed boundary method.

We next study the convergence behaviours of all the measured quantities by computing the L_1 error:

$$E_1(g) = |g(\delta_x) - g_*|, \qquad (32)$$

where $g(\delta_x)$ is a measured value depending on the grid size δ_x, and g_* is the reference value of the quantity g obtained with the largest mesh in our calculations. The results are shown in Figure 8. In most cases, both the interpolated bounce-back and the immersed-boundary methods exhibit a second-order rate of convergence. For the the interpolated bounce-back method, the error of the lift coefficient C_L does not behave smoothly and the reason is that the results of C_L have yet to converge and its relative error in large (about 50%), as discussed previously. For the immersed-boundary method, the recirculation length L_r does not change much as the mesh resolution is refined.

Figure 7 Surface velocity with Re $= 20$. The radial velocity $u_r/U_{\max}$ and tangential velocity $u_\theta/U_{\max}$ at the cylinder surface: (a) $D = 40\delta_x$ and (b) $D = 80\delta_x$



(a) (b)

It should be noted that the CPU time for the interpolated bounce-back method is almost one order of magnitude longer than an optimised code. Usually, one can combine collision and advection together in one single do-loop to minimise data access time. In the present study, the collision and advection are carried out separately and our code is not optimised. However, the ratio between the CPU times of two methods should not be affected by whether the codes are optimised or not, so long as the LBE part in both methods are the same.

3.3 Unsteady flow at Re $= 100$

The calculations for the unsteady flow at $\mathrm{Re} = 100$ are carried out with the same meshes for the steady flow of $\mathrm{Re} = 20$. The maximum velocity at the entrance is $U_{\max} = 0.15c$.

For each run, a number of iterations, N_t, is carried out so that the solution reaches a periodic state. As an example, we show in Figure 9 the C_L-C_D phase diagram for the case of $D = 40\delta_x$. Clearly, the solution has reached the periodic state at this stage, the values of $C_D^{\max}$ and $C_L^{\max}$ only vary in their fourth significant digits after each period. The values of $C_D^{\max}$ (corresponding to $C_L < 0$), $C_L^{\max}$, Δp and St are the mid-values of maxima and minima of these quantities within 2,000 iterations after N_t initial iterations.

Figure 8 Steady flow at $\mathrm{Re} = 20$, convergence of (a) C_D, (b) C_L; (c) Δp and (d) L_r. The dashed lines have a slop of -2

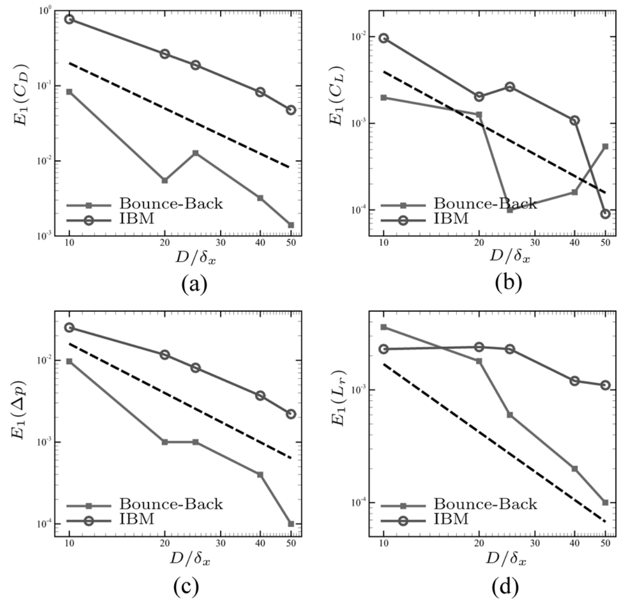


Figure 9 $C_L(t_n)$ vs. $C_D(t_n)$, $\mathrm{Re} = 100$ and $D = 40\delta_x$. After N_t iterations (cf. Table 3), 2000 iterations are plotted in this figure: (a) the interpolated bounce-back boundary conditions and (b) the immersed-boundary method

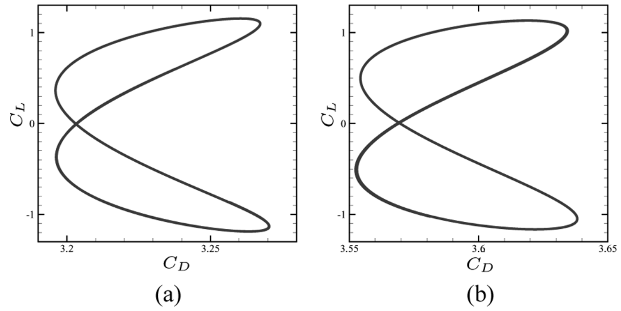
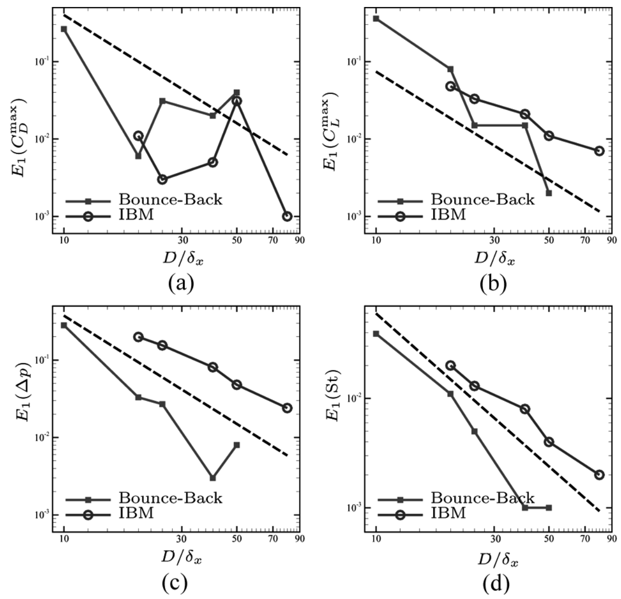


Table 3 compiles the results of $C_D^{\max}$, $C_L^{\max}$, Δp, St, N_t and the corresponding CPU time for the unsteady flow of $\mathrm{Re} = 100$. We note that, for the immersed-boundary method with the resolution of $D = 10\delta_x$, the solution does not have the vortex shedding, indicating that the mesh is too coarse for the immersed-boundary method in this case. In contrast, the interpolated bounce-back method with the resolution of $D = 10\delta_x$ produces reasonable results. The largest mesh used in the immersed-boundary calculation is $D = 100\delta_x$, finer than that of $D = 80\delta_x$ in the interpolated bounce-back calculation. We note that value of $C_L^{\max}$ is systematically larger than the value given by Schäfer and Turek (1996). Overall, the interpolated bounce-back method produces more accurate results than the immersed-boundary method.

Figure 10 shows the convergence behaviour of $C_D^{\max}$, $C_L^{\max}$, Δp, and St. Except the case of $C_D^{\max}$, other quantities, $C_L^{\max}$, Δp, and St, all exhibit a second-order rate of convergence. With the interpolated bounce-back method, the value of $C_D^{\max}$ is rather accurate even with a low resolution of $D = 20\delta_x$ – it has two significant figures accurately computed.

Figure 10 Order of accuracy in space for IBM and interpolated bounce back scheme at $\mathrm{Re} = 100$



4 Conclusion

In this work we present a preliminary comparison of the interpolated bounce-back and the immersed-boundary methods in the lattice Boltzmann equation in two dimensions. The two methods are validated through the simulations of the flow past a cylinder asymmetrically placed in the channel with the Reynolds number $\mathrm{Re} = 20$ and 100 (Schäfer and Turek, 1996). Our observations are summarised follows.

For the benchmark problem of the flow past a cylinder in channel (Schäfer and Turek, 1996), both the interpolated bounce-back and the immersed-boundary methods have

Table 3 Grid size dependence of $C_D^{\max}$, $C_L^{\max}$, Δp, St, N_t and the CPU time (s) for the unsteady flow at Re = 100

D/δ_x	$N_x \times N_y$	$C_D^{\max}$	$C_L^{\max}$	Δp	St	N_t	*CPU* (s)
Interpolated bounce-back method							
10	50 × 40	3.554	1.498	2.799	0.253	24,000	52.1
20	95 × 79	3.285	1.219	2.550	0.281	24,000	259.7
25	123 × 101	3.260	1.154	2.544	0.287	24,000	635.4
40	190 × 158	3.271	1.154	2.520	0.293	40,000	2,640
50	248 × 199	3.251	1.137	2.509	0.293	40,000	4,871
80	375 × 315	3.291	1.139	2.517	0.292	64,000	17,851
Error		1.9%	13.9%	1.5%	2.7%		
Immersed-boundary method							
20	95 × 79	3.658	1.155	2.823	0.278	24,000	280.6
25	123 × 101	3.644	1.140	2.779	0.285	24,000	644.8
40	190 × 158	3.642	1.128	2.705	0.290	40,000	1,975
50	248 × 199	3.616	1.118	2.672	0.294	40,000	3,541
80	375 × 315	3.648	1.114	2.648	0.296	64,000	13,469
100	451 × 393	3.647	1.107	2.624	0.298	80,000	24,181
Error		12.9%	10.7%	5.8%	0.67%		
Lower and upper bounds given by Schäfer and Turek (1996)							
Lower bounds		3.2200	0.9900	2.4600	0.2950		
Upper bounds		3.2400	1.0100	2.5000	0.3050		

comparable computational speed, i.e., with an identical mesh, the CPU times per iteration for the two method are almost the same. However, the immersed-boundary method is relatively easier to implement. Based on the results we have shown, the interpolated bounce-back method is much more accurate than the immersed-boundary method, although, interestingly, both methods exhibit a second-order rate of convergence. This is interesting because the immersed-boundary method is a first-order method. We note that the LBE coupled with front tracking method also exhibits a second-order rate of convergence (Lallemand et al., 2007). Since the two methods have comparable computational speed, the interpolated bounce-back method is more computationally efficient because it is much more accurate thus is less demanding of grid resolution. We note that the no-slip boundary conditions are not satisfied accurately in the immersed boundary method when compared to the interpolated bounce-back method. Consequently, the immersed boundary method yields less accurate results than the interpolated bounce-back method.

Because of its easy implementation, the immersed-boundary method has a great potential for modelling and simulations of moving boundary problems, such as particulate suspensions and deformable objects in fluids. Since there have been only few works on the subject of the IBM combined with the LBE, this subject deserves further investigation.

Acknowledgement

The support from the US National Science Foundation under Grant No. CBET-0500213 is gratefully acknowledged.

References

Bouzidi, M., Firdaouss, M. and Lallemand, P. (2001) 'Momentum transfer of a Boltzmann-lattice fluid with boundaries', *Phys. Fluids*, Vol. 13, No. 11, pp.3452–3459.

Chen, H., Chen, S. and Matthaeus, H.W. (1992) 'Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method', *Phys. Rev. A*, Vol. 45, No. 8, pp.R5339–R5342.

Chen, S., Martinez, D. and Mei, R. (1996) 'On boundary conditions in lattice Boltzmann method', *Phys. Fluids*, Vol. 8, No. 9, pp.2527–2536.

Crouse, B., Rank, E., Krafczyk, M. and Tölke, J. (2003) 'A LB based approach for adaptive flow simulations', *Int. J. Mod. Phys. C*, Vol. 17, Nos. 1–2, pp.109–112.

d'Humières, D. (1992) 'Generalized lattice-Boltzmann equations', in Shizgal, B.D. and Weave, D.P. (Eds.): *Rarefied Gas Dynamics: Theory and Simulations*, Vol. 159 of Prog. Astronaut. Aeronaut., AIAA, Washington DC, pp.450–458.

d'Humières, D., Ginzburg, I., Krafczyk, M., Lallemand, P. and Luo, L-S. (2002) 'Multiple-relaxation-time lattice Boltzmann models in three-dimensions', *Philos. Trans. R. Soc. Lond. A*, Vol. 360, No. 1792, pp.437–451.

Feng, Z. and Michaelides, E. (2004) 'The immersed boundarylattice Boltzmann method for solving fluid-particles interaction problems', *J. Comput. Phys.*, Vol. 195, No. 2, pp.602–628.

Feng, Z. and Michaelides, E. (2005) 'Proteus: a direct forcing method in the simulations of particulate flows', *J. Comput. Phys.*, Vol. 202, No. 1, pp.20–51.

Filippova, O. and Hänel, D. (1998) 'Grid refinement for lattice-BGK models', *J. Comput. Phys.*, Vol. 147, No. 1, pp.219–228.

Ginzburg, I. (2005a) 'Equilibrium-type and link-type lattice Boltzmann models for generic advection and anisotropic-dispersion equation', *Adv. Water Res.*, Vol. 28, No. 11, pp.1171–1195.

Ginzburg, I. (2005b) 'Generic boundary conditions for lattice Boltzmann models and their application to advection and anisotropic-dispersion equations', *Adv. Water Res.*, Vol. 28, No. 11, pp.1196–1216.

Ginzburg, I. (2006) 'Variably saturated flow with the anisotropic lattice Boltzmann methods', *Comput. Fluids*, Vol. 35, Nos. 8–9, pp.831–848.

Ginzburg, I. (2007) 'Lattice Boltzmann modeling with discontinuous collision components: hydrodynamic and advection-diffiusion equations', *J. Stat. Phys.*, Vol. 126, No. 1, pp.157–206.

Ginzbourg, I. and Adler, P. (1994) 'Boundary flow condition analysis for the three-dimensional lattice Boltzmann method', *J. Phys. II*, Vol. 4, No. 2, pp.191–214.

Ginzburg, I. and d'Humières, D. (1996) 'Local second-order boundary methods for lattice Boltzmann models', *J. Stat. Phys.*, Vol. 84, Nos. 5–6, pp.927–971.

Ginzburg, I. and d'Humières, D. (2003) 'Multireflection boundary conditions for lattice Boltzmann models', *Phys. Rev. E*, Vol. 68, No. 6, p.066614.

He, X. and Luo, L-S. (1997) 'A priori derivation of the lattice Boltzmann equation', *Phys. Rev. E*, Vol. 55, No. 6, pp.R6333–R6336.

He, X., Luo, L-S. and Dembo, M. (1996) 'Some progress in lattice Boltzmann method: Part I. Nonuniform mesh grids', *J. Comput. Phys.*, Vol. 129, No. 2, pp.357–363.

He, X., Luo, L-S. and Dembo, M. (1997) 'Some progress in lattice Boltzmann method: enhancement of Reynolds number in simulations', *Physica A*, Vol. 239, Nos. 1–3, pp.276–285.

Junk, M., Klar, A. and Luo, L-S. (2005) 'Asymptotic analysis of the lattice Boltzmann equation', *J. Comput. Phys.*, Vol. 210, No. 2, pp.676–704.

Kim, J., Kim, D. and Choi, H. (2001) 'An immersed-boundary finite-volume method for simulations of flow in complex geometries', *J. Comput. Phys.*, Vol. 171, No. 1, pp.132–150.

Ladd, A.J.C. (1994a) 'Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation', *J. Fluid Mech.*, Vol. 271, pp.285–309.

Ladd, A.J.C. (1994b) 'Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 2. Numerical results', *J. Fluid Mech.*, Vol. 271, pp.311–339.

Ladd, A.J.C. and Verberg, R. (2001) 'Lattice Boltzmann simulations of particle-fluid suspensions', *J. Stat. Phys.*, Vol. 104, Nos. 5–6, pp.1191–1251.

Lai, M.C. and Peskin, C.S. (2000) 'An immersed boundary method with formal second-order accuracy and reduced numerical viscosity', *J. Comput. Phys.*, Vol. 160, No. 2, pp.705–719.

Lallemand, P. and Luo, L-S. (2000) 'Theory of the lattice Boltzmann method: dispersion, isotropy, Galilean invariance and stability', *Phys. Rev. E*, Vol. 61, No. 3, pp.6546–6562.

Lallemand, P. and Luo, L-S. (2003) 'Theory of the lattice Boltzmann method: acoustic and thermal properties in two and three dimensions', *Phys. Rev. E*, Vol. 63, No. 3, p.036706.

Lallemand, P., Luo, L-S. and Peng, Y. (2007) 'A lattice Boltzmann front-tracking method for interface dynamics with surface tension in two-dimensions', *J. Comput. Phys.*, Vol. 226, No. 2, pp.1367–1384.

Li, Z. and Ito, K. (2006) *The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*, SIAM, Philadelphia.

Mei, R., Luo, L-S., Lallemand, P. and d'Humières, D. (2006) 'Consistent initial conditions for lattice Boltzmann simulations', *Comput. Fluids*, Vol. 35, Nos. 8–9, pp.855–862.

Mittal, R. and Iaccarino, G. (2005) 'Immersed boundary methods', *Annu. Rev. Fluid Mech.*, Vol. 37, pp.239–261.

Pan, C., Luo, L-S. and Miller, C.T. (2006) 'An evaluation of lattice Boltzmann schemes for porous medium flow simulation', *Comput. Fluids*, Vol. 35, No. 8, pp.898–909.

Peskin, C.S. (1977) 'Numerical analysis of blood flow in the heart', Phys. *Fluids*, Vol. 13, No. 3, pp.3452–3459.

Peskin, C.S. (2002) 'The immersed boundary method', *Acta Numer.*, Vol. 11, pp.479–517.

Qi, D. and Luo, L-S. (2002) 'Transitions in rotations of a nonspherical particle in a three-dimensional moderate Reynolds number Couette flow', *Phys. Fluids*, Vol. 14, No. 12, pp.4440–4443.

Qi, D. and Luo, L-S. (2003) 'Rotational and orientational behaviour of a three-dimensional spheroidal particles in Couette flow', *J. Fluid Mech.*, Vol. 477, pp.201–213.

Qi, D., Luo, L-S., Aravamuthan, R. and Strieder, W. (2002) 'Lateral migration and orientation of elliptical particles in poiseuille flows', *J. Stat. Phys.*, Vol. 107, Nos. 1–2, pp.102–120.

Qian, Y.H., d'Humières, D. and Lallemand, P. (1992) 'Lattice BGK models for Navier-Stokes equation', *Europhys. Lett.*, Vol. 17, No. 6, pp.479–484.

Schäfer, M. and Turek, S. (1996) 'Benchmark computations of laminar flow around a cylinder', in Hirschel, E.H. (Ed.): *Notes in Numerical Fluid Mechanics: Flow Simulations with High Performance Computers II*, Vieweg, Braunschweig, Germany, pp.547–566.

Skordos, P.A. (1993) 'Initial and boundary conditions for the lattice Boltzmann method', *Phys. Rev. E*, Vol. 48, No. 6, pp.4823–4842.

Tölke, J., Freudiger, S. and Krafczyk, M. (2006) 'An adaptive scheme using hierarchical grids for lattice Boltzmann multi-phase flow simulations', *Comput. Fluids*, Vol. 35, Nos. 8–9, pp.820–830.

Tyagi, M. and Acharya, S. (2005) 'Large eddy simulation of turbulent flows in complex and moving rigid geometries using the immersed boundary method', *Int. J. Numer. Meth. Fluids*, Vol. 48, No. 7, pp.691–722.

Yu, D., Mei, R., Luo, L-S. and Shyy, W. (2003) 'Viscous flow computations with the method of lattice Boltzmann equation', *Prog. Aerospace Sci.*, Vol. 39, No. 5, pp.329–367.