Journal of Computational Physics 228 (2009) 1963–1979

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications

J. Wu, C. Shu *

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

ARTICLE INFO

Article history:
Received 13 May 2008
Received in revised form 8 November 2008
Accepted 15 November 2008
Available online 28 November 2008

Keywords:
Immersed boundary method
Lattice Boltzmann method
Velocity correction
Incompressible flow
Numerical simulation
Non-slip boundary condition

ABSTRACT

A version of immersed boundary-lattice Boltzmann method (IB-LBM) is proposed in this work. It is based on the lattice Boltzmann equation with external forcing term proposed by Guo et al. [Z. Guo, C. Zheng, B. Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, Phys. Rev. E 65 (2002) 046308], which can well consider the effect of external force to the momentum and momentum flux as well as the discrete lattice effect. In this model, the velocity is contributed by two parts. One is from the density distribution function and can be termed as intermediate velocity, and the other is from the external force and can be considered as velocity correction. In the conventional IB-LBM, the force density (external force) is explicitly computed in advance. As a result, we cannot manipulate the velocity correction to enforce the non-slip boundary condition at the boundary point. In the present work, the velocity corrections (force density) at all boundary points are considered as unknowns which are computed in such a way that the non-slip boundary condition at the boundary points is enforced. The solution procedure of present IB-LBM is exactly the same as the conventional IB-LBM except that the non-slip boundary condition can be satisfied in the present model while it is only approximately satisfied in the conventional model. Numerical experiments for the flows around a circular cylinder and an airfoil show that there is no any penetration of streamlines to the solid body in the present results. This is not the case for the results obtained by the conventional IB-LBM. Another advantage of the present method is its simple calculation of force on the boundary. The force can be directly calculated from the relationship between the velocity correction and the force density.

1. Introduction

Currently, most of flow problems can be simulated by finite difference, finite volume and finite element methods. When a complex geometry is immersed in the flow domain, these approaches often involve the tedious grid generation and complicated solution process arising from strong coupling between the discretization of governing equations and implementation of boundary conditions. To simplify the solution process, it is desired to develop an approach which can decouple the solution of governing equations and the implementation of boundary conditions. The immersed boundary method (IBM) is such an approach. It was initially proposed by Peskin [1] in 1970s for simulation of blood flows in the heart. This method uses a fixed Eulerian mesh for the flow field, and a set of Lagrangian points to represent the boundary immersed in the fluid. The basic idea of IBM is to treat the physical boundary as deformable with high stiffness. A small distortion of the boundary will yield a force which tends to restore the boundary into its original shape. The balances of such forces are distributed into the

* Corresponding author. Tel.: +65 6516 6476; fax: +65 6779 1459.
E-mail address: mpeshuc@nus.edu.sg (C. Shu).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.11.019

Eulerian nodes and the Navier–Stokes (N–S) equations with a body force are solved on the whole domain including exterior and interior of the body.

Based on the work of Peskin [1], various improvements have been made recently [2–5]. Among the remarkable works, Lai and Peskin [2] proposed a so-called second-order accurate immersed boundary method which was used to simulate flows over a circular cylinder. The interaction between the fluid and the immersed boundary is modeled using a well-chosen Dirac delta function. As compared to their first-order model [2], numerical viscosity is reduced in this model. However, due to the use of the first-order Dirac delta function interpolation, the model does not truly have the second-order accuracy. By employing the feedback forcing to represent the solid body, Goldstein et al. [3] developed a model called the virtual boundary method which was used to simulate laminar and turbulent flows. Different from the way in Peskin [1] where Hook's law is used to calculate the restoring force at the boundary, feedback forcing method directly computes the restoring force by using fluid and boundary velocities. However, there are two user-specified parameters in this model, which may lead to inconvenience for its application. Ye et al. [4] proposed a method named the Cartesian grid method which combines the cell-merging technique with the finite volume method to simulate two-dimensional unsteady incompressible viscous flows. Due to irregular shapes of cells cut by the boundary, complex interpolation is required to calculate the fluxes, which may affect the computational efficiency. Lima E Silva et al. [5] proposed a version named physical virtual model to simulate an internal channel flow and the flow around a circular cylinder. This model is very similar to the work of Peskin [1] except that the restoring force is calculated by applying the momentum equations at the boundary points. The process involves tedious derivative approximation and interpolation of velocity and pressure.

In the above IBM versions, the solution of flow field is obtained by solving incompressible Navier–Stokes (N–S) equations. As an alternative computational technique to the N–S solvers, the lattice Boltzmann method (LBM) [6] has been proven to be an efficient approach for simulation of flow field. LBM is a particle-based numerical technique, which studies the dynamics of fictitious particles. The major advantage of LBM is its simplicity, easy implementation, algebraic operation and intrinsic parallel nature. Like the IBM, the standard LBM is usually applied on the Cartesian mesh. Due to this common feature, it is desirable to combine these two methods together. Many efforts have been made in this aspect. The first attempt was given by Feng and Michaelides [7,8]. They successfully applied IB-LBM to simulate the rigid particle motion. Niu et al. [9] proposed the momentum exchange-based IB-LBM for simulation of several incompressible flows. Peng et al. [10] developed the multi-block IB-LBM for simulation of flows around a circular cylinder and an airfoil. Both works adopt the multi-relaxation LBM to get the flow field.

The key issue in the IBM is the computation of restoring force. The popular way is the penalty method [1,7]. This method introduces a user-defined spring parameter which may have a significant effect on the computational efficiency and accuracy. Another way is the direct forcing method, which was first introduced by Fadlun et al. [11]. As direct forcing method needs to solve N–S equations to compute the force at the boundary point, it may spoil the merits of LBM when IBM is combined with it. Recently, a simple method for computing the restoring force was proposed by Niu et al [9], in which the momentum exchange at the boundary is used to compute the force. As compared to the work of Ladd [12], the force computation of Niu et al. [9] is simpler and more convenient since the force is computed at the boundary points and one does not need to care the details of the boundary position and mesh points.

In the conventional IBM, the non-slip boundary condition is not enforced in its solution process. This is different from the conventional body-fitted solvers, where the non-slip condition is imposed at the boundary. For example, in the LBM, the non-slip condition is implemented by the bounce-back (BB) rule. Ladd [12] well applied BB-LBM to simulate solid–fluid suspensions. Since then, many efforts have been made to develop high-order BB-LBM to simulate flows around complex geometries and moving objects. Bouzidi et al. [13] presented quadratic interpolation to implement bounce-back boundary condition on the boundary of moving objects. The implementation has the second-order accuracy, and the circular Couette flow and steady flow over a periodic array of circular cylinders are accurately simulated. Chun and Ladd [14] proposed an equilibrium interpolation LBM for simulation of flows in narrow gaps. Only the equilibrium part of distribution function is interpolated on the boundary to achieve the second-order accuracy. With the help of multi-relaxation time model, this method can accurately simulate highly viscous flows. As compared to the body-fitted solvers, the major drawback of IBM is that the non-slip boundary condition is only approximately satisfied at the converged state. As a consequence, some streamlines may penetrate the solid body. To remove this drawback, Kim et al. [15] introduced the mass source or sink into the computation. Using this way, the non-slip condition can be well kept, but the complexity is introduced into the computation. Recently, Shu et al. [16] found that unsatisfying of non-slip boundary condition in IBM is in fact due to pre-calculated restoring force. Using the fractional step technique, they concluded that, adding a body force in the momentum equations is equivalent to make a correction in the velocity field. To enforce the non-slip boundary condition, the velocity correction (or restoring force) should be considered as unknown, which is determined in such a way that the velocity at the boundary interpolated from the corrected velocity field satisfies the non-slip boundary condition. In the work of Shu et al. [16], the velocity correction is made at adjacent points to the boundary along the horizontal and vertical mesh lines. The approach is very simple. However, it only has the first-order accuracy and the computed forces at the boundary have some oscillations. The reason may be that the linear relationship is applied along the horizontal/vertical mesh lines and the smooth Dirac delta function is not used.

In this work, we will follow the idea of Shu et al. [16] to propose a variant of IB-LBM. To effectively consider the body force in the lattice Boltzmann equation (LBE), we adopt the model proposed by Guo et al. [17]. As shown in [17], the conventional LBE with body force such as the one used in [7,8] cannot properly consider the discrete lattice effects to the density and

momentum. To remove this drawback, Guo et al. [17] proposed a representation of forcing term in the LBE which is similar to the form in the work of He et al. [18]. In this model, the velocity is naturally contributed by two parts. One is from the density distribution function, while the other is from the body force. Similar to the work of Shu et al. [16], we can term the velocity from the density distribution function as the intermediate velocity, and the velocity from the body force as the velocity correction. In the conventional IB-LBM, the velocity correction is pre-computed and cannot be manipulated to satisfy the non-slip boundary condition. In this work, it is considered as unknown and is determined from enforcement of non-slip condition. Through the relationship between the force density and velocity correction, the drag and lift forces can be directly computed from the obtained velocity correction. The solution process is exactly the same as the conventional IB-LBM except that the body force is determined by enforcing the non-slip boundary condition in this work. The present method is validated by its application to simulate the steady and unsteady flows past a circular cylinder and airfoil. The obtained results basically agree well with available data in the literature. Since the boundary condition is accurately satisfied, the obtained numerical results do not show any streamline penetration to the solid body.

2. Conventional immersed boundary-lattice Boltzmann method

For the viscous incompressible flows in a two-dimensional domain Ω containing an immersed boundary in the form of closed curve Γ, as shown in Fig. 1, the governing equations of immersed boundary method can be written as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)+\nabla p=\mu\Delta\mathbf{u}+\mathbf{f}, \tag{1}$$

$$\nabla\cdot\mathbf{u}=0, \tag{2}$$

$$\mathbf{f}(\mathbf{x},t)=\int_{\Gamma}\mathbf{F}(s,t)\delta(\mathbf{x}-\mathbf{X}(s,t))\,ds, \tag{3}$$

$$\frac{\partial \mathbf{X}(s,t)}{\partial t}=\mathbf{u}(\mathbf{X}(s,t),t)=\int_{\Omega}\mathbf{u}(\mathbf{x},t)\delta(\mathbf{x}-\mathbf{X}(s,t))\,d\mathbf{x} \tag{4}$$

$$\mathbf{F}(s,t)=\mathbf{S}(\mathbf{X}(s,t),t). \tag{5}$$

Here $\mathbf{x}$, $\mathbf{u}$, p and $\mathbf{f}$ are the Eulerian coordinates, fluid velocity, fluid pressure and force density acting on the fluid phase, respectively. $\mathbf{X}$ and $\mathbf{F}$ stand for the Lagrangian coordinates and boundary force density. $\delta(\mathbf{x}-\mathbf{X}(s,t))$ is the Dirac delta function. Eqs. (1) and (2) are the N–S equations of viscous and incompressible fluid flows. Eqs. (3) and (4) describe the interaction between the immersed boundary and the fluid flow, by distributing the boundary force at the Lagrangian points to Eulerian points and interpolating the velocity at the Eulerian points to Lagrangian points. Eq. (5) states that the boundary force on the segment is determined by the boundary configuration at time t, where the function $\mathbf{S}$ satisfies a generalized Hooke's law if the boundary is elastic.

In order to represent Eqs. (1) and (2) in the lattice Boltzmann frame, an external forcing term should be added to the standard LBE. The modified LBE can be written as [7,8]

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta t,t+\delta t)-f_\alpha(\mathbf{x},t)=-\frac{1}{\tau}(f_\alpha(\mathbf{x},t)-f_\alpha^{eq}(\mathbf{x},t))+\frac{3}{2}w_\alpha\mathbf{f}\cdot\mathbf{e}_\alpha\delta t, \tag{6}$$

where f_α is the distribution function; f_α^{eq} is its corresponding equilibrium state; τ is the single relaxation parameter; $\mathbf{e}_\alpha$ is the particle velocity and $\mathbf{f}$ is the external force density added. w_α are coefficients which depend on the selected lattice velocity model. For the D2Q9 model [19], the velocity set is given by

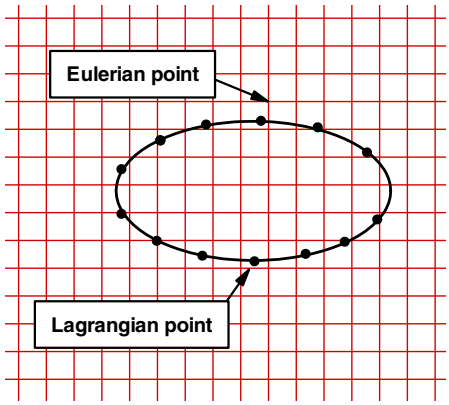


Fig. 1. Immersed boundary illustration. The solid points are used to represent the Lagrangian points and the Eulerian points are represented by the intersection points of mesh lines.

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

where $c = \delta x/\delta t$, δx and δt are the lattice spacing and time step. For the case of $\delta x = \delta t$, c is taken as 1. The corresponding equilibrium distribution function is

$$f_\alpha^{eq}(\mathbf{x},t) = \rho w_\alpha \left[1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2 - (c_s|\mathbf{u}|)^2}{2c_s^4}\right], \tag{8}$$

with $w_0 = 4/9$, $w_1 = w_2 = w_3 = w_4 = 1/9$ and $w_1 = w_2 = w_3 = w_4 = 1/36$. $c_s = c/\sqrt{3}$ is the sound speed of the model.

To model the interaction of fluid and rigid body, Feng and Michaelides used the penalty method [7] and direct forcing method [8] to compute the boundary force density. For the penalty method, at time t, it can be assumed that the center of mass of the rigid body is at $\mathbf{X}(t)$, and the instantaneous body rotational matrix is $\mathbf{R}(t)$. So the position of a reference point $\mathbf{X}_j^r$ may be determined by

$$\mathbf{X}_j^r(t) = \mathbf{X}(t) + \mathbf{R}(t)[\mathbf{X}_j^r(0) - \mathbf{X}(0)]. \tag{9}$$

For the boundary point $\mathbf{X}_j^t$ which is correspondent to the reference point $\mathbf{X}_j^r$, it is allowed to be slightly deformed by the fluid. When the reference point and the boundary point are not at the same position, there occurs a displacement $\xi_j = \mathbf{X}_j^t - \mathbf{X}_j^r$, and a restoration force $\mathbf{F}_j$ is generated that tends to restore the boundary point back to the reference point. It can be modeled by a linear spring relation

$$\mathbf{F}_j = -k\xi_j \tag{10}$$

where k is the spring constant.

To avoid using the user-defined parameter k in the calculation, the direct forcing method was proposed in the work of Feng and Michaelides [8]. It is assumed that Eq. (1) is also valid at the Lagrangian points which represent the boundary. So the force density at the boundary points can be written as

$$\mathbf{f} = \rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) + \nabla p - \mu\Delta\mathbf{u}. \tag{11}$$

If the velocity and pressure field at the time step $t = t_n$ are known, in order to satisfy the boundary condition at $t = t_{n+1}$, the velocity at the Lagrangian boundary points is equal to the velocity of the boundary $\mathbf{U}^{n+1}$ at the same point, so the force density at the Lagrangian points and $t = t_{n+1}$ can be determined using an explicit scheme

$$\mathbf{f}_i^{n+1} = \rho\left(\frac{\mathbf{U}_i^{n+1} - \mathbf{u}_i^n}{\delta t} + \mathbf{u}_i^n\cdot\nabla\mathbf{u}_i^n\right) + \nabla p_i^n - \mu\Delta\mathbf{u}_i^n. \tag{12}$$

Apart from the penalty and the direct forcing methods, there is a simple scheme called momentum exchange to compute the boundary force. This scheme was proposed by Niu et al [9]. By using the Lagrangian interpolated polynomials, the distribution functions on the boundary points at all lattice velocity directions are calculated. Hence, a new set of distribution functions on the boundary points could be achieved through the bounce-back rules

$$f_\beta(\mathbf{X}_B,t) = f_\alpha(\mathbf{X}_B,t) - 2w_\alpha\rho\frac{\mathbf{e}_\alpha\cdot\mathbf{U}_B}{c_s^2}, \tag{13}$$

where β denotes the opposite direction of α; $\mathbf{U}_B(=\mathbf{U} + \boldsymbol{\Omega}\times(\mathbf{X}_B - \mathbf{X}))$ is the velocity of the boundary with $\mathbf{U}$ and $\boldsymbol{\Omega}$ representing the translational and angular velocity of the rigid body, and $\mathbf{X}$ is the mass center of the body, $\mathbf{X}_B$ is its boundary position; w_α are the coefficients in the equilibrium distribution function. Consequently, the boundary force density can be calculated via the momentum exchange, that is,

$$\mathbf{F}(\mathbf{X}_B,t) = \sum_\beta \mathbf{e}_\beta[f_\beta(\mathbf{X}_B,t) - f_\alpha(\mathbf{X}_B,t)]. \tag{14}$$

The basic solution process of IB-LBM can be illustrated as follows:

(1) Compute the force density at the boundary points using Eqs. (10) or (12) or (14) and then distribute it into the Eulerian points using Eq. (3).
(2) Solve the lattice Boltzmann Eq. (6) with the force term $\mathbf{f}$ to update the velocity field.
(3) Interpolate the new (corrected) velocity field from the lattice (Eulerian points) to the boundary points using Eq. (4).
(4) Go back to (1) till the convergence criterion is satisfied.It should be noted that in the above process, although the wall velocity $\mathbf{U}^{n+1}$ has been used in the direct forcing method (Eq. (12)), like the penalty method (Eq. (10)) and the momentum exchange method (Eq. (14)), the force density is computed explicitly in step (1). With the known force density $\mathbf{f}$, the new (corrected) velocity field is also explicitly determined in step (2). However, there is no guarantee in step (3)

that the velocity at the boundary point obtained from interpolation of corrected velocity field at Eulerian points equals to the wall velocity. This means that the non-slip wall boundary condition may not be satisfied in step (3). This could be the major reason to cause flow penetration to the solid body in the conventional IB-LBM results. As shown in the following section, we will present a version of IB-LBM, in which the force density is considered as unknown in step (1), and the non-slip wall condition is enforced in step (3).

3. Implicit velocity correction-based immersed boundary-lattice Boltzmann method

We start with the LBE. As shown in [17,20], the LBE with external force (Eq. (6)) cannot properly consider the discrete lattice effects to the density and momentum. In order to correctly recover the viscous and incompressible N–S equations involving the external force, the contribution of the force to both momentum $\rho\mathbf{u}$ and momentum flux $\rho\mathbf{uu}$ should be considered. The first attempt to propose a better representation of forcing term in LBE was given by He et al. [18]. Later, Guo et al. [17] proposed a similar formulation with higher-order terms in $\mathbf{u}$. In this work, the form of LBE proposed by Guo et al. [17] is adopted, which can be written as

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta t, t+\delta t) - f_\alpha(\mathbf{x},t) = -\frac{1}{\tau}(f_\alpha(\mathbf{x},t) - f_\alpha^{eq}(\mathbf{x},t)) + F_\alpha\delta t, \tag{15}$$

$$F_\alpha = \left(1-\frac{1}{2\tau}\right)w_\alpha\left(\frac{\mathbf{e}_\alpha-\mathbf{u}}{c_s^2}+\frac{\mathbf{e}_\alpha\cdot\mathbf{u}}{c_s^4}\cdot\mathbf{e}_\alpha\right)\cdot\mathbf{f}, \tag{16}$$

$$\rho\mathbf{u} = \sum_\alpha \mathbf{e}_\alpha f_\alpha + \frac{1}{2}\mathbf{f}\delta t. \tag{17}$$

Here $\mathbf{f}$ is the force density at the Eulerian point, which is distributed from the Lagrangian boundary points; w_α are the coefficients in the equilibrium distribution function. It can be seen clearly from Eq. (17) that the fluid velocity consists of two parts. One is contributed from the density distribution function (solution of Eq. (15)), and the other is contributed from the force density $\mathbf{f}$. If we define the intermediate velocity $\mathbf{u}^*$ as

$$\rho\mathbf{u}^* = \sum_\alpha \mathbf{e}_\alpha f_\alpha \tag{18}$$

and the velocity correction $\delta\mathbf{u}$ as

$$\rho\delta\mathbf{u} = \frac{1}{2}\mathbf{f}\delta t \tag{19}$$

then Eq. (17) can be written as

$$\mathbf{u} = \mathbf{u}^* + \delta\mathbf{u}. \tag{20}$$

In the conventional IBM, $\mathbf{f}$ is computed in advance. From Eqs. (19) and (20), we can see clearly that the velocity correction $\delta\mathbf{u}$ and corrected velocity $\mathbf{u}$ are explicitly calculated. However, there is no guarantee that the velocity at the boundary point interpolated from $\mathbf{u}$ satisfies the non-slip boundary condition. To overcome this drawback, we have to consider the force density $\mathbf{f}$ as unknown, which is determined in such a way that the velocity at the boundary point interpolated from $\mathbf{u}$ satisfies the non-slip boundary condition. As shown in Fig. 2, the velocity correction $\delta\mathbf{u}$ at Eulerian points is distributed from the velocity correction at the boundary (Lagrangian) points. In the IBM, the boundary of rigid body is represented by a set of Lagrangian points $\mathbf{X}_B(s_l,t)$, $l = 1, 2, \cdots, m$. Here, we can set an unknown velocity correction vector $\delta\mathbf{u}_B^l$ at every Lagrangian point. The velocity correction $\delta\mathbf{u}$ at the Eulerian point can be obtained by the following Dirac delta function interpolation

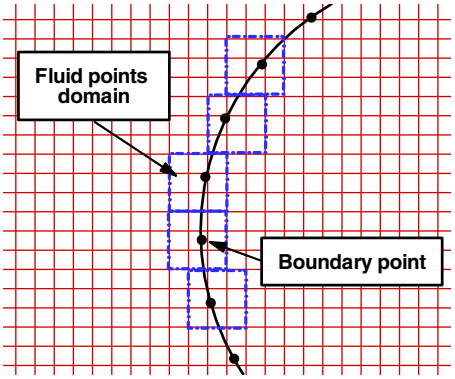


Fig. 2. Configuration of boundary points and their surrounding fluid points. The domain enclosed by the blue dash dot line is the region where the fluid phase is affected by one boundary point. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\delta\mathbf{u}(\mathbf{x},t)=\int_{\Gamma}\delta\mathbf{u}_B(\mathbf{X}_B,t)\delta(\mathbf{x}-\mathbf{X}_B(s,t))\,ds \tag{21}$$

In the actual implementation, $\delta(\mathbf{x}-\mathbf{X}_B(s,t))$ is smoothly approximated by a continuous kernel distribution D_{ij}, which was proposed by Peskin [1,21]

$$\delta(r)=\begin{cases}\frac{1}{4}(1+\cos(\pi|r|/2)) & |r|\leqslant 2\\ 0 & |r|>2\end{cases} \tag{22}$$

$$D_{ij}(\mathbf{x}_{ij}-\mathbf{X}_B^l)=\delta(x_{ij}-X_B^l)\delta(y_{ij}-Y_B^l) \tag{23}$$

Using Eqs. (23), (21) can be approximated by

$$\delta\mathbf{u}(\mathbf{x}_{ij},t)=\sum_l\delta\mathbf{u}_B^l(\mathbf{X}_B^l,t)D_{ij}(\mathbf{x}_{ij}-\mathbf{X}_B^l)\Delta s_l\quad(l=1,2,\cdots,m) \tag{24}$$

where Δs_l is the arc length of the boundary element. According to Eq. (20), the fluid velocity at Eulerian points can be corrected as

$$\mathbf{u}(\mathbf{x}_{ij},t)=\mathbf{u}^*(\mathbf{x}_{ij},t)+\delta\mathbf{u}(\mathbf{x}_{ij},t) \tag{25}$$

where $\mathbf{u}^*(\mathbf{x}_{ij},t)$ is the intermediate fluid velocity obtained from Eq. (18). Note that the unknowns in Eqs. (24) and (25) are the velocity corrections at the boundary points, $\delta\mathbf{u}_B^l$. To satisfy the non-slip boundary condition, the fluid velocities at the boundary points obtained by interpolation using the smooth delta function must be equal to the wall velocities $\mathbf{U}_B^l(\mathbf{X}_B^l,t)$ at the same positions. Its mathematical expression is

$$\mathbf{U}_B^l(\mathbf{X}_B^l,t)=\sum_{i,j}\mathbf{u}(\mathbf{x}_{ij},t)D_{ij}(\mathbf{x}_{ij}-\mathbf{X}_B^l)\Delta x\Delta y \tag{26}$$

Substituting Eq. (25) into Eq. (26), we can get the following equation system

$$\mathbf{U}_B^l(\mathbf{X}_B^l,t)=\sum_{i,j}\mathbf{u}^*(\mathbf{x}_{ij},t)D_{ij}(\mathbf{x}_{ij}-\mathbf{X}_B^l)\Delta x\Delta y+\sum_{i,j}\sum_l\delta\mathbf{u}_B^l(\mathbf{X}_B^l,t)D_{ij}(\mathbf{x}_{ij}-\mathbf{X}_B^l)\Delta s_l D_{ij}(\mathbf{x}_{ij}-\mathbf{X}_B^l)\Delta x\Delta y \tag{27}$$

Equation system (27) can be further re-written as the following matrix form

$$\mathbf{AX}=\mathbf{B}, \tag{28}$$

where $\mathbf{X}=\{\delta\mathbf{u}_B^1,\delta\mathbf{u}_B^2,\cdots,\delta\mathbf{u}_B^m\}$; $\mathbf{B}=\{\Delta\mathbf{u}_1,\Delta\mathbf{u}_2,\cdots,\Delta\mathbf{u}_m\}^T$ with

$$\Delta\mathbf{u}_l=\mathbf{U}_B^l(\mathbf{X}_B^l,t)-\sum_{i,j}\mathbf{u}^*(\mathbf{x}_{ij},t)D_{ij}(\mathbf{x}_{ij}-\mathbf{X}_B^l)\Delta x\Delta y\quad(l=1,2,\cdots,m) \tag{29}$$

and the elements of matrix **A** are only related to the Lagrangian boundary points and their neighboring Eulerian points (see Fig. 2). By solving the above equation system (28), we can obtain the unknown velocity correction at all Lagrangian boundary points. Note that the number of unknowns in (28) is the same as the number of boundary points and the velocity corrections at all the boundary points are computed simultaneously by using a direct method or iterative method to solve (28). In this work, the direct method is adopted. After obtaining the velocity correction at the boundary point, the velocity correction and the corrected velocity at Eulerian points are then calculated by Eqs. (24) and (25).

Another advantage of present method is the simple calculation of force on the boundary point. From Eq. (19), we can easily compute the force density by

$$\mathbf{f}=2\rho\delta\mathbf{u}/\delta t. \tag{30}$$

In our simulation, the density and pressure are calculated by

$$\rho=\sum_\alpha f_\alpha,\quad P=c_s^2\rho. \tag{31}$$

The basic solution procedure of present approach can be outlined below,

(1) Set initial values, compute the elements of matrix **A** and get $\mathbf{A}^{-1}$.
(2) Use Eq. (15) to obtain the density distribution function at time $t=t_n$ (initially setting $F_\alpha=0$) and compute the macroscopic variables using Eqs. (18) and (31).
(3) Solve equation system (28) to get the velocity corrections at all boundary points and use Eq. (24) to get velocity corrections at Eulerian points.
(4) Correct the fluid velocity at Eulerian points using Eq. (25) and obtain the force density using Eq. (30).
(5) Compute the equilibrium distribution function using Eq. (8).
(6) Repeat step 2 to step 5 until convergence is reached.

It should be indicated that in the present approach, the force density **f** is determined by the non-slip boundary condition, which is different from the conventional IB-LBM where **f** is given either by the penalty method or the direct forcing method or the momentum exchange method. In this work, the velocity at the boundary point obtained from interpolation of corrected velocity field at Eulerian points is enforced to the wall velocity. This idea has also been used in the work of Lowe et al. [22], where the new fluid velocity with consideration of force and torque at the boundary point is set to the wall velocity of the colloidal particle.

4. Some numerical examples

4.1. Numerical test of overall accuracy

LBM is adopted in the present work to obtain the basic flow field. As we know, LBM has the second-order accuracy in time and space. However, when LBM is combined with IBM, the Dirac delta function interpolation and the smoothing kernel are used to get the velocity correction at Eulerian points and the velocity at boundary points. The interpolation only has the first-order accuracy. Although it is only applied in the region nearby the boundary, it may have effect on the global accuracy of solution in the whole domain. As the process is very complicated, it is impossible to do a theoretical analysis for such an effect. In the following, we will take the decaying vortex problem as an example to access the overall accuracy of present method. The test problem has been chosen by Chen et al. [23] to access the accuracy of their IBM results. The problem is defined in a square domain of $-1 \leqslant x \leqslant 1$, $-1 \leqslant y \leqslant 1$, with periodic boundary conditions. Its exact solution is expressed as

$$u(x,y,t) = -\cos(x\pi)\sin(y\pi)e^{-(2t\pi^2/Re)}, \tag{32}$$

$$v(x,y,t) = \sin(x\pi)\cos(y\pi)e^{-(2t\pi^2/Re)}, \tag{33}$$

$$\rho(x,y,t) = \rho_0 - \frac{\rho_0 U^2}{4C_s^2}[\cos(2x\pi) + \sin(2y\pi)]e^{-(4t\pi^2/Re)}. \tag{34}$$

Similar to the work of Chen et al. [23], for simplicity, the exact solution is fixed no matter what the solid body is immersed in the computational domain. The exact solution is used to provide the initial condition at $t = 0$ and the boundary condition on the boundary including the outer boundary and the surface of immersed body. It is also used to compute the numerical error so that the overall accuracy of solution can be measured. As in Chen et al. [23], the simulation is carried out at $Re = 10$ and the single relaxation time of $\tau = 0.65$. A circular cylinder with radius of 0.5 is immersed in the computational domain. Four different meshes of 21×21, 41×41, 81×81, and 161×161 are used in the simulation. The solutions at $t = 1$ are obtained and the numerical error of u is quantified using L_2 norm, which is defined as

$$L_2\ error = \sqrt{\frac{\sum_N (u^{numerical} - u^{exact})^2}{N}}.$$

Here $u^{numerical}$ and u^{exact} represent the numerical velocity and the exact solution, respectively, N is the total number of Eulerian points in the whole domain. The numerical error versus mesh spacing in the log scale is plotted in Fig. 3. As can be seen in Fig. 3, the overall accuracy of numerical results is slightly less than the second-order as the slope of the line is about 1.9. The reduction of overall accuracy is attributed to the use of the first-order smoothing kernel in interpolation.

4.2. Flow over an array of circular cylinders placed at the middle of a straight channel

The second test example is the simulation of flow past an array of circular cylinders placed at the middle of a straight channel. This problem has been investigated by Aidum and Lu [24] and Inamuro et al. [25] using the finite element method and the lattice Boltzmann method. Similar to previous studies [24,25], this problem can be simplified to placing a fixed cylinder at the middle of the domain and two walls of the channel moving tangentially with a constant velocity. The periodic boundary condition is applied at the inlet and outlet of channel.

In the present simulation, a uniform mesh size of 128×128 is used and the Reynolds number based on the velocity of wall U_w and diameter of the cylinder D is set as 1. Three sets of velocity and cylinder diameter are considered: (1) $D = 10.8\Delta x$, $U_w = 0.04$; (2) $D = 20.8\Delta x$, $U_w = 0.02$; and (3) $D = 60.8\Delta x$, $U_w = 0.01$. Here, Δx is the mesh spacing. Since the mesh size is fixed as 128×128 for the three cases, when D is increased, the cylinder will take more and more space in the whole computational domain. Two-dimensionless forces are calculated for comparison. They are $f_w = F_w/(\rho L U_w^2)$, the force per unit area of the channel wall, and $f_c = F_c/(\rho\pi D U_w^2)$, the force per unit area of the cylinder surface. Here, F_w and F_c are the total forces acting on the channel wall and cylinder surface, respectively; L is the length of wall in the simulation. The comparison of f_c and f_w with previous results for the three cases is shown in Table 1. Basically, the present results agree well with those of Aidum and Lu [24] and Inamuro et al. [25] for all the three cases. In fact, the present result lies in-between the data of Aidum and Lu [24] and that of Inamuro et al. [25].

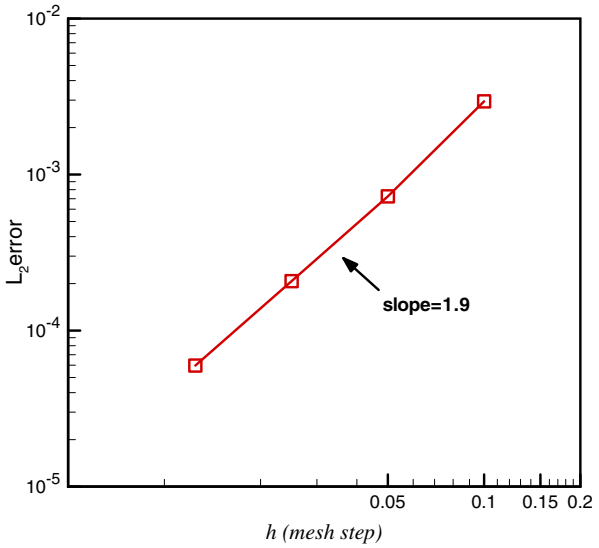


Fig. 3. Convergence of numerical error versus mesh spacing for decaying vortex problem. The overall accuracy of proposed IB-LBM is about 1.9.

Table 1
Comparison of forces f_c and f_w for flow over an array of circular cylinders placed at the middle of a straight channel. Here, f_c and f_w are the dimensionless force per unit area of the cylinder surface and the channel wall, respectively.

Cases	References	f_c	f_w
$D = 10.8\Delta x$, $U_w = 0.04$	Aidum and Lu [24] FEM	0.966	0.137
	Aidum and Lu [24] LBM	1.022	0.132
	Inamuro et al. [25]	1.053	0.142
	Present	1.044	0.139
$D = 20.8\Delta x$, $U_w = 0.02$	Aidum and Lu [24] FEM	1.158	0.316
	Aidum and Lu [24] LBM	1.229	0.313
	Inamuro et al. [25]	1.251	0.322
	Present	1.246	0.318
$D = 60.8\Delta x$, $U_w = 0.01$	Aidum and Lu [24] FEM	2.067	1.543
	Aidum and Lu [24] LBM	2.054	1.532
	Inamuro et al. [25]	2.093	1.561
	Present	2.065	1.541

4.3. Flows over a circular cylinder

The flow over a circular cylinder is chosen to further validate the proposed method. This problem has been studied extensively and there are many theoretical, experimental, and numerical results available. Depending on the Reynolds number, different kind of flow behaviors can be characterized. The Reynolds number in this flow is defined as

$$Re = \frac{U_\infty D}{\upsilon}, \tag{35}$$

where D is the diameter of the cylinder. As pointed out by Lai and Peskin [2], the drag force arises from two sources: the shear stress and the pressure distribution along the body. In the present work, the force can be directly computed from the velocity correction. The drag coefficient is defined as

$$C_d = \frac{F_D}{(1/2)\rho U_\infty^2 D}, \tag{36}$$

where F_D is the drag force. Here, it can be calculated by

$$F_D = -\int_\Omega f_x d\mathbf{x}, \tag{37}$$

where f_x is the x component of force density $\mathbf{f}$.

When the body starts shedding a vortex, a lift force on the body is generated by the fluid. Similar to the drag coefficient, the lift coefficient can be defined as

$$C_l = \frac{F_L}{(1/2)\rho U_\infty^2 D}, \tag{38}$$

where F_L is the lift force, which can be calculated by

$$F_L = -\int_\Omega f_y d\mathbf{x}, \tag{39}$$

where f_y is the y component of force density $\mathbf{f}$. For the unsteady flow, the Strouhal number is defined as the dimensionless frequency with which the vortices are shed from the body

$$St = \frac{f_q D}{U_\infty}, \tag{40}$$

where f_q is the vortex shedding frequency.

In the present simulation, the fluid density is taken as $\rho = 1.0$ and the free stream velocity is $U_\infty = 0.1$. The cylinder surface is represented by 120 Lagrangian points with uniform distribution. The computation starts with the given free stream velocity. At the far field boundaries of the square domain Ω, the equilibrium distribution functions are used to implement the boundary conditions. To obtain high resolution near the cylinder surface and in the meantime save the computational effort, fine grid is used around the cylinder, and the coarse grid is put near the far field boundaries. To accommodate the use of non-uniform mesh, the Taylor series expansion and least square-based LBM (TLLBM) [26] is employed in the present computation. The TLLBM is based on the standard LBM with introduction of Taylor series expansion in the spatial direction and least-squares optimization. Its final form is an algebraic formulation, in which the coefficients only depend on the coordinates of mesh points and lattice velocity, and are computed in advance. It essentially has no limitation on the mesh structure and lattice model. The trade-off of TLLBM is that additional memory is needed to store the coefficients.

For the simulations at $Re = 20$ and 40 (steady flow), the computational domain is $40D \times 40D$ with the whole mesh size of 401×401. The region around the circular cylinder is $1.2D \times 1.2D$ with a uniform mesh size of 97×97. The cylinder is located at $(16D, 20D)$. Figs. 4 and 5 show the streamlines when flows reach their steady state for the cases of $Re = 20$ and 40, respectively. Also included in these figures are the results using conventional IB-LBM proposed by Feng and Michaelides [8]. It can be seen clearly that the streamlines obtained by the present method do not penetrate the cylinder surface. In fact, the streamlines inside the cylinder are enclosed by the boundary of cylinder. This means that there is no mass exchange between the fluid inside the cylinder and the fluid outside the cylinder. In contrast, the penetration of streamlines to the cylinder surface by the conventional IB-LBM is very obvious. The main reason behind this is that the non-slip boundary condition is accurately enforced in the present method while it is only approximately satisfied in the conventional IB-LBM.

When the flows reach steady state, a pair of stationary recirculating eddies develop behind the cylinder. The length of the recirculation region from the rearmost point of the cylinder to the end of the wake increases with the Reynolds

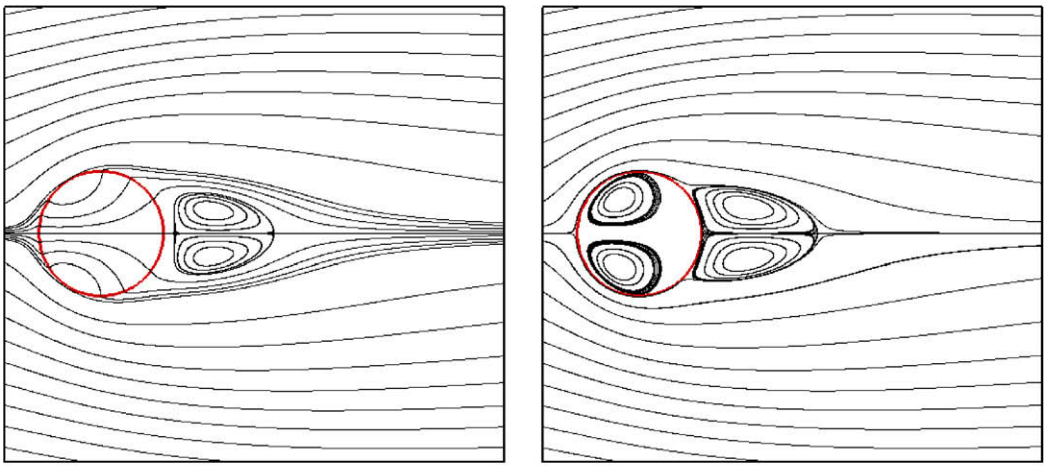


Fig. 4. Streamlines for the flow over a circular cylinder at $Re = 20$. Both the results of present IB-LBM and conventional IB-LBM are included. For the conventional IB-LBM, the direct forcing scheme is used to compute the restoring force.

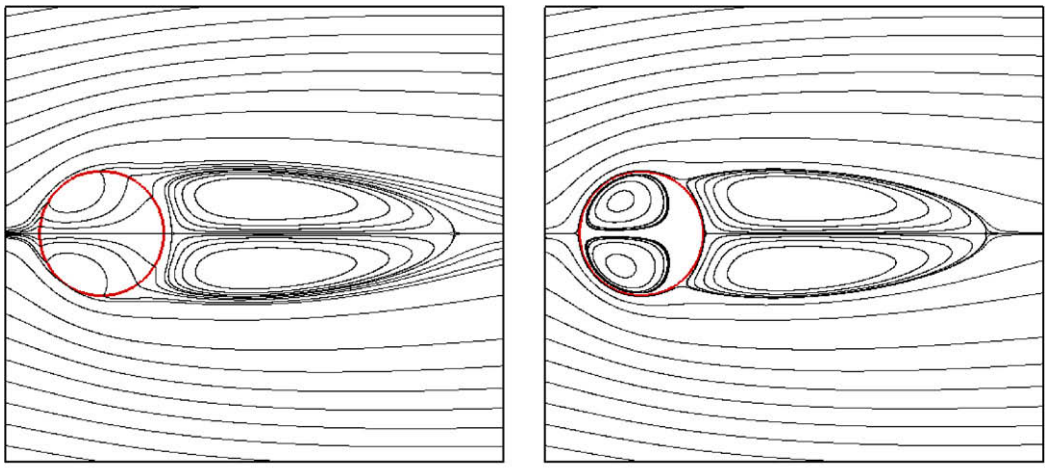


Fig. 5. Streamlines for the flow over a circular cylinder at $Re = 40$.

number. The drag coefficient C_d and length of recirculation region L (scaled by $D/2$) are compared with the previous results [27–29] in Table 2. From the table, it can be found that our numerical results basically agree well with those in the literature.

In the present work, we need to solve equation system (27) to obtain the velocity correction at the boundary points. The process involves inverting the element matrix **A**. For the flow around stationary objects, this needs to be done only once as the elements of matrix **A** are constants. However, for the flow around moving objects, the inverse of the element matrix **A** has to be done at every time step. This would increase the computational effort. On the other hand, we notice that the number of boundary points is much less than the number of Eulerian points in the flow field, and the dimension of matrix **A** is the same as the number of boundary points. The increment of computational effort due to inverse of matrix **A** at every time step may not be very much. To study this issue, we simulate the same problem with two different ways. One is the flow around the stationary cylinder with a free stream velocity, and the other is the cylinder moving with a constant velocity in the stationary fluid. Note that when the flow around the stationary cylinder is simulated, the use of non-uniform mesh and TLLBM can greatly improve the computational efficiency. However, when the flow around the moving cylinder is simulated and non-uniform mesh is used, the mesh must be changed at every time step. This brings complexity into the computation. To make a fair comparison and ease the computation for the moving cylinder case, the uniform mesh and standard LBM are used in this study. The Reynolds number is taken as $Re = 20$, and the computational domain is set by $32D \times 32D$ with a mesh size of 1281×1281. The stationary cylinder is located at $(8D, 16D)$, while the moving cylinder moves towards the left from the position of $(30D, 16D)$. Notice that the origin of the frame is at the lower corner of the left boundary. It was found that numerical results obtained by two ways are the same. This can be seen clearly in Figs. 6 and 7, where the vorticity distribution and pressure profile on the surface of cylinder are compared for the two ways. The agreement of two results well demonstrates Galilean invariance of present approach. Fig. 8 compares the CPU time required by the two ways when the steady state resolution is reached. The moving cylinder case needs a little bit more CPU time than the stationary cylinder case. This reveals that the increment of CPU time due to inverse of the element matrix **A** at every time step is very little.

Table 2
Comparison of drag coefficient C_d and recirculation length L for steady flow over a circular cylinder at $Re = 20$ and 40. Previous numerical results are also included in this table for comparison.

Case	References	C_d	L
$Re = 20$	Dennis and Chang [27]	2.045	1.88
	Fornberg [28]	2.000	1.82
	He and Doolen [29]	2.152	1.842
	Present	2.091	1.86
$Re = 40$	Dennis and Chang [27]	1.522	4.69
	Fornberg [28]	1.498	4.48
	He and Doolen [29]	1.499	4.49
	Present	1.565	4.62

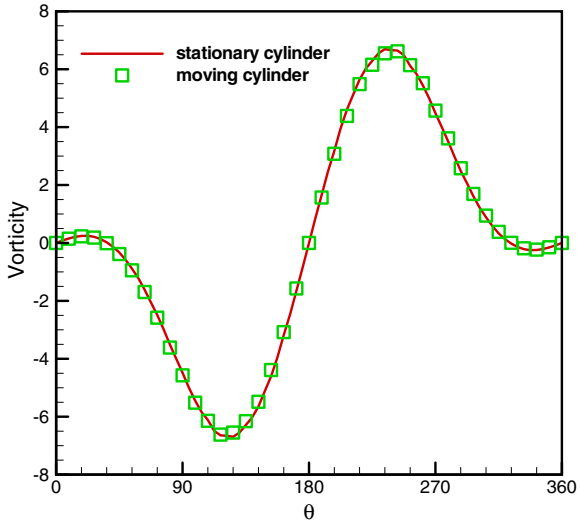


Fig. 6. Vorticity distribution on the surface of cylinder at $Re = 20$ for stationary and moving cylinder cases. The Galilean invariance of proposed method is tested.

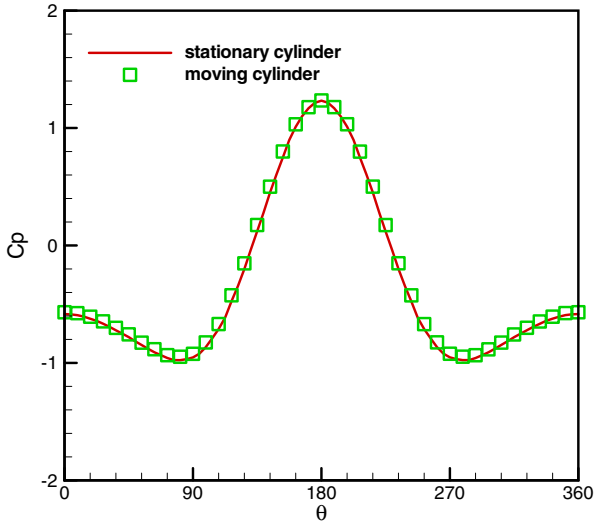


Fig. 7. Pressure distribution on the surface of cylinder at $Re = 20$ for stationary and moving cylinder cases.

For the simulations at $Re = 100$ and 200 (unsteady flow), the computational domain is taken as $50D \times 40D$ and the corresponding mesh size is 501×351. The region around the circular cylinder is $1.2D \times 1.2D$ with a uniform mesh size of 97×97. The cylinder is located at $(20D, 20D)$. Table 3 shows our results of drag and lift coefficients together with those of Lai and Peskin [2] at $Re = 100$. Table 4 provides our results of time-averaged drag coefficient and Strouhal number together with previous experimental and numerical results [30–33].

From Table 2, we can see that the present drag coefficients have a better agreement with the second-order result of Lai and Peskin [2] than their first-order counterpart. It is believed that the force calculation in the present work is more accurate than the conventional IB-LBM. This is because when the penetration happens, the boundary points may serve as holes for mass exchange between interior and exterior of the cylinder. The mass exchange would bring the momentum exchange, which will contribute to the force. Table 4 compares the present results with the previous experimental and computational results. The Strouhal numbers computed by our method are in agreement with the experimental data by Williamson, as reported in [33]. On the other hand, we can see from Tables 3 and 4 that the drag coefficient obtained by IB-LBM is generally higher than the experimental data and the result of body-fitted solver. This may imply that the force calculation by IB-LBM is not very accurate. The instantaneous vorticity contours of vortex shedding and the streamlines at $Re = 100$ and $Re = 200$ are

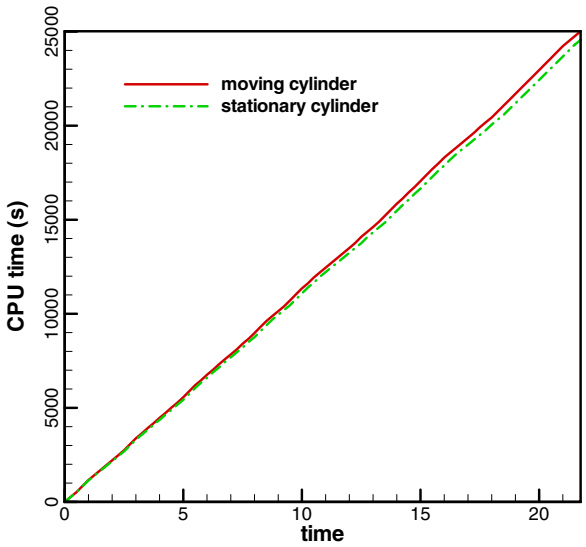


Fig. 8. CPU time required for stationary and moving cylinder cases at $Re = 20$.

Table 3
Comparison of drag coefficient C_d and lift coefficient C_l for unsteady flow over a cylinder at $Re = 100$. Here, the value of C_d is time-averaged. The first-order and second-order results of Lai and Peskin [2] are shown for comparison.

Method	C_d	C_l
1st order of Lai and Peskin [2]	1.4630	0.3290
2nd order of Lai and Peskin [2]	1.4473	0.3299
Present	1.364	0.344

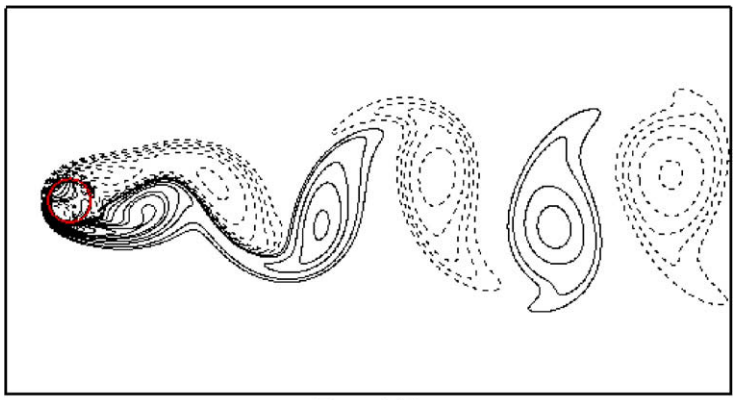
Table 4
Comparison of time-averaged drag coefficient C_d and Strouhal number St for unsteady flow over a cylinder at $Re = 100$ and 200. Both experimental and numerical results are included in this table for comparison.

Case	References	C_d	St
$Re = 100$	Gresho et al. [30]	1.76	0.18
	Saiki and Biringen [31]	1.26	0.171
	Clift et al. [32]	1.24	–
	Williamson reported in [33]	–	0.166
	Present	1.364	0.163
$Re = 200$	Gresho et al. [30]	1.76	0.21
	Saiki and Biringen [31]	1.18	0.197
	Clift et al. [32]	1.16	–
	Williamson reported in [33]	–	0.197
	Present	1.349	0.193

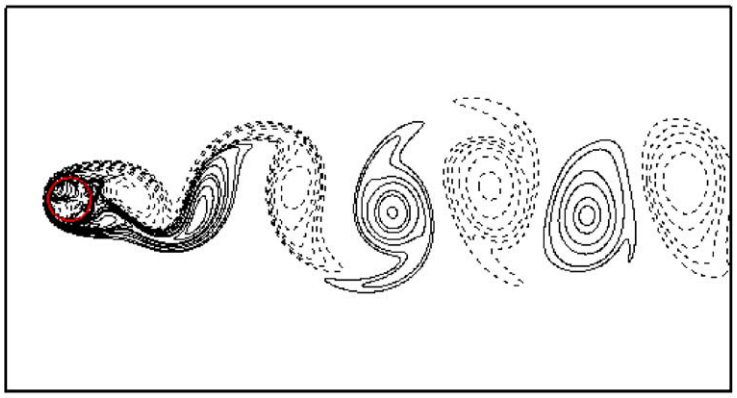
plotted in Figs. 9 and 10, respectively. The Karman vortex street of the flow over a circular cylinder can be clearly seen in these figures. Fig. 11 shows the time evolution of the drag and lift coefficients on the surface of cylinder at $Re = 100$ and 200. They both demonstrate clear periodicity.

4.4. Flow over a NACA-0012 airfoil

For the practical application of the implicit velocity correction-based immersed boundary-lattice Boltzmann method, the flow over a NACA-0012 airfoil at $Re = 500$ and the zero angle of attack is selected for study. Similar to the flow around the circular cylinder, the fluid density is taken as $\rho = 1.0$ and the free stream velocity is $U_\infty = 0.1$. The airfoil surface is represented by 120 Lagrangian points with uniform distribution. The computation starts with the given free stream velocity. At the far field boundaries of the square domain Ω, the equilibrium distribution functions are used to implement the boundary conditions. For this case, the pressure and drag coefficients are defined as



Re = 100



Re = 200

Fig. 9. Vorticity contours for flow over a cylinder at *Re* = 100 and 200. The dotted and solid lines denote the negative and positive levels of vorticity, respectively.

$$C_p = \frac{p_B - p_0}{(1/2)\rho U_\infty^2}, \quad C_d = \frac{F_D}{(1/2)\rho U_\infty^2 l}, \tag{41}$$

where l is the length of chord. Similar to Eq. (26), the pressure at the boundary of airfoil can be obtained from

$$p_B^l(\mathbf{X}_B^l) = \sum_{i,j} p(\mathbf{x}_{ij}) D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l)\Delta x \Delta y. \tag{42}$$

The pressure contours together with the streamlines and the distribution of pressure coefficient along the airfoil surface are presented in Figs. 12 and 13, respectively. They compare well with the numerical results obtained by the N–S solver of CFL3D [34]. For the drag coefficient, our result of $C_d = 0.1759$ is also in agreement with the value of 0.1762 reported by Lockard et al. [35] using an N–S equations-based finite difference solver.

For the unsteady flow around the NACA-0012 airfoil, the simulation at *Re* = 1000 with the angle of attack of 10°is carried out. The instantaneous vorticity contours and streamlines in one complete cycle are plotted in Fig. 14. These results compare well with those of Johnson and Tezduyar [36] who solved N–S equations using the finite element method. For the unsteady case, the Strouhal number is defined as

$$St = \frac{fl}{U_\infty}. \tag{43}$$

In our simulation, the calculated Strouhal number is 0.858, while the value given by Johnson and Tezduyar [36] is 0.86 and the value given by Mittal and Tezduya [37] is 0.862. Obviously, our result compares well with the data in the literature.

5. Conclusions

This paper presents a variant of immersed boundary-lattice Boltzmann method (IB-LBM). To well consider the effect of external force to the momentum and momentum flux as well as the discrete lattice effect, the lattice Boltzmann equation

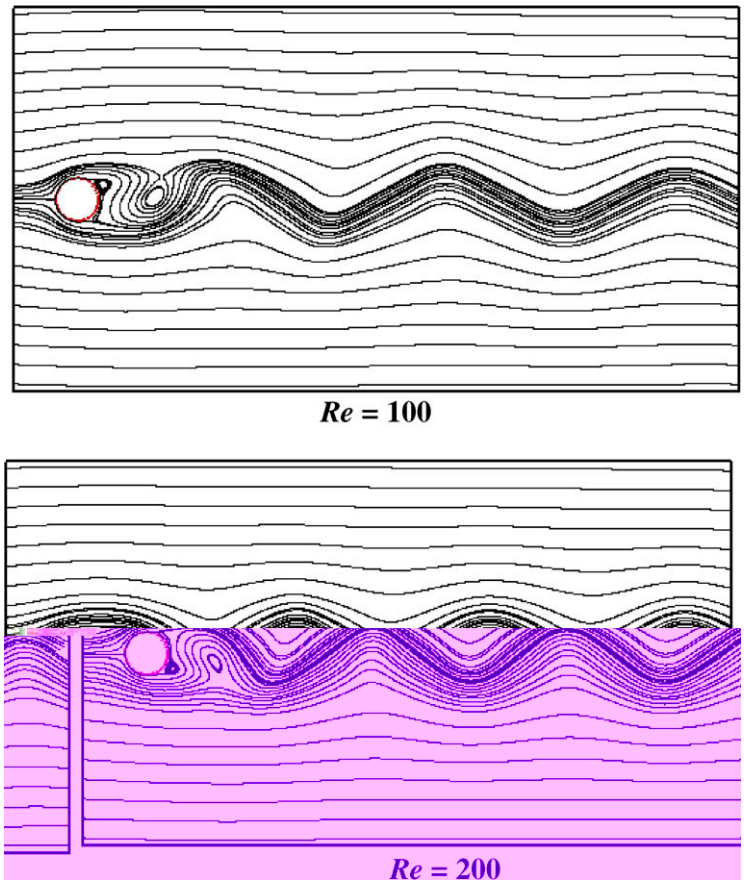


Fig. 10. Streamlines for flow over a cylinder at Re = 100 and 200.

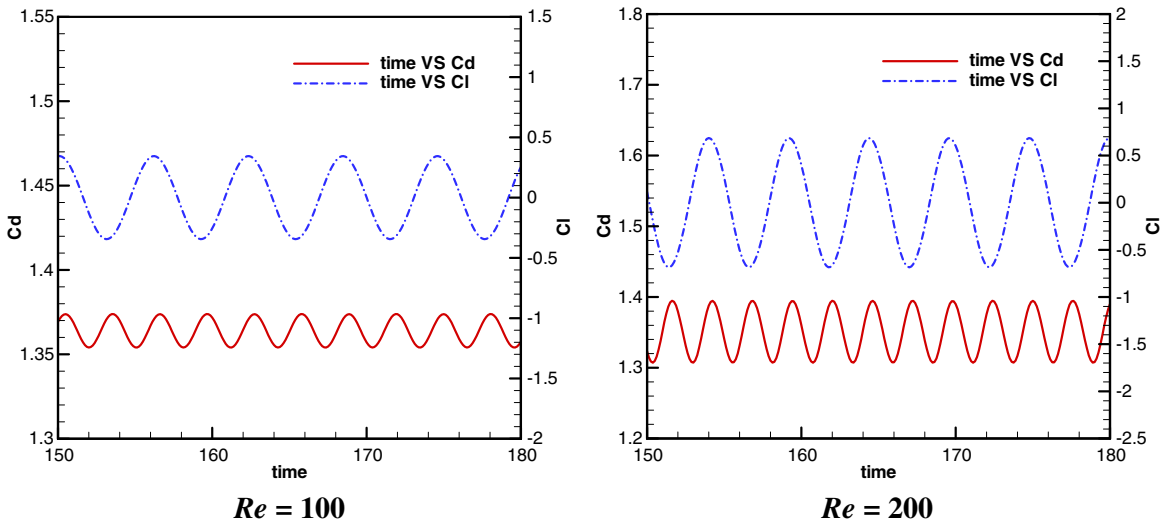


Fig. 11. Evolution of drag and lift coefficients for flow over a cylinder at Re = 100 and 200. Here, the time is dimensionless.

with external forcing term proposed by Guo et al. [17] is adopted. In this lattice Boltzmann model, both the density distribution function and the external force contribute to the fluid velocity. As the non-slip boundary condition at solid surface is usually for the velocity, the velocity contributed from the external force (force density in IB-LBM) is considered as velocity correction.

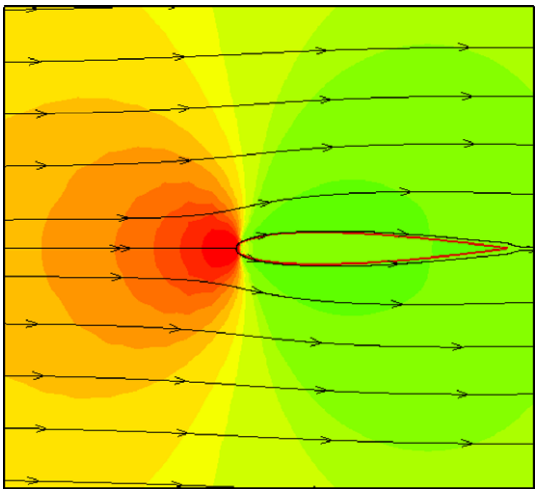


Fig. 12. Pressure contours and streamlines for flow over NACA-0012 airfoil at $Re = 500$ and AOA = 0.

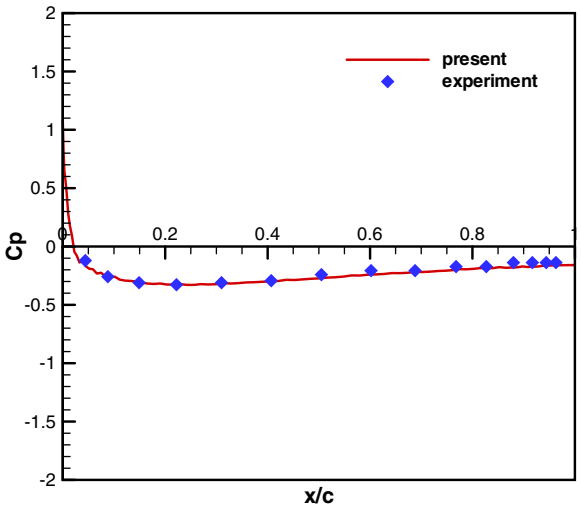


Fig. 13. Distribution of pressure coefficient C_p along the NACA-0012 airfoil at $Re = 500$ and AOA = 0. The result reported in [34] is included.

The conventional IB-LBM usually computes the force density explicitly. Thus, the velocity correction is fixed and cannot be manipulated to ensure the velocity at the boundary interpolated from the flow field being equal to the wall velocity. As a result, the non-slip boundary condition is not accurately satisfied. This leads to penetration of some streamlines to the solid body. The penetration may also degrade the accuracy of force calculation on the boundary. To improve the conventional IB-LBM, the velocity correction in the present model is considered as unknown, which is computed implicitly in such a way that the non-slip boundary condition is enforced at the boundary points. With this constraint, a set of algebraic equations for velocity corrections is obtained, which can be solved by the direct method to give velocity corrections at all boundary points simultaneously. The present IB-LBM keeps advantages of conventional IB-LBM such as simplicity and smooth solution. In fact, its solution procedure is exactly the same as the conventional IB-LBM except that the non-slip boundary condition is enforced in the present model. Another advantage of present method is the simple calculation of the force on the boundary. It is directly computed from the relationship between the velocity correction and the force density.

The present method is validated by its application to simulate steady and unsteady flows around a circular cylinder and airfoil. The obtained numerical results are in agreement with the data in the literature. It is believed that the present method has a great potential for practical application as it is simple and can accurately satisfy the non-slip boundary condition.

On the other hand, we have to indicate that although the streamline penetration is avoided in the present work, it still uses Dirac delta function interpolation near the boundary, which only has the first-order of accuracy. In this sense, it may not be as accurate as the body-fitted solver. To obtain accurate numerical results, one needs to use very fine mesh near the boundary. This could be a challenging issue for the case with very high Reynolds number.

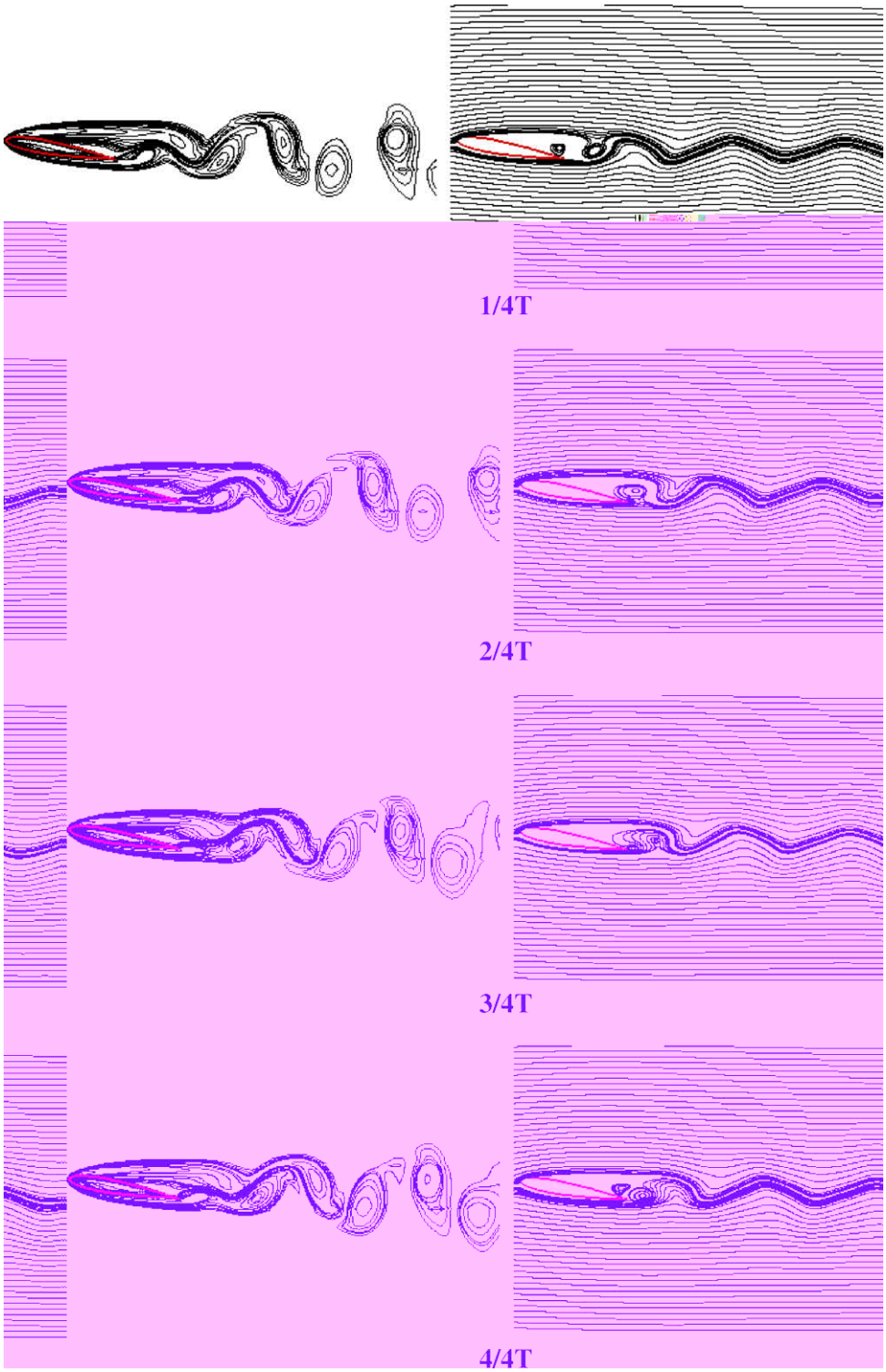


Fig. 14. Vorticity contours and streamlines for flow over NACA-0012 airfoil at $Re = 1000$ and AOA = 10° in a complete cycle. The flow at this situation is unsteady and periodic vortex shedding appears.

References

[1] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comp. Phys. 25 (1977) 220–252.

[2] M.C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comp. Phys. 160 (2000) 705–719.
[3] D. Goldstein, R. Hadler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, J. Comp. Phys. 105 (1993) 354–366.
[4] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex boundaries, J. Comp. Phys. 156 (1999) 209–240.
[5] L.F. Lima E Silva, A. Silveira-Neto, J.J.R. Damasceno, Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method, J. Comp. Phys. 189 (2003) 351–370.
[6] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, Ann. Rev. Fluid Mech. 30 (1998) 329–364.
[7] Z. Feng, E. Michaelides, The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems, J. Comp. Phys. 195 (2004) 602–628.
[8] Z. Feng, E. Michaelides, Proteus: a direct forcing method in the simulations of particulate flows, J. Comp. Phys. 202 (2005) 20–51.
[9] X.D. Niu, C. Shu, Y.T. Chew, Y. Peng, A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows, Phys. Lett. A 354 (2006) 173–182.
[10] Y. Peng, C. Shu, Y.T. Chew, X.D. Niu, X.Y. Lu, Application of multi-block approach in the immersed boundary-lattice Boltzmann method for viscous fluid flows, J. Comp. Phys. 218 (2006) 460–478.
[11] E. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three dimensional complex flow simulations, J. Comp. Phys. 161 (2000) 35–60.
[12] A.J.C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation Part I. Theoretical foundation, J. Fluid Mech. 271 (1994) 285–310.
[13] M. Bouzidi, M. Firdaouss, P. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries, Phys. Fluids 13 (2001) 3452–3459.
[14] B. Chun, A.J.C. Ladd, Interpolated boundary condition for lattice Boltzmann simulations of flows in narrow gaps, Phys. Rev. E 75 (2007) 066705.
[15] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comp. Phys. 171 (2001) 132–150.
[16] C. Shu, N.Y. Liu, Y.T. Chew, A novel immersed boundary velocity correction-lattice Boltzmann method and its application to simulate flow past a circular cylinder, J. Comp. Phys. 226 (2007) 1607–1622.
[17] Z. Guo, C. Zheng, B. Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, Phys. Rev. E 65 (2002) 046308.
[18] X. He, S. Chen, R. Zhang, A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh–Taylor instability, J. Comp. Phys. 152 (1999) 642–663.
[19] Y.H. Qian, D. d'Humieres, P. Lallemand, Lattice BGK model for Navier–Stokes equation, Europhys. Lett. 17 (1992) 479–484.
[20] A.J.C. Ladd, R. Verberg, Lattice-Boltzmann simulations of particle-fluid suspensions, J. Stat. Phys. 104 (2001) 1191–1251.
[21] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[22] C.P. Lowe, D. Frenkel, A.J. Masters, Long-time tails in angular momentum correlations, J. Chem. Phys. 103 (1995) 1582–1587.
[23] D.J. Chen, K.H. Lin, C.A. Lin, Immersed boundary method based lattice Boltzmann to simulate 2D and 3D complex geometry flows, Int. J. Mod. Phys. C 18 (2007) 585–594.
[24] C.K. Aidum, Y. Lu, Lattice Boltzmann simulation of solid particles suspended in fluid, J. Stat. Phys. 81 (1995) 49–61.
[25] T. Inamuro, K. Maeha, F. Ogino, Flow between parallel walls containing the lines of neutrally buoyant circular cylinders, Int. J. Multiphase Flow 26 (2000) 1981–2004.
[26] C. Shu, X.D. Niu, Y.T. Chew, Taylor-series expansion and least-squares-based lattice Boltzmann method: two-dimensional formulation and its applications, Phys. Rev. E 65 (2002) 036708.
[27] S.C.R. Dennis, G.Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds number up to 100, J. Fluid Mech. 42 (1970) 471–489.
[28] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, J. Fluid Mech. 98 (1980) 819–855.
[29] X. He, G. Doolen, Lattice Boltzmann method on curvilinear coordinates system: flow around a circular cylinder, J. Comp. Phys. 134 (1997) 306–315.
[30] P.M. Gresho, R. Chan, C. Upson, R. Lee, A modified finite element method for solving the time-dependent incompressible Navier–Stokes equations: part 2: applications, Int. J. Numer. Meth. Fluids 4 (1984) 619–640.
[31] E.M. Saiki, S. Biringen, Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method, J. Comp. Phys. 123 (1996) 450–465.
[32] R. Clift, J.R. Grace, M.E. Weber, Bubbles Drops and Particles, vol. 154, Academic Press, New York, 1978.
[33] C. Liu, X. Zheng, C.H. Sung, Preconditioned multigrid methods for unsteady incompressible flow, J. Comp. Phys. 139 (1998) 35–57.
[34] T. Imamura, K. Suzuki, T. Nakamura, M. Yoshida, Flow simulation around an airfoil using lattice Boltzmann method on generalized coordinates, AIAA 2004-0244.
[35] D.P. Lockard, L.S. Luo, S.D. Milder, B.A. Singer, Evaluation of powerflow for aerodynamic applications, J. Stat. Phys. 107 (2002) 423–478.
[36] A.A. Johnson, T.E. Tezduyar, Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces, Comput. Meth. Appl. Mech. Eng. 119 (1994) 73–94.
[37] S. Mittal, T.E. Tezduya, Massively parallel finite element computation of incompressible flows involving fluid–body interactions, Comput. Meth. Appl. Mech. Eng. 112 (1994) 253–282.