Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 218 (2006) 460–478
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Application of multi-block approach in the immersed boundary–lattice Boltzmann method for viscous fluid flows

Y. Peng [a], C. Shu [a,*], Y.T. Chew [a], X.D. Niu [a], X.Y. Lu [b]

[a] Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore
[b] Department of Modern Mechanics, University of Science and Technology of China, Hefei, Anhui 230026, PR China

Received 31 March 2005; received in revised form 17 February 2006; accepted 20 February 2006

Abstract

The immersed boundary–lattice Boltzmann method was presented recently to simulate the rigid particle motion. It combines the desirable features of the lattice Boltzmann and immersed boundary methods. It uses a regular Eulerian grid for the flow domain and a Lagrangian grid for the boundary. For the lattice Boltzmann method, as compared with the single-relaxation-time collision scheme, the multi-relaxation-time collision scheme has better computational stability due to separation of the relaxations of various kinetic models, especially near the geometric singularity. So the multi-relaxation-time collision scheme is used to replace the single-relaxation-time collision scheme in the original immersed boundary–lattice Boltzmann method. In order to obtain an accurate result, very fine lattice grid is needed near the solid boundary. To reduce the computational effort, local grid refinement is adopted to offer high resolution near a solid body and to place the outer boundary far away from the body. So the multi-block scheme with the multi-relaxation-time collision model is used in the immersed boundary–lattice Boltzmann method. In each block, uniform lattice spacing can still be used. In order to validate the multi-block approach for the immersed boundary–lattice Boltzmann method with multi-relaxation-time collision scheme, the numerical simulations of steady and unsteady flows past a circular cylinder and airfoil are carried out and good results are obtained.

Keywords: Lattice Boltzmann method; Immersed boundary method; Multi-block approach; Incompressible flow

1. Introduction

A novel numerical method called the immersed boundary method (IBM) was developed by Peskin in 1970s in order to model the blood flow in the heart [1]. Since then, numerous modifications and refinements have been proposed and a number of variants of this approach were proposed [2]. It uses a fixed Cartesian mesh for the fluid. For the boundaries that are immersed in the fluid, the method uses a set of boundary points. The basic idea of this method is to treat the boundary as deformable, but with high stiffness. A small distortion

* Corresponding author. Tel.: +65 6874 6476; fax: +65 6779 1459.
E-mail address: mpeshuc@nus.edu.sg (C. Shu).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.02.017

of the particle boundary will yield a force that tends to restore the boundary into its original shape. The balances of such forces are distributed into the Eulerian nodes of the grid and the Navier–Stokes equations with a body force are solved over the whole fluid domain. The same approach of IBM was applied by Feng and Michaelides in the lattice Boltzmann method (LBM) [3] to simulate the rigid particle motion. It combines the desirable features of the lattice Boltzmann [4] and immersed boundary methods. The lattice grid is used for the fluid flow field and the boundary points are to represent rigid particle surface. Any small deformation results in a large force that is distributed into the fluid field. Hence, the lattice Boltzmann equation with body force is solved to obtain the fluid velocity. The deformation is calculated by comparing the boundary point and the reference point that undergoes rigid body motion with particles. The restoration force due to the deformation is modeled by a linear spring relation, where the spring constant in the calculation should make the particle surface stiff enough for the displacement to be small but not large enough that would affect the convergence of the computations. There is no significant effect to the simulation results by choosing different values of the spring constant as long as the particle deformation is small. In order to avoid the need to determine this free parameter of the spring constant, a direct forcing scheme is proposed by Feng and Michaelides [5]. It presents a significant improvement over the previously introduced IBM–LBM where the forcing term was computed using a penalty method and a user-defined parameter. It allows the enforcement of the rigid body motion in a more efficient way.

In the original IBM–LBM, the single-relaxation-time collision scheme is used. Compared with the single-relaxation-time collision scheme, the multi-relaxation-time collision scheme has better computational stability. The multi-relaxation-time collision scheme [6] is constructed in the moment space based on the generalized lattice Boltzmann equation according to the work of d'Humières [7]. This model has a maximum number of adjustable parameters allowed by the freedom provided by a given discrete velocity set. These adjustable parameters are used to optimize the properties of the model through a systematical analysis of the generalized hydrodynamics of the model. Compared with the single-relaxation-time collision scheme, through the linearized analysis [6], the multi-relaxation-time collision scheme gives the same results to the second order of accuracy for various flows. They are equivalent in the long wave length limit for macroscopic variables and the difference is a high order effect. When flow problems possess mathematical singularities, which will affect numerical solutions in high wave numbers, the multi-relaxation-time scheme has better computational stability due to the separation of the relaxations of the various kinetic models, especially near the geometric singularity. In attempting to obtain solutions for the high Reynolds number flow using single-relaxation-time scheme, the pressure and velocity fields often exhibit spatial oscillations in regions of large gradients such as stagnation point and sharp convex corners. For convection-dominated problems, these oscillations will propagate over a large scale. However, the multi-relaxation-time scheme can efficiently suppress the oscillation associated with the high-frequency pressure wave [8]. In order to improve the stability of IBM–LBM, in this paper, the multi-relaxation-time scheme is used to replace the single-relaxation-time scheme in the original IBM–LBM by Feng and Michaelides [3].

In the application of IBM–LBM, one limitation to the numerical efficiency is that it is constrained by a special uniform lattice. The challenge for the uniform grid is to offer high resolution near a solid body and to place the outer boundary far away from the body. So it is desirable to divide the computational domain into a number of grid blocks so that within each block, uniform lattice spacing can be used [9]. For the grid block near a solid body, the lattice spacing is very small, while near the outer boundary, the lattice spacing could be very large. The multi-block approach has also been actively employed in the Navier–Stokes solvers with Cartesian coordinates, such as the work of Wan et al. [10]. Both the work of Wan et al. [10] and the present work use the idea of fictitious overlapping sub-domains. The main difference is that in Wan et al.'s scheme, the blocks with different grid sizes have different fictitious overlapping zones, while in the present method, only the coarse block has fictitious overlapping zone. The blocks are connected through the interface. An accurate interface treatment between neighboring blocks is derived by Yu et al. [9] to satisfy the continuity of mass, momentum and stresses across the interface. This technique can also be applied to the lattice Boltzmann method with multi-relaxation-time scheme [8]. In this paper, we used this multi-block method with multi-relaxation-time collision scheme to improve the efficiency and accuracy of the IBM–LBM.

The accuracy of the multi-block scheme for the lattice Boltzmann method with multi-relaxation-time collision scheme is studied by the Taylor problem and its application is validated by the driven cavity flow. Then,

the multi-block method with multi-relaxation-time collision scheme is used for IBM–LBM. It is validated by the steady and unsteady flows past a cylinder and an airfoil. Good results are obtained.

2. Immersed boundary–lattice Boltzmann method

2.1. Immersed boundary method

In the concept of immersed boundary method, the body force term is used in the fluid solver to implement constraints acting on the fluid such as to mimic the presence of rigid boundary in appropriate regions of the flow field. They should comprise sufficiently large number of grid points in order to represent the geometry of the physical boundary.

For the two-dimensional domain Ω containing an immersed boundary in the form of closed curve Γ, the configuration of the curve is given in the parametric form of $X(s,t), 0 \leqslant s \leqslant L_b$, $X(0,t) = X(L_b,t)$, where s tracks a material point of the immersed boundary. The equations of motion of the system are:

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

$$\nabla\cdot\boldsymbol{u} = 0 \tag{2}$$

$$\boldsymbol{f}(\boldsymbol{x},t) = \int_0^{L_b} \boldsymbol{F}(s,t)\delta(\boldsymbol{x} - \boldsymbol{X}(s,t))\,\mathrm{d}s \tag{3}$$

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

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

Here $\boldsymbol{x}$, $\boldsymbol{u}$, p and $\boldsymbol{f}$ are the fixed Eulerian grid, fluid velocity, fluid pressure and the force density with respect to $\mathrm{d}\boldsymbol{x} = \mathrm{d}x\,\mathrm{d}y$ acting on the fluid, respectively. $\boldsymbol{X}$ and $\boldsymbol{F}$ are the Lagrangian points and boundary force density with respect to $\mathrm{d}s$. Eqs. (1) and (2) are the Navier–Stokes equations for a viscous incompressible fluid flow. Eq. (3) distributes the restoring force from Lagrangian points on the boundary to Eulerian points in the flow domain. So, the integral in Eq. (3) is over the boundary domain formed by the Lagrangian points. Eq. (4) interpolates the velocity from Eulerian points in the flow domain to Lagrangian points on the boundary. So, the integral in Eq. (4) is over the flow domain formed by the Eulerian points. Eq. (5) states that the boundary force on the particular segment at time t is determined by the boundary configuration at time t, where the function $\boldsymbol{S}$ satisfies a generalized Hooke's law which will be shown in Eq. (9).

2.2. Lattice Boltzmann equation with single-relaxation-time collision model

For the IBM–LBM proposed by Feng and Michaelides [3], Eqs. (1) and (2) are represented in the lattice Boltzmann frame with single-relaxation-time collision scheme as:

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

where τ is the single-relaxation-time; f_α is the distribution function; f_α^{eq} is its corresponding equilibrium state; δt is the time step; $\boldsymbol{e}_\alpha$ is the particle velocity and $\boldsymbol{f}(\boldsymbol{x},t)$ is the body force in the flow field. w_α are coefficients that depend on the selected particle velocity model. Take the nine-speed particle velocity model as an example. $w_0 = 4/9$, $w_1 = w_2 = w_3 = w_4 = 1/9$ and $w_5 = w_6 = w_7 = w_8 = 1/36$. There are several ways to include the contribution of an external force into the lattice Boltzmann equation. The way used in the above equation is that an external force is introduced by adding a term to the collision function.

2.3. Evaluation of force density at the boundary point

In order to model the interaction of fluid and rigid body, usually the penalty technique as applied by Höfler and Schwarzer [11] is used. Firstly, introduce a number of m reference positions $\boldsymbol{r}_j$, $j = 1,\ldots,m$ distributed over

the rigid body with the initial boundary mesh width Δs. The $\boldsymbol{r}_j$ are vectors related to the center of mass of the rigid body. The associated spatial coordinates $\boldsymbol{X}_j^r$ change only due to the movement and rotation of the rigid body,

$$\boldsymbol{X}_j^r(t) = \boldsymbol{X}(t) + \boldsymbol{O}(t) \cdot \boldsymbol{r}_j \tag{7}$$

where $\boldsymbol{O}(t)$ describes the instantaneous orientation and $\boldsymbol{X}(t)$ is the position of the rigid body.

Associated with each reference position is a tracer $\boldsymbol{X}_j^t$, $j = 1,\ldots,m$ which tracks the motion of the fluid,

$$\dot{\boldsymbol{X}}_j^t(t) = \boldsymbol{u}(\boldsymbol{X}_j^t) \tag{8}$$

where the dot indicates the time derivative along the trajectories. The tracers are passively convected with the fluid. Whenever between the tracer and reference position, there occurs a non-vanishing difference in position $\xi_j = \boldsymbol{X}_j^t(t) - \boldsymbol{X}_j^r(t)$, an additive contribution $\boldsymbol{F}_j$ to the force density $\boldsymbol{f}_j$ in the fluid is generated that tends to drive the fluid and thus trace back to the reference position and to diminish their relative velocity. This restoration force, which is due to the displacement between the reference point and the boundary point, is modeled by a linear spring relation:

$$\boldsymbol{F}_j = -k\xi_j \tag{9}$$

where k is the spring constant. The actual value of the spring constant should make the rigid body stiff enough for the displacement to be small but not large enough to affect the convergence of computations. There is no significant effect to the simulation results by choosing different values of the spring constant as long as the deformation is small.

To avoid using a free parameter in Eq. (9) in the calculation, a direct forcing scheme for IBM–LBM is proposed by Feng and Michaelides [5] recently. As mentioned above, a set of Lagrangian boundary points is used to describe the boundary. Eq. (1) is also valid at the Lagrangian boundary points. Hence the force density at the boundary points can be written as:

$$\boldsymbol{f}(\boldsymbol{x} = \boldsymbol{X}(s,t),t) = \rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}\right) + \nabla p - \mu \Delta \boldsymbol{u} \tag{10}$$

If we assume that the velocity and pressure field at the time step $t = t_n$ are known, then we have an explicit scheme to determine the force term at the Lagrangian boundary points at time $t = t_{n+1}$, which is

$$\boldsymbol{f}_i^{n+1} = \rho\left(\frac{\boldsymbol{u}_i^{n+1} - \boldsymbol{u}_i^n}{\Delta t} + \boldsymbol{u}_i^n \cdot \nabla \boldsymbol{u}_i^n\right) + \nabla p_i^n - \mu \Delta \boldsymbol{u}_i^n \tag{11}$$

where Δt is the time step.

In order to impose the boundary condition at $t = t_{n+1}$, the velocity at the immersed Lagrangian boundary points is equal to the velocity of the boundary at the same point, $\boldsymbol{U}_i^{n+1}$. So the force density at these Lagrangian boundary points is given by

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

2.4. Calculation of body force at the fluid point

The calculated force density using Eq. (12) is at a Lagrangian boundary point and we have to spread it into the neighboring Eulerian nodes using the delta function. According to the work of Feng and Michaelides [5], the force distributed to the Eulerian node in the flow domain can be calculated by

$$\boldsymbol{f}(\boldsymbol{x},t) = \sum_{j=1}^{m-1} \boldsymbol{f}(\boldsymbol{X}(s_j,t),t) D_j(\boldsymbol{x} - \boldsymbol{X}(s_j,t)) \Delta s h \tag{13}$$

where D_j is a continuous kernel distribution function that approximates the delta function, Δs is the length of the small boundary element and h is the mesh spacing of Eulerian grid. The choice of the kernel distribution

function has to meet certain criteria [12]. For the two-dimensional simulations, the following equations are used:

$$\delta(r) = \begin{cases} \frac{1}{8h}\left(3 - 2|r|/h + \sqrt{1 + 4|r|/h - 4(|r|/h)^2}\right), & |r| \leqslant h \\ \frac{1}{8h}\left(5 - 2|r|/h - \sqrt{-7 + 12|r|/h - 4(|r|/h)^2}\right), & h \leqslant |r| \leqslant 2h \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

$$D_j(\boldsymbol{x} - \boldsymbol{x}_j) = \delta(x - x_j)\delta(y - y_j) \tag{15}$$

The same continuous kernel distribution function has also to be used for Eq. (4). The process for IBM–LBM can be illustrated as follows:

(1) Compute the force density at the boundary points using Eq. (12) and spread them into the Eulerian points using Eq. (13).
(2) Solve the lattice Boltzmann equation (6) with the force term $\boldsymbol{f}^{n+1}(\boldsymbol{x})$ to update the velocity field.
(3) Interpolate the new velocity from the lattice to the boundary points using

$$\boldsymbol{u}^{n+1}(s) = \sum_x \boldsymbol{u}^{n+1}(\boldsymbol{x}) D(\boldsymbol{x} - \boldsymbol{X}^n(s))h^2 \tag{16}$$

(4) Go back to (1) till the convergence criterion is satisfied.

3. Multi-relaxation-time collision scheme

In order to make the computation more stable, the multi-relaxation-time collision scheme is used to replace the single-relaxation-time collision scheme in the original IBM–LBM [3].

In this paper, the nine-velocity model is used, which is given as:

$$\boldsymbol{e}_\alpha = \begin{cases} (0,0), & \alpha = 0 \\ (\cos[(\alpha-1)\pi/2], \sin[(\alpha-1)\pi/2]), & \alpha = 1\text{–}4 \\ \sqrt{2}(\cos[(2\alpha-9)\pi/4], \sin[(2\alpha-9)\pi/4]), & \alpha = 5\text{–}8 \end{cases} \tag{17}$$

Given a set of 9 discrete velocities with corresponding distribution functions, one can construct a space based on the (velocity) moments of $\{f_\alpha\}$. The following moments are chosen [6]:

$$\begin{aligned} |m_1\rangle &= (1,1,1,1,1,1,1,1,1)^{\mathrm{T}} \\ |m_2\rangle &= (-4,-1,-1,-1,-1,2,2,2,2)^{\mathrm{T}} \\ |m_3\rangle &= (4,2,2,2,2,1,1,1,1)^{\mathrm{T}} \\ |m_4\rangle &= (0,1,0,-1,0,1,-1,-1,1)^{\mathrm{T}} \\ |m_5\rangle &= (0,-2,0,2,0,1,-1,-1,1)^{\mathrm{T}} \\ |m_6\rangle &= (0,0,1,0,-1,1,1,-1,-1)^{\mathrm{T}} \\ |m_7\rangle &= (0,0,-2,0,2,1,1,-1,-1)^{\mathrm{T}} \\ |m_8\rangle &= (0,1,-1,1,-1,0,0,0,0)^{\mathrm{T}} \\ |m_9\rangle &= (0,0,0,0,0,1,-1,1,-1)^{\mathrm{T}} \end{aligned} \tag{18}$$

Define a new set of variables.

$|R\rangle = (\rho, e, \varepsilon, j_x, q_x, j_y, q_y, p_{xx}, p_{xy})^{\mathrm{T}}$, which are related to the distributions as follows:

$$|R\rangle = \begin{pmatrix} \rho \\ e \\ \varepsilon \\ u \\ q_x \\ v \\ q_y \\ p_{xx} \\ p_{xy} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \end{pmatrix} = M|F\rangle \tag{19}$$

These 9 moments are separated into two groups: (ρ, u, v) are the conserved moments and they are locally conserved in the collision process; $(e, \varepsilon, q_x, q_y, p_{xx}, p_{xy})$ are the non-conserved moments and they relax linearly towards their equilibrium values that are functions of the conserved quantities. The relaxation equations for the non-conserved moments are prescribed as follows:

$$\begin{aligned} \tilde{e} &= e - s_2[e - e^{\mathrm{eq}}] \\ \tilde{\varepsilon} &= \varepsilon - s_3[\varepsilon - \varepsilon^{\mathrm{eq}}] \\ \tilde{q}_x &= q_x - s_5\lfloor q_x - q_x^{\mathrm{eq}} \rfloor \\ \tilde{q}_y &= q_y - s_7\lfloor q_y - q_y^{\mathrm{eq}} \rfloor \\ \tilde{p}_{xx} &= p_{xx} - s_8\lfloor p_{xx} - p_{xx}^{\mathrm{eq}} \rfloor \\ \tilde{p}_{xy} &= p_{xy} - s_9\lfloor p_{xy} - p_{xy}^{\mathrm{eq}} \rfloor \end{aligned} \tag{20}$$

where $\sim$ denotes post-collision values. The equilibrium values of the non-conserved moments are chosen to be:

$$\begin{aligned} e^{\mathrm{eq}} &= -2\rho + 3(u^2 + v^2) \\ \varepsilon^{\mathrm{eq}} &= \rho - 3(u^2 + v^2) \\ q_x^{\mathrm{eq}} &= -u \\ q_y^{\mathrm{eq}} &= -v \\ p_{xx}^{\mathrm{eq}} &= u^2 - v^2 \\ p_{xy}^{\mathrm{eq}} &= uv \end{aligned} \tag{21}$$

The evolution equation of the LBM with multi-relaxation-time collision scheme becomes

$$|f(\boldsymbol{x} + \boldsymbol{e}_\alpha \delta t, t + \delta t)\rangle - |f(x,t)\rangle = -M^{-1}S[|R(\boldsymbol{x},t)\rangle - |R^{\mathrm{eq}}(\boldsymbol{x},t)\rangle] \tag{22}$$

where

$$S \equiv \mathrm{diag}(s_1, s_2, \ldots, s_9)$$

There is no need to do the matrix inverse since $(M \cdot M^{\mathrm{T}})$ is a diagonal matrix with diagonal elements $\langle m_\beta | m_\beta \rangle$.

The kinematic viscosity v and the bulk viscosity ζ of this model are:

$$v = \frac{1}{3}\left(\frac{1}{s_8} - \frac{1}{2}\right) = \frac{1}{3}\left(\frac{1}{s_9} - \frac{1}{2}\right) \tag{23}$$

$$\zeta = \frac{1}{6}\left(\frac{1}{s_2} - \frac{1}{2}\right) \tag{24}$$

Compared to Eq. (6) which uses the single-relaxation-time collision model, the stream process is the same. The only difference is the collision process, which is replaced by the right hand side of Eq. (22).

When the LBM with multi-relaxation-time scheme is used for IBM, a force term has to be added. To include the external force, one can simply add it to the momentum by $u = \sum f \boldsymbol{e} + \boldsymbol{f}\ \delta t$. It is understood that, in order to conserve the mass up to the second order in the Chapman–Enskog expansion, the net effect of the

forcing term is that the resultant momentum is equal to $\boldsymbol{u} = \sum f\boldsymbol{e} + \frac{1}{2}\boldsymbol{f}\ \delta t$. Therefore it is preferable to execute the forcing term in two steps, adding one-half of the forcing before relaxation and another-half after relaxation. This can be concisely illustrated as follows:

Step 1: Advection of $\{f_\alpha\}$.
Step 2: Compute the macroscopic variables with $\boldsymbol{u} = \sum f\boldsymbol{e} + \frac{1}{2}\boldsymbol{f}\ \delta t$.
Step 3: Relaxations of the moments (collision) with $\boldsymbol{u}' = \boldsymbol{u} + \frac{1}{2}\boldsymbol{f}\ \delta t$.

4. Multi-block method in the lattice Boltzmann model

When the LBM with multi-relaxation-time scheme is used for IBM, in order to offer high resolution near a solid body and to place the outer boundary far away from the body, the multi-block method is used. An accurate interface treatment between neighboring blocks is derived by Yu et al. [9] to satisfy the continuity of mass, momentum and stresses across the interface.

4.1. Multi-block method for the single-relaxation-time scheme

To illustrate the basic idea, a two-block (a coarse block and a fine block) system as shown in Fig. 1 is considered in the derivation for the interfacial information exchange. The ratio of the lattice space between the coarse and fine blocks is

$$n = \delta x_c/\delta x_f \tag{25}$$

For a given *Re*, in order to keep a consistent viscosity $v = (2\tau - 1)\delta x/6$, the relation between relaxation times τ_f on the fine block and on the coarse block must obey

$$\tau_f = \tfrac{1}{2} + n\left(\tau_c - \tfrac{1}{2}\right) \tag{26}$$

For easy analysis, the density distributions are written as

$$f_\alpha(\boldsymbol{x}, t) = f_\alpha^{eq}(\boldsymbol{x}, t) + f_\alpha^{neq}(\boldsymbol{x}, t) \tag{27}$$

where $f_\alpha^{neq}(\boldsymbol{x}, t)$ is the non-equilibrium part of the distribution function, based on which the stresses are evaluated as $\tau_{ij} = (1 - \frac{1}{2\tau})\sum_{\alpha=1}^{8} f_\alpha^{neq}(e_{\alpha i}e_{\alpha j} - \frac{1}{2}\boldsymbol{e}_\alpha \cdot \boldsymbol{e}_\alpha \delta_{ij})$ in two dimensions.

Since the velocity and density are continuous across the interface between two blocks, the equilibrium part across the interface follows:

$$f_\alpha^{eq,c}(\boldsymbol{x}, t) = f_\alpha^{eq,f}(\boldsymbol{x}, t) \tag{28}$$

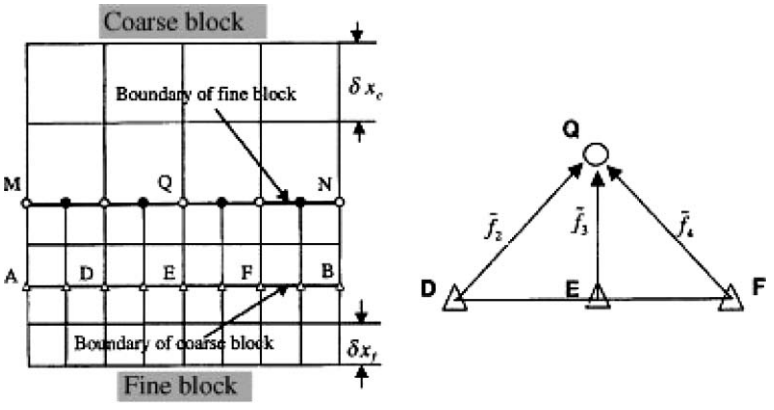

Fig. 1. Interface structure between two blocks of different lattice spacing.

To maintain the continuity for the stress across the interface, the non-equilibrium part across interface follows:

$$\left(1-\frac{1}{2\tau_c}\right)f_\alpha^{neq,c}=\left(1-\frac{1}{2\tau_f}\right)f_\alpha^{neq,f} \tag{29}$$

This can also be written as

$$f_\alpha^{neq,c}=n\frac{\tau_c}{\tau_f}f_\alpha^{neq,f} \tag{30}$$

The post-collision distribution is obtained by the single-relaxation-time scheme as

$$\tilde{f}_\alpha(\boldsymbol{x},t)=\left(1-\frac{1}{\tau}\right)f(\boldsymbol{x},t)+\frac{1}{\tau}f_\alpha^{eq}(\boldsymbol{x},t)=f_\alpha^{eq}(\boldsymbol{x},t)+\frac{\tau-1}{\tau}f_\alpha^{neq}(\boldsymbol{x},t) \tag{31}$$

For the coarse block, the post-collision distribution is written as:

$$\tilde{f}_\alpha^c=f_\alpha^{eq,c}+\frac{\tau_c-1}{\tau_c}f_\alpha^{neq,c} \tag{32}$$

When transferring the data from the fine block to the coarse block, using Eqs. (28) and (30) for Eq. (32), one obtains:

$$\tilde{f}_\alpha^c=f_\alpha^{eq,f}+n\frac{\tau_c-1}{\tau_f-1}(\tilde{f}_\alpha^f-f_\alpha^{eq,f}) \tag{33}$$

where the post-collision distribution for the fine block is written as:

$$\tilde{f}_\alpha^f=f_\alpha^{eq,f}+\frac{\tau_f-1}{\tau_f}f_\alpha^{neq,f} \tag{34}$$

Similarly, when transferring the data from the coarse block to the fine block, one obtains:

$$\tilde{f}_\alpha^f=f_\alpha^{eq,c}+\frac{\tau_f-1}{n(\tau_c-1)}(\tilde{f}_\alpha^c-f_\alpha^{eq,c}) \tag{35}$$

At the interface, the spatial interpolation is needed. The typical interface structure is shown in Fig. 1. The line MN is the fine block boundary, while the line AB is the coarse block boundary. The coarse block boundary is in the interior of the fine block and the fine block boundary is in the interior of the coarse block. This arrangement of the interface is convenient for the information exchange between two neighboring blocks. On the fine block boundary MN, the information on the nodes denoted by the open symbol "○" is known from the data in the coarse block. However, the information on the nodes denoted by the solid symbol "•" is unknown and there is no corresponding information on the coarse block. It can be obtained through spatial interpolation based on the information at the nodes denoted by the open symbol "○" on the line MN. To eliminate the possibility of spatial asymmetry caused by interpolations, a symmetric cubic spline fitting is used [7]. That is

$$\tilde{f}(x)=a_i+b_ix+c_ix^2+d_ix^3,\qquad x_{i-1}\leqslant x\leqslant x_i,\quad i=1,\ldots,m \tag{36}$$

where the constants (a_i,b_i,c_i,d_i) are determined by using the continuity of the nodal conditions of $\tilde{f},\tilde{f}',\tilde{f}''$ and suitable end condition (such as zero second order derivative for $\tilde{f}$).

For the lattice Boltzmann method, in order to make sure that the interface uses information at the correct time level, the temporal interpolation at the interface is also needed. Three-point Lagrangian formulation is used:

$$y(x)=\sum_{k=1}^{3}y_k\left(\prod_{\substack{j=1\\ j\neq k}}^{3}\frac{x-x_j}{x_k-x_j}\right) \tag{37}$$

4.2. Multi-block method for the multi-relaxation-time scheme

For the multi-relaxation-time collision scheme, the interface structure, spatial interpolation and temporal interpolation are the same as for the single-relaxation-time collision scheme. The expression for the data transfer on the interface has to be derived. But it is still based on the requirement of continuity of mass, momentum and stresses across the interface.

To maintain the continuity of macroscopic variables across the interface, one can obtain:

$$R^{\mathrm{eq,c}}(\boldsymbol{x},t) = R^{\mathrm{eq,f}}(\boldsymbol{x},t) \tag{38}$$

For the multi-relaxation-time scheme, stresses are determined by:

$$\tau_{xy} = \left(1-\frac{s_8}{2}\right)\left(p_{xy}-p_{xy}^{\mathrm{eq}}\right) = \left(1-\frac{s_8}{2}\right)p_{xy}^{\mathrm{neq}} \tag{39}$$

$$\tau_{xx} = \left(1-\frac{s_9}{2}\right)\left(p_{xx}-p_{xx}^{\mathrm{eq}}\right) = \left(1-\frac{s_9}{2}\right)p_{xx}^{\mathrm{neq}} \tag{40}$$

So to maintain the continuity of stresses across the interface, it requires that:

$$p_{xy}^{\mathrm{neq,c}} = n\frac{s_8^{\mathrm{f}}}{s_8^{\mathrm{c}}}p_{xy}^{\mathrm{neq,f}} \tag{41}$$

$$p_{xx}^{\mathrm{neq,c}} = n\frac{s_9^{\mathrm{f}}}{s_9^{\mathrm{c}}}p_{xx}^{\mathrm{neq,f}} \tag{42}$$

which can be written in the matrix form as

$$|R^{\mathrm{neq,c}}\rangle = \begin{pmatrix} 0 \\ e-e^{\mathrm{eq}} \\ \varepsilon-\varepsilon^{\mathrm{eq}} \\ 0 \\ q_x-q_x^{\mathrm{eq}} \\ 0 \\ q_y-q_y^{\mathrm{eq}} \\ p_{xx}-p_{xx}^{\mathrm{eq}} \\ p_{xy}-p_{xy}^{\mathrm{eq}} \end{pmatrix}^{\mathrm{c}} = T^{\mathrm{f}} \begin{pmatrix} 0 \\ e-e^{\mathrm{eq}} \\ \varepsilon-\varepsilon^{\mathrm{eq}} \\ 0 \\ q_x-q_x^{\mathrm{eq}} \\ 0 \\ q_y-q_y^{\mathrm{eq}} \\ p_{xx}-p_{xx}^{\mathrm{eq}} \\ p_{xy}-p_{xy}^{\mathrm{eq}} \end{pmatrix}^{\mathrm{f}} \tag{43}$$

where $T^{\mathrm{f}} = \mathrm{diag}(1,1,1,1,1,1,1,n\frac{s_8^{\mathrm{f}}}{s_8^{\mathrm{c}}},n\frac{s_9^{\mathrm{f}}}{s_9^{\mathrm{c}}})$.

Using Eqs. (38) and (43), to transfer the data from the fine block to the coarse block, one can obtain:

$$\tilde{f}^{\mathrm{c}} = M^{-1}R^{\mathrm{c}} - M^{-1}S^{\mathrm{c}}(R^{\mathrm{c}}-R^{\mathrm{eq,c}}) = M^{-1}R^{\mathrm{eq,c}} + M^{-1}(I-S^{\mathrm{c}})R^{\mathrm{neq,c}} = M^{-1}R^{\mathrm{eq,f}} + M^{-1}(I-S^{\mathrm{c}})T^{\mathrm{f}}R^{\mathrm{neq,f}} \tag{44}$$

which can also be written as:

$$\tilde{f}^{\mathrm{c}} = M^{-1}R^{\mathrm{eq,f}} + M^{-1}(I-S^{\mathrm{c}})T^{\mathrm{f}}(I-S^{\mathrm{f}})^{-1}[M\tilde{f}^{\mathrm{f}} - R^{\mathrm{eq,f}}] \tag{45}$$

For transferring the data from the coarse block to the fine block, one can similarly obtain

$$\tilde{f}^{\mathrm{f}} = M^{-1}R^{\mathrm{eq,c}} + M^{-1}(I-S^{\mathrm{f}})T^{\mathrm{c}}(I-S^{\mathrm{c}})^{-1}[M\tilde{f}^{\mathrm{c}} - R^{\mathrm{eq,c}}] \tag{46}$$

with $T^{\mathrm{c}} = \mathrm{diag}(1,1,1,1,1,1,1,\frac{s_8^{\mathrm{c}}}{ns_8^{\mathrm{f}}},\frac{s_9^{\mathrm{c}}}{ns_9^{\mathrm{f}}})$.

For the determination of relaxation parameters, in order to maintain the consistency of viscous coefficients, three relaxation times s_2, s_8 and s_9 need to be rescaled as

$$\frac{1}{s_i^{\mathrm{f}}} = \frac{1}{2} + n\left(\frac{1}{s_i^{\mathrm{c}}}-\frac{1}{2}\right), \quad i = 2,8,9 \tag{47}$$

For other relaxation parameters, they also need to be rescaled as Eq. (47). For the convergence of the solution, the relaxation parameters s_2, s_3, s_5 and s_7 in the finest block are required to be larger than one [8].

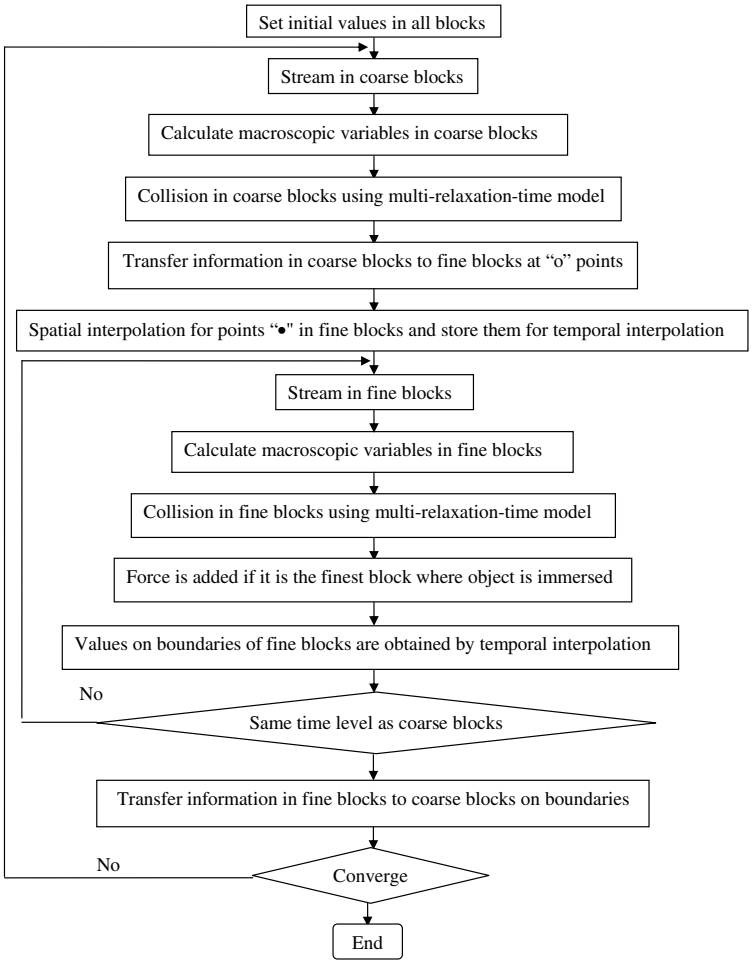

Fig. 2. Flow chart of the computational procedure.

The flow chart of the computational sequence for the multi-block immersed boundary lattice Boltzmann method with multi-relaxation-time collision model is shown in Fig. 2.

5. Results and discussions

As a precursor, the Taylor problem is solved to study the accuracy of the multi-block lattice Boltzmann method with multi-relaxation-time collision scheme. It is a problem defined in a square domain $[0,2\pi]\times[0,2\pi]$ with periodic boundary condition. The initial condition is:

$$\begin{aligned} u(x,y,0) &= -\cos(x)\sin(y) \\ v(x,y,0) &= \sin(x)\cos(y) \end{aligned} \tag{48}$$

The exact solution is given by:

$$\begin{aligned} u(x,y,t) &= -\cos(x)\sin(y)\mathrm{e}^{-(2t/Re)} \\ v(x,y,t) &= \sin(x)\cos(y)\mathrm{e}^{-(2t/Re)} \end{aligned} \tag{49}$$

Our calculations are conducted at the Reynolds number of 100. The L_2 and L_∞ errors between the numerical results and the exact solution are listed in Table 1. From Table 1, we can see that the accurate results can be obtained when more than 20 points ($N = 20$) are used in each direction.

Table 1
Errors for the Taylor problem at $Re = 100$

Mesh size	Time	L_2 error	L_∞ error
10×10	0.5	9.041×10^{-4}	1.110×10^{-3}
	1.0	8.328×10^{-4}	1.022×10^{-3}
	1.5	2.988×10^{-4}	3.658×10^{-3}
	2.0	3.358×10^{-4}	4.123×10^{-3}
20×20	0.5	6.460×10^{-5}	9.069×10^{-5}
	1.0	1.046×10^{-4}	8.447×10^{-5}
	1.5	7.270×10^{-5}	6.713×10^{-5}
	2.0	1.151×10^{-4}	7.586×10^{-5}
50×50	0.5	4.248×10^{-5}	5.972×10^{-5}
	1.0	3.848×10^{-5}	5.409×10^{-5}
	1.5	2.010×10^{-5}	2.753×10^{-5}
	2.0	4.240×10^{-5}	5.941×10^{-5}

5.1. Driven cavity flows

To validate our code using the multi-block scheme for the lattice Boltzmann method with multi-relaxation-time collision scheme, the numerical simulations of driven cavity flows are carried out.

For $Re = 100$, the multi-block is arranged as shown in Fig. 3. There are two levels, and the level ratio is 4. Fine lattice of 41×41 is used in two bottom corner regions where two smaller recirculation zones appear. Coarse grid is used in the central smooth region. The total number of grid points is 4864. With the same grid points, if the uniform grid is used, the mesh size would be 70×70. The calculation on this uniform grid using the standard LBM is also carried out for comparison.

According to the present study, as shown in Table 2, the center of the primary vortex is at $x = 0.615$, $y = 0.725$, the center of the left corner vortex is at $x = 0.034$, $y = 0.034$, and the center of the right corner vortex is at $x = 0.945$, $y = 0.058$ for the multi-block scheme. For the standard LBM, the center of the primary vortex is at $x = 0.615$, $y = 0.729$, the center of the left corner vortex is at $x = 0.033$, $y = 0.033$, and the center of the right corner vortex is at $x = 0.945$, $y = 0.059$. Both results are in good agreement with those of Ghia

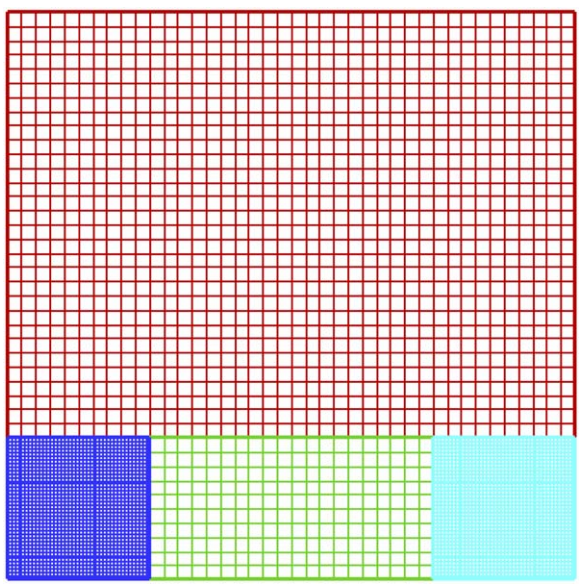

Fig. 3. Schematic diagram of multi-block arrangement for the driven cavity flow.

Table 2
Comparison of the vortex centers with those of Ghia et al. [13]

Re	Primary vortex	Lower left vortex	Lower right vortex
100			
Present	(0.615,0.725)	(0.034,0.034)	(0.945,0.058)
Ref. [13]	(0.6172,0.7344)	(0.0313,0.0391)	(0.9453,0.0625)
5000			
Present	(0.515,0.535)	(0.073,0.136)	(0.807,0.073)
Ref. [13]	(0.5117,0.5352)	(0.0703,0.1367)	(0.8086,0.0742)

et al. [13] (for $Re = 100$, the results of Ghia et al. are: $x = 0.6172$, $y = 0.7344$ for the primary vortex, $x = 0.0313$, $y = 0.0391$ for the left corner vortex, and $x = 0.9453$, $y = 0.0625$ for the right corner vortex).

The above calculations are performed on PC-2.4GHz. The CPU time needed is 74.9 s and 79.3 s for the multi-block scheme and the standard LBM, respectively. Although it will take some time to transform information between blocks for the multi-block scheme, it converges faster because fine mesh is used in the flow field where flow changes rapidly. This leads to the less CPU time needed for the multi-block scheme.

In order to obtain the same accuracy of numerical results at two bottom corners for the uniform grid, 161×161 mesh size is required for the standard LBM. This calculation is also carried out and the CPU time needed to get the converged results is 639.41 s. It takes much longer CPU time as compared to the multi-block scheme.

Fig. 4 shows the streamlines. It can be seen from this figure that, the streamlines are very smooth across the interface between different blocks. This shows that the implementation of multi-block scheme for the lattice Boltzmann method with multi-relaxation-time scheme has good property at the interface.

The same flow at $Re = 5000$ is also simulated. The streamlines are shown in Fig. 5. We can see from Fig. 5 that, there are three levels and seven blocks. Very fine grids belonging to the third level are used at four corners. According to the present study, as shown in Table 2, the center of the primary vortex is at $x = 0.515$, $y = 0.535$, the center of the left corner vortex is at $x = 0.073$, $y = 0.136$, and the center of the right corner vortex is at $x = 0.807$, $y = 0.073$. These results are in good agreement with those of Ghia et al. [13] (for $Re = 5000$, the results of Ghia et al. are: $x = 0.5117$, $y = 0.5352$ for the primary vortex, $x = 0.0703$, $y = 0.1367$ for the left corner vortex, and $x = 0.8086$, $y = 0.0742$ for the right corner vortex). Figs. 6 and 7 show velocity components along vertical and horizontal center lines, respectively. They agree well with benchmark data.

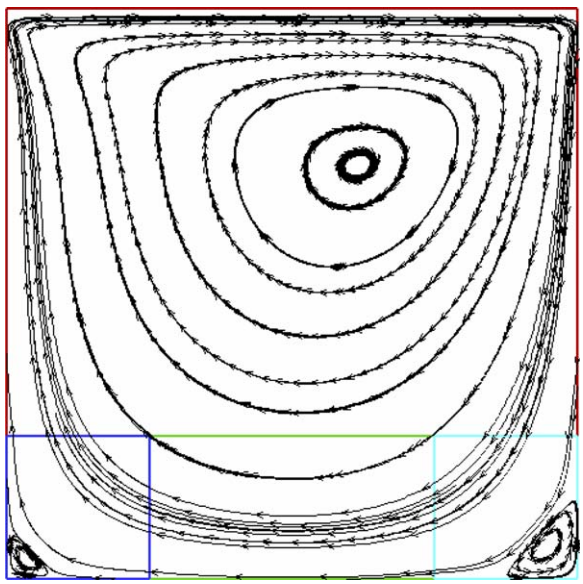

Fig. 4. Streamlines for the driven cavity flow at $Re = 100$.

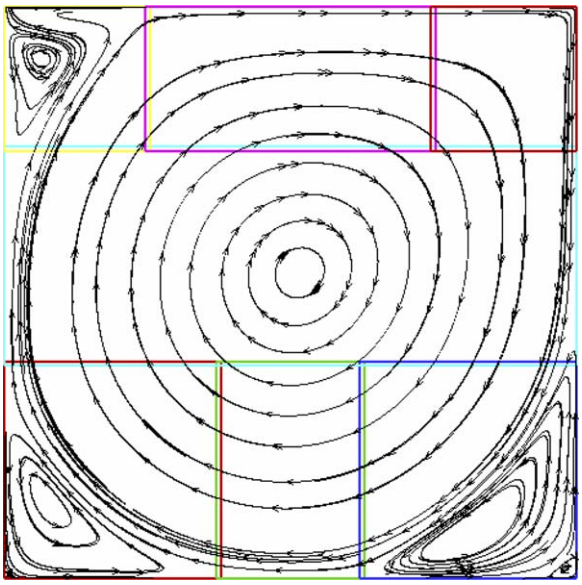

Fig. 5. Streamlines for the driven cavity flow at $Re = 5000$.

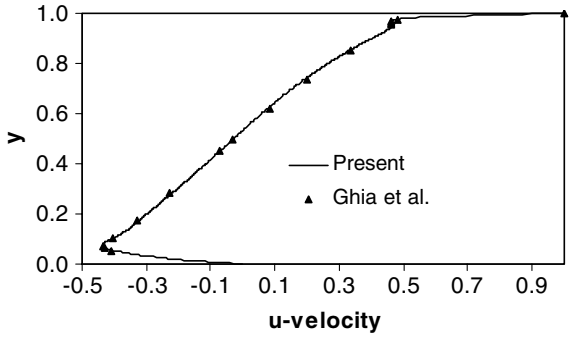

Fig. 6. Distribution of u-velocity along vertical center line for driven cavity flow at $Re = 5000$.

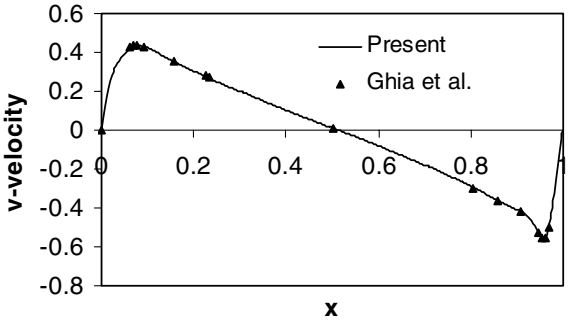

Fig. 7. Distribution of v-velocity along horizontal center line for driven cavity flow at $Re = 5000$.

5.2. Steady and unsteady flows past a circular cylinder

To validate the multi-block scheme for IBM–LBM with multi-relaxation-time collision scheme, the numerical simulation of the steady and unsteady flows past a circular cylinder is carried out.

As shown in Fig. 8, there are three levels of blocks in the calculation. Block 1 to block 4 belong to the first level; block 5 to block 8 belong to the second level and block 9 belongs to the finest level. The ratio of mesh spacing between level two and level one is 2 and the ratio between level three and level two is 4. The cylinder is immersed in block 9.

The Reynolds number in this flow is defined as

$$Re = \frac{u_\infty D}{\nu} \tag{50}$$

where D is the diameter of the cylinder.

The drag force is contributed from two sources: the shear stress and the pressure distribution along the body. For the immersed boundary method, it can be easily determined by [14]:

$$F_{\mathrm{D}} = -\int_\Omega f_1 \,\mathrm{d}\boldsymbol{x} \tag{51}$$

where f_1 is the x-component of the force density $\boldsymbol{f}$. The drag coefficient is defined by

$$C_{\mathrm{d}} = \frac{F_{\mathrm{D}}}{(1/2)\rho u_\infty^2 D} \tag{52}$$

When the body starts shedding a vortex, a lift force on the body is generated by the fluid. For the immersed boundary method, it can be easily determined by:

$$F_{\mathrm{L}} = \int_\Omega f_2 \,\mathrm{d}\boldsymbol{x} \tag{53}$$

where f_2 is the y-component of the force density $\boldsymbol{f}$.

The dimensionless lift coefficient is defined by

$$C_{\mathrm{l}} = \frac{F_{\mathrm{L}}}{(1/2)\rho u_\infty^2 D} \tag{54}$$

When the steady flow becomes unstable and the body starts shedding vortices, the frequency with which the vortices are shed from the body is non-dimensionalized as

$$St = \frac{f_{\mathrm{q}} D}{u_\infty} \tag{55}$$

where St is called the Strouhal number.

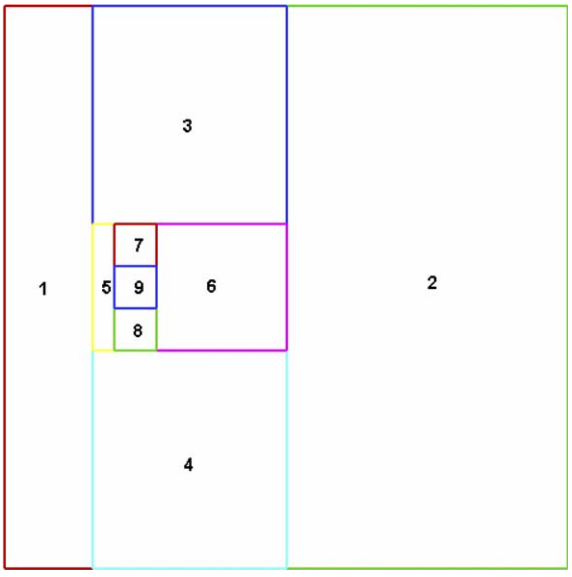

Fig. 8. Schematic diagram of multi-block arrangement for flow past a circular cylinder.

In the present calculations, the fluid density is $\rho = 1.0$ and the far field velocity is $u_\infty = 0.1$. The computation starts with the given far field velocity. At the inlet, upper, lower and downstream boundaries, the equilibrium distribution functions are used to implement the boundary conditions.

Firstly, the calculation of the steady flow at $Re = 20$ is carried out. Fig. 9 shows the streamlines when flow reaches its final steady state. A pair of stationary recirculating eddies develop behind the cylinder. The length of the recirculating region L, which is from the rearmost point of the cylinder to the end of the wake increases with the Reynolds number. The calculated non-dimensional recirculating length defined as L/r for $Re = 20$ as well as the previous computational and experimental data are listed in Table 3 [15–19]. Table 4 shows the calculated drag coefficient together with the previous computational and experimental data. From Tables 3 and 4, we can see that our numerical results compare very well with the benchmark data.

Then the calculations of the unsteady flow at $Re = 100$ and $Re = 200$ are carried out. We choose the same computational domain $\Omega = [0, 8] \times [0, 8]$ and a cylinder with diameter of $D = 0.30$. The center of the cylinder is located at (1.85, 4.0), which is the same as the one in the work of Lai and Peskin [14] using the immersed

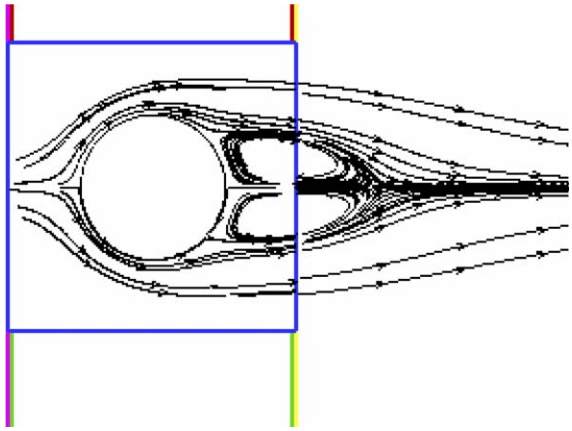

Fig. 9. Streamlines for flow past a circular cylinder at $Re = 20$.

Table 3
Comparison of the recirculating length with previous studies

Authors	$Re = 20$
Dennis and Chang [15]	1.88
Nieuwstadt and Keller [16]	1.786
Coutanceau and Bouard [17]	1.86
Fornberg [18]	1.82
He and Doolen [19]	1.842
Present	1.870

Table 4
Comparison of the drag coefficient with previous studies

Authors	$Re = 20$
Dennis and Chang [15]	2.045
Nieuwstadt and Keller [16]	2.053
Fornberg [18]	2.000
He and Doolen [19]	2.152
Present	2.148

boundary method with NS solver. Table 5 shows our numerical results together with those of Lai and Peskin [14] and previous experimental and computational results [20,21] at $Re = 100$. Table 6 displays our numerical results together with previous experiment and computational results [20–23] at $Re = 200$.

For $Re = 100$, from Table 5, we can see that the Strouhal number using present method has a better agreement with the experimental data than the first order immersed boundary method of Lai and Peskin [14]. It compares very well with the result of their second order scheme. In the paper of Saiki and Biringen [20], the computed Strouhal numbers obtained from different researchers' computations range from 0.16–0.18 at $Re = 100$. As for the drag coefficient, our present result is closer to the experimental data [21] than that of Lai and Peskin [14]. As mentioned in [14], the drag coefficient of Lai and Peskin [14] is larger than the experimental data. It seems that the present work improves the accuracy of conventional immersed boundary method. The improvement may be attributed to the use of a very fine mesh near the solid body through the multi-block technique.

For $Re = 200$, from Table 6, we can see that our present results compare very well with the previous experimental and computational results. The Strouhal number obtained from the present method is slightly higher than the experimental value. However, it is better than that obtained by the majority of the previous computational studies.

The instantaneous vorticity contours of vortex shedding at $Re = 100$ are plotted in Fig. 10. We can see clearly the Karman vortex street of the flow around a circular cylinder in these vorticity contours.

5.3. Steady flow over a NACA-0012 airfoil

For the practical applications of the multi-block IBM–LBM with multi-relaxation-time collision scheme, the steady flow over NACA-0012 airfoil at the Reynolds number of $Re = 500$ with the zero angle of attack was chosen for study. The multi-block arrangement is the same as that for the flow past a circular cylinder shown in Fig. 8. The NACA-0012 airfoil is immersed in the finest block.

Similar to the flow past a circular cylinder, the fluid density is taken as $\rho = 1.0$ and the far field velocity is $u_\infty = 0.1$. The computation starts with the given far field velocity. At the inlet, upper, lower and outlet boundaries, the equilibrium distribution functions are used to implement the boundary conditions.

The pressure coefficient and drag coefficient are, respectively, defined as

$$C_p = \frac{(P - p_\infty)}{1/2\rho u_\infty^2}, \quad C_d = \frac{F_D}{1/2\rho u_\infty^2 l} \tag{56}$$

where l is the chord. Pressure on the airfoil surface is obtained from

Table 5
Comparison of quantities at $Re = 100$ with Lai and Peskin [14] and previous computational and experiment results

Method	C_d	C_l	St
First order of Lai and Peskin [14]	1.4630	0.3290	0.144
Second order of Lai and Peskin [14]	1.4473	0.3299	0.165
Saiki and Biringen [20]	1.26	–	0.171
Clift et al. [21]	1.24	–	–
Present	1.18	0.317	0.167

Table 6
Comparison of the quantities at $Re = 200$ with previous computational and experiment results

Method	C_d	St
Saiki and Biringen [20]	1.18	0.171
Clift et al. [21]	1.16	–
Berger and Wille [22]	–	0.18–0.19
Present	1.10	0.197

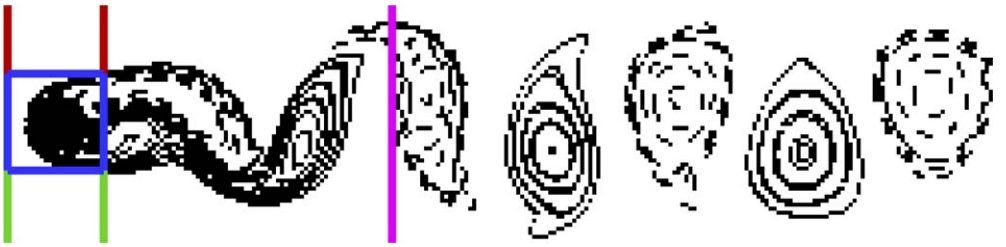

Fig. 10. Vorticity contours of the flow around a cylinder at $Re = 100$.

$$p(s) = \sum_x p(\boldsymbol{x}) D(\boldsymbol{x} - \boldsymbol{X}^n(s)) h^2 \tag{57}$$

The discrete delta function in Eq. (57) is defined in Eqs. (14) and (15).

Fig. 11 shows the pressure contours and Fig. 12 shows the distribution of pressure coefficient along the airfoil surface, respectively. They compare very well with the numerical results using a NS solver of CFL3D [24].

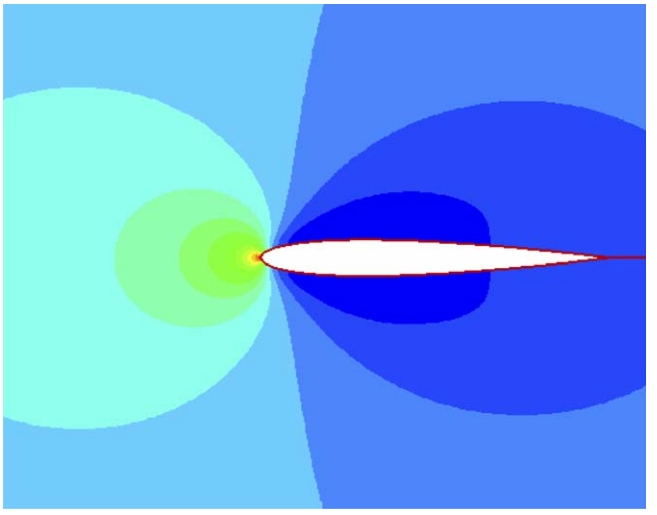

Fig. 11. Pressure contours for flow over NACA-0012 airfoil at $Re = 500$.

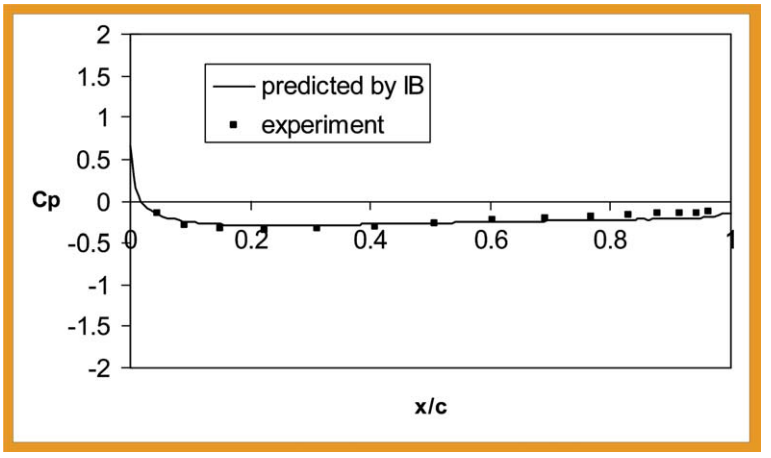

Fig. 12. Distribution of pressure coefficient along the airfoil NACA-0012 at $Re = 500$.

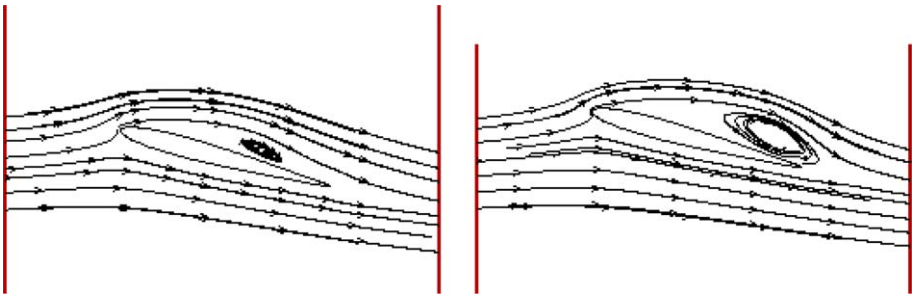

Fig. 13. Streamlines for flow over an airfoil at $t = 7.13$ (left) and $t = 11.73$ (right).

For the drag coefficient, the present value of $C_d = 0.176$ compares very well with the result reported by Lockard et al. [25] with 0.1762 obtained using a Navier–Stokes equation-based finite difference method.

5.4. Unsteady flow over an airfoil

The shape of the airfoil is mapped into a unit circle using the transformation

$$z = \frac{1}{k} + \gamma + \frac{kc^2}{1+\gamma k} \tag{58}$$

where $z = \frac{1}{k} + \gamma + \frac{kc^2}{1+\gamma k}$, $k = re^{i\theta}$ and $\gamma = \xi + i\eta$. The trailing edge of the airfoil is rounded off by defining

$$c = \left[\xi + (1-\eta^2)^{\frac{1}{2}}\right](1-\delta) \tag{59}$$

In this calculation, the shape of the airfoil is defined by

$$\xi = -0.05214, \quad \eta = 0, \quad \delta = 0.025 \quad \text{and} \quad r = \frac{\tanh(k_1) + k_4}{k_3}$$

with:

$$k_1 = 2.0, \quad k_2 = 2.8, \quad r_0 = 0.02, \quad k_3 = [\tanh(k_1) + \tanh(k_2)][1 - r_0/(r_0 - 1)],$$
$$k_4 = \tanh(k_2) - r_0[\tanh(k_1) + \tanh(k_2)]/(r_0 - 1)$$

The computation is conducted at 15° angle of attack with $Re = 1000$. The Strouhal number is defined as

$$St = \frac{fl}{u_\infty} \tag{60}$$

where l is the chord. In the present study, the calculated Strouhal number is 0.20, while the value given by Mehta and Lavan [26] is 0.2162. Fig. 13 shows the streamlines at $t = 7.13$ and $t = 11.73$. They compare very well with those of Mehta and Lavan [26] at corresponding times.

6. Conclusions

The immersed boundary–lattice Boltzmann method combines the desirable features of the lattice Boltzmann and immersed boundary methods. The simplicity of the Cartesian grid for the Eulerian variables facilitates the LBM, and the generality of the curvilinear grid for the Lagrangian variables makes it easy to model the flow problem. In order to avoid the need to determine the free parameter of the spring constant, a direct forcing scheme is used. It presents a significant improvement over the previously introduced IBM–LBM where the forcing term was computed using a penalty method and a user-defined parameter. It allows the enforcement of the rigid body motion in a more efficient way.

In this paper, we have used the multi-relaxation-time collision scheme to replace the single-relaxation-time collision scheme so as to improve the stability of the IBM–LBM for flows at high Reynolds numbers. To offer

high resolution near a solid body and to place the outer boundary far away from the body, the multi-block method is used for IBM–LBM with multi-relaxation-time collision scheme. For validation, the numerical simulations of the steady and unsteady flows past a circular cylinder and the steady and unsteady flows over an airfoil are carried out. The numerical results compare well with the benchmark data. The present technique is easily applied to the arbitrarily complicated geometry and the moving boundary problem.

References

[1] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.
[2] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
[3] Z. Feng, E. Michaelides, The immersed boundary–lattice Boltzmann method for solving fluid–particles interaction problems, J. Comput. Phys. 195 (2004) 602–628.
[4] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, Annu. Rev. Fluid Mech. 30 (1998) 329–364.
[5] Z. Feng, E. Michaelides, Proteus: a direct forcing method in the simulations of particulate flows, J. Comput. Phys. 202 (2005) 20–51.
[6] P. Lallemand, L.-S. Luo, Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, Galilean invariance, and stability, Phys. Rev. E 61 (2000) 6546–6562.
[7] D. d'Humières, in: B.D. Shizgal, D.P. Weaver (Eds.), Rarefied Gas Dynamics: Theory and Simulations, Prog. Astronaut. Aeronaut., vol. 159, AIAA, Washington, DC, 1992.
[8] D. Yu, Viscous Flow Computations with the Lattice Boltzmann Equation Method, Ph.D. Thesis, University of Florida, 2002.
[9] D. Yu, R. Mei, W. Shyy, A multi-block lattice Boltzmann method for viscous fluid flows, Int. J. Numer. Meth. Fluids 39 (2002) 99–120.
[10] D.C. Wan, B.S.V. Patnaik, G.W. Wei, Discrete singular convolution-finite subdomain method for the solution of incompressible viscous flows, J. Comput. Phys. 180 (2002) 229–255.
[11] K. Höfler, S. Schwarzer, Navier–Stokes simulation with constraint forces: finite-difference method for particle-laden flows and complex geometries, Phys. Rev. E 61 (2000) 7146–7160.
[12] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[13] U. Ghia, K.N. Ghia, C.T. Shin, High-*Re* solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387.
[14] M. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2000) 705–719.
[15] 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.
[16] F. Nieuwstadt, H.B. Keller, Viscous flow past circular cylinders, Comp. Fluids 1 (1973) 59.
[17] M. Coutanceau, R. Bouard, Experimental determination of the main features of the viscous flow in the wake of circular cylinder in uniform translation. Part I. Steady flow, J. Fluid Mech. 79 (1977) 231.
[18] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, J. Fluid Mech. 98 (1980) 819.
[19] X. He, G. Doolen, Lattice Boltzmann method on curvilinear coordinates system: flow around a circular cylinder, J. Comput. Phys. 134 (1997) 306–315.
[20] E.M. Saiki, S. Biringen, Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method, J. Comput. Phys. 123 (1996) 450.
[21] R. Clift, J.R. Grace, M.E. Weber, Bubbles, Drops and Particles, vol. 154, Academic Press, New York, 1978.
[22] E. Berger, R. Wille, Periodic flow phenomena, Annu. Rev. Fluid Mech. 4 (1972) 313–340.
[23] G.E. Karniadakis, G.S. Triantafyllou, Three-dimensional dynamics and transition to turbulence in the wake of bluff object, J. Fluid. Mech. 238 (1992) 1.
[24] T. Imamura, K. Suzuki, T. Nakamura, M. Yoshida, Flow simulation around an airfoil using lattice Boltzmann method on generalized coordinates, AIAA 2004-0244.
[25] 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.
[26] U.B. Mehta, Z. Lavan, Starting vortex, separation bubbles and stall: a numerical study of laminar unsteady flow around an airfoil, J. Fluid. Mech. 67 (1975) 227–256.