

A Boundary Condition-Implemented Immersed Boundary-Lattice Boltzmann Method and Its Application for Simulation of Flows Around a Circular Cylinder

X. Wang¹, C. Shu^{2,*}, J. Wu¹ and L. M. Yang¹

¹ *Department of Aerodynamics, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Yudao Street, Nanjing 210016, China*

² *Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore*

Received 15 October 2012; Accepted (in revised version) 23 December 2013

Available online 17 September 2014

Abstract. A boundary condition-implemented immersed boundary-lattice Boltzmann method (IB-LBM) is presented in this work. The present approach is an improvement to the conventional IB-LBM. In the conventional IB-LBM, the no-slip boundary condition is only approximately satisfied. As a result, there is flow penetration to the solid boundary. Another drawback of conventional IB-LBM is the use of Dirac delta function interpolation, which only has the first order of accuracy. In this work, the no-slip boundary condition is directly implemented, and used to correct the velocity at two adjacent mesh points from both sides of the boundary point. The velocity correction is made through the second-order polynomial interpolation rather than the first-order delta function interpolation. Obviously, the two drawbacks of conventional IB-LBM are removed in the present study. Another important contribution of this paper is to present a simple way to compute the hydrodynamic forces on the boundary from Newton's second law. To validate the proposed method, the two-dimensional vortex decaying problem and incompressible flow over a circular cylinder are simulated. As shown in the present results, the flow penetration problem is eliminated, and the obtained results compare very well with available data in the literature.

AMS subject classifications: 76M28, 76M25

Key words: Immersed boundary method, lattice Boltzmann method, velocity correction, Lagrange interpolation, no-slip boundary condition.

*Corresponding author.

URL: <http://serve.me.nus.edu.sg/shuchang/>

Email: mpeshuc@nus.edu.sg (C. Shu)

1 Introduction

Nowadays, one of the most challenging issues in computational fluid dynamics is the simulation of incompressible flows around complex geometries or moving boundaries. One approach that has been proven particularly effective in handling a variety of these problems is the immersed boundary method (IBM).

IBM was originally developed by Peskin [1] in 1970s to simulate the blood flow in a two-dimensional model of heart. It uses a fixed Cartesian mesh for the flow field, which is composed of Eulerian nodes. Meanwhile, the physical boundary immersed in the fluid is represented by a set of Lagrangian points. The basic idea of IBM is that the boundary is treated as deformable but with high stiffness. The deformation or moving of the boundary will yield a force that tends to restore the boundary to its original shape or position. The restoring force on the boundary is then distributed into the surrounding nodes (Eulerian nodes) as the body force. Subsequently, the governing equation of flow field with a body force is solved over the whole fluid domain including both inside and outside of the boundary.

Based on the work of Peskin [1], various modifications and refinements have been made. As a result, a number of variants of this approach were proposed and applied. Among the remarkable works, Goldstein et al. [2] proposed a model named virtual boundary method which permits simulations with complex geometries. They employed feedback forcing to represent the effect of solid body. The feedback force, which can be easily computed via the velocities of fluid and boundary, is added to the momentum equation to bring the fluid velocity to zero at the desired points. The main drawback of virtual boundary method is that it contains two user-defined parameters that need to be tuned according to the flow. Lai and Peskin [3] presented a second-order accurate immersed boundary method to simulate the flow past a circular cylinder. The interaction between the fluid and the immersed boundary is modeled using a well-chosen discretized approximation to the Dirac delta function. However, this method does not truly have the second-order of accuracy due to the use of first-order Dirac delta function. Tseng et al. [4] proposed a ghost-cell immersed boundary method for flow in complex geometry. This approach imposes the specified boundary condition by extrapolating the variable to a ghost node inside the body. High-order extrapolation is used to preserve the overall accuracy. Lima E Silva et al. [5] proposed a version named physical virtual model, which is based on the conservation laws, and simulated an internal channel flow and the flow around a circular cylinder. In this model, the restoring force is calculated by applying the momentum equations at the boundary points and then distributed to the surrounding Eulerian grid.

Due to its simplicity and efficiency, the lattice Boltzmann method (LBM) has been broadly used to simulate complex flows as an alternative to the Navier-Stokes (N-S) equations-based solvers. It is a particle-based numerical technique, which studies the dynamics of fictitious particles. Same as the IBM, the Cartesian mesh is usually used in the standard LBM. This common feature motivates the coupling of two methods, which

is called the immersed boundary-lattice Boltzmann method (IB-LBM). The first attempt was made by Feng and Michaelides [6,7]. They used the penalty method [6] or the direct forcing method [7] to compute the restoring force on the boundary. Applying the IB-LBM, they successfully simulated both the two-dimensional and three-dimensional rigid particle motion. However, the methods of computing the restoring force are either inaccurate or inefficient. For the penalty method [6], which is also named the Hook's law and employed in Peskin's work [1], a user-defined spring constant is needed. The selection of the spring parameter may affect the accuracy of solution. For the direct forcing method [7], which was first proposed by Fadlun et al. [8], the N-S equations are required to be solved on the boundary points, which may spoil the merits of LBM. Recently, Niu et al. [9] presented a momentum exchange-based immersed boundary-lattice Boltzmann method where a simple way was proposed for computing the force at the boundary point. In their work, based on the LBM, the forcing term is calculated by the momentum-exchange of the boundary particles. As compared to the work of Ladd [10], the force computation of Niu et al. [9] is simpler and more convenient because one does not need to care the details of the boundary position and mesh points.

In the conventional IBM, the major drawback is that the no-slip boundary condition is not enforced in its solution process. As a consequence, there is penetration of some streamlines to the solid body. Another drawback is the low order of accuracy near the boundary. That is because of the use of Dirac delta function interpolation near the boundary, which only has the first-order of accuracy. To remove these drawbacks, one has to improve the way of implementing the boundary condition. For example, Kim et al. [11] developed a new immersed boundary method for simulating flows in complex geometries. They used both mass sources/sinks and momentum forcing to satisfy the no-slip boundary condition on the immersed boundary and also to satisfy the mass conservation for the cell containing the immersed boundary. Nevertheless, the complexity is introduced in the computation. Recently, Shu et al. [12] proposed a new immersed boundary-lattice Boltzmann method based on the idea of velocity correction. In this work, they found that unsatisfying of no-slip boundary condition in IBM is due to the pre-calculated restoring force. Using the fractional step technique, they concluded that, adding a body force in the momentum equation is equivalent to making a correction in the velocity field. Therefore, there is no calculation of restoring force in this method. The fluid velocity at the boundary point is modified to be equal to the boundary velocity, which may create a series of velocity corrections at all boundary points. By distributing these velocity corrections into the surrounding Eulerian mesh points, the fluid velocity field is corrected. Since the no-slip boundary condition is satisfied, the streamline penetration is avoided. However, this method only has the first-order of accuracy due to the use of linear interpolation. At the same time, the computation of drag and lift forces by using the control volume method is quite tedious. Following the idea of Shu et al. [12], Wu et al. [13] proposed an implicit velocity correction-based immersed boundary-lattice Boltzmann method. In this work, the velocity correction is considered as unknown, which is determined in such a way that the velocity at the boundary point interpolated from the corrected velocity field

satisfies the no-slip boundary condition. However, this method still uses the Dirac delta function interpolation near the boundary, which only has the first-order of accuracy.

In this work, we present a new approach to directly enforce the no-slip boundary condition, which is inspired from the idea of Shu et al. [12]. Our approach is very simple and straightforward. First, the basic velocity field is obtained by LBM on the Cartesian mesh. After that, the velocity at the boundary point is directly used to modify the velocity at two adjacent Eulerian mesh points from both sides of the boundary through the second-order Lagrange interpolation. The current method to compute the velocity correction is different from the work of Shu et al. [12] and Wu et al. [13]. To effectively consider the body force in the lattice Boltzmann equation (LBE), we use the model proposed by Guo et al. [14]. In this model, the fluid velocity consists of two parts. One is from the density distribution function, and the other is from the body force. Through the relationship between the velocity correction and body force, we can directly calculate the drag and lift forces from the obtained velocity correction. To validate the presented method, its application to simulate the steady and unsteady two-dimensional flows past a circular cylinder is performed. The obtained numerical results basically agree well with available data in the literature. Since the no-slip boundary condition is accurately satisfied, there is no any streamline passing through the solid body.

2 Methodology

The boundary condition-implemented immersed boundary-lattice Boltzmann method (IB-LBM) is developed from the conventional immersed boundary method (IBM). In the following subsection, we will briefly describe the conventional IBM first. After that, our proposed approach, which directly enforces the no-slip boundary condition, will be introduced in detail.

2.1 Conventional immersed boundary method

In the concept of immersed boundary method, the body forcing term is used in the fluid solver to represent the influence of boundary on the surrounding flow field. Generally, the immersed boundary should comprise a large number of boundary points in order to accurately represent the geometry of the physical boundary.

For the two-dimensional domain Ω containing an immersed boundary in the form of a simple closed curve Γ , as shown in Fig. 1, the immersed boundary is given in the parametric form of $\mathbf{X}(s,t)$, $0 \leq s \leq L_b$, $\mathbf{X}(0,t) = \mathbf{X}(L_b,t)$, where s tracks a material point of the immersed boundary and L_b is the length of Γ . The governing equations for flow field can be written as:

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

$$\nabla \cdot \mathbf{u} = 0, \quad (2.1b)$$

$$\mathbf{f}(x,t) = \int_0^{L_b} \mathbf{F}(s,t)\delta(\mathbf{x}-\mathbf{X}(s,t))ds, \tag{2.1c}$$

$$\frac{\partial \mathbf{X}(s,t)}{\partial t} = \mathbf{u}(\mathbf{X}(s,t),t) = \int_{\Omega} \mathbf{u}(\mathbf{x},t)\delta(\mathbf{x}-\mathbf{X}(s,t))dx, \tag{2.1d}$$

$$\mathbf{F}(s,t) = \mathbf{S}(\mathbf{X}(s,t),t). \tag{2.1e}$$

Here, $\mathbf{x} = (x,y)$ is the Eulerian coordinate, \mathbf{u} is the fluid velocity vector having components u and v , p is the fluid pressure and $\mathbf{f}(x,t)$ is the force density acting on the fluid, respectively. $\mathbf{X}(s,t)$ and $\mathbf{F}(s,t)$ are the Lagrangian coordinate and boundary force density. Eqs. (2.1a) and (2.1b) are the familiar Navier-Stokes equations of a viscous incompressible fluid. Eq. (2.1c) distributes the restoring force from Lagrangian points on the boundary to Eulerian points in the flow domain. Eq. (2.1d) interpolates the velocity from Eulerian points in the flow domain to Lagrangian points on the boundary. They describe the interaction between the immersed boundary and the fluid flow by a Dirac delta function $\delta(\mathbf{x}-\mathbf{X}(s,t))$ interpolation. Eq. (2.1e) states that the boundary force on the segment is determined by the boundary configuration at time t , where \mathbf{S} is a function of the boundary deformation. For elastic materials, it satisfies a generalized form of Hook's law.

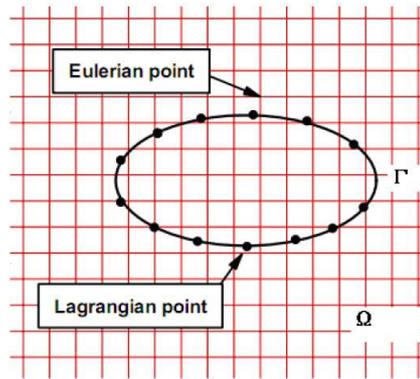


Figure 1: Flow domain Ω with an immersed boundary Γ . The solid points are used to represent the Lagrangian points and the Eulerian points are represented by the intersection points of mesh lines.

In conventional IBM, the force density \mathbf{f} is calculated in advance. As a consequence, the no-slip boundary condition is only approximately satisfied. This may induce the phenomenon of streamline penetration to the solid boundary. Based on the idea of velocity correction [12], we propose an improved version of IB-LBM, which enforces the no-slip boundary condition.

2.2 A boundary condition-implemented immersed boundary-lattice Boltzmann method

In order to represent Eqs. (2.1a) and (2.1b) in the lattice Boltzmann framework, an external forcing term should be added to the standard LBE. The modified lattice Boltzmann

equation is [6,7]

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = -\frac{1}{\tau} (f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)) + F_i \Delta t. \quad (2.2)$$

Here, f_i is the density distribution function, f_i^{eq} is its corresponding equilibrium state, τ is the single relaxation parameter, \mathbf{e}_i is the lattice velocity and \mathbf{f} is the external force density. F_i is the discrete force distribution function. Considering the two-dimensional situation, the popular D2Q9 lattice velocity model is used [15]. Here, D2 means two-dimension and Q9 denotes 9 discrete lattice velocities. The velocity set is given by

$$\mathbf{e}_i = \begin{cases} (0,0), & i=0, \\ (\cos[(i-1)\pi/2], \sin[(i-1)\pi/2]), & i=1,2,3,4, \\ (\sqrt{2}\cos[(i-5)\pi/2 + \pi/4], \sqrt{2}\sin[(i-5)\pi/2 + \pi/4]), & i=5,6,7,8. \end{cases} \quad (2.3)$$

The corresponding equilibrium distribution function is taken as

$$f_i^{eq}(\mathbf{x}, t) = \rho w_i \left[1 + \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} \right], \quad (2.4)$$

where w_i is coefficient which depends on the selected lattice velocity model. For D2Q9 model used here, $w_0 = 4/9$, $w_1 = w_2 = w_3 = w_4 = 1/9$ and $w_5 = w_6 = w_7 = w_8 = 1/36$. $c_s = 1/\sqrt{3}$ is the sound speed of the model. Finally, the relationship between the relaxation parameter and the kinematic viscosity of fluid is

$$\nu = \frac{2\tau - 1}{6} \Delta t. \quad (2.5)$$

In order to correctly recover the viscous and incompressible N-S equations involving external force, the contribution of the body force to both momentum $\rho \mathbf{u}$ and momentum flux $\rho \mathbf{u} \mathbf{u}$ should be considered [14, 16]. He et al. [16] firstly proposed a form of forcing term in LBE. Later, Guo et al. [14] presented a similar formulation with higher-order term in \mathbf{u} , which can be written as

$$f_i = \left(1 - \frac{1}{2\tau} \right) w_i \left(\frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^2} + \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^4} \cdot \mathbf{e}_i \right) \cdot \mathbf{f}. \quad (2.6)$$

With the forcing term, the momentum can be computed by [14]

$$\rho \mathbf{u} = \sum_i \mathbf{e}_i f_i + \frac{1}{2} \mathbf{f} \Delta t. \quad (2.7)$$

Furthermore, in the LBM framework, the density and pressure are calculated by

$$\rho = \sum_i f_i, \quad P = c_s^2 \rho. \quad (2.8)$$

It can be seen clearly from Eq. (2.7) that the velocity is contributed by two parts. One is contributed from the density distribution function, while the other is from the force density \mathbf{f} . Similar to the work of Shu et al. [12], we can define the velocity from the density distribution function as the intermediate velocity, and the velocity from the body force as the velocity correction. That is, the intermediate velocity \mathbf{u}^* is defined as

$$\rho \mathbf{u}^* = \sum_i \mathbf{e}_i f_i, \tag{2.9}$$

and $\mathbf{u}^c = \mathbf{u}(\mathbf{x}, t)$ is the correct velocity that satisfies the no-slip condition. Then Eq. (2.7) can be written as

$$\rho \mathbf{u}^c = \rho \mathbf{u}^* + \frac{1}{2} \mathbf{f} \Delta t. \tag{2.10}$$

So we can easily obtain the force density as

$$\mathbf{f} = 2\rho \frac{(\mathbf{u}^c - \mathbf{u}^*)}{\Delta t}. \tag{2.11}$$

To solve the unknown force density \mathbf{f} , the procedure of current method is depicted as follows. At first, we use Eq. (2.2) to obtain the density distribution function and thereby compute the macroscopic variables using Eqs. (2.8) and (2.9) at Eulerian points. Note that the obtained flow field may not satisfy the boundary condition. On the other hand, as we know, the no-slip boundary condition is necessary for the viscous fluid flow. So, in the second step, we need to implement the physical boundary condition. In the computational procedure of immersed boundary method, for every evolution time step, if the velocity of fluid at the boundary point is equal to the velocity of boundary at the same position, the no-slip boundary condition is strictly guaranteed. Inspired by this idea, we directly use the wall velocity at the boundary point to correct the velocity at two adjacent Eulerian points on both sides of the boundary point by the second-order Lagrange interpolation.

As shown in Fig. 2, it is supposed that the intersection point p between immersed boundary and Eulerian mesh line only affects the velocity correction at two mesh points

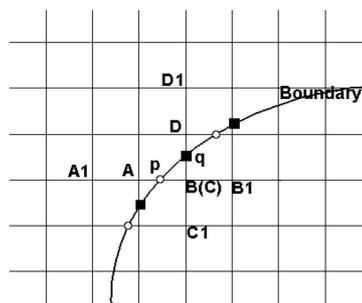


Figure 2: Cartesian mesh lines, immersed boundary and their intersection points. "o" represents the intersection points of solid boundary with horizontal mesh lines, "■" represents the intersection points of solid boundary with vertical mesh lines.

A and B (horizontal mesh line), and the intersection point q only affects the velocity correction at two mesh points $B(C)$ and D (vertical mesh line). Like the conventional IBM, we limit the effect of intersection point in a local region of the flow domain. Since points p and q are on the boundary, their velocity is given by the no-slip boundary condition (u_p, v_p) and (u_q, v_q) .

Consider the horizontal mesh line first. To satisfy the boundary condition at point p , we have to correct the velocity at its left and right adjacent Eulerian points A and B . With the known velocities at the points $A1$, $B1$ and p , we can easily get the corrected velocity at point A by the second-order Lagrange interpolation. The velocity at point A can be calculated by

$$u_A^c = \frac{(x_A - x_{A1})(x_A - x_{B1})}{(x_p - x_{A1})(x_p - x_{B1})} u_p + \frac{(x_A - x_p)(x_A - x_{B1})}{(x_{A1} - x_p)(x_{A1} - x_{B1})} u_{A1} + \frac{(x_A - x_{A1})(x_A - x_p)}{(x_{B1} - x_{A1})(x_{B1} - x_p)} u_{B1}, \quad (2.12a)$$

$$v_A^c = \frac{(x_A - x_{A1})(x_A - x_{B1})}{(x_p - x_{A1})(x_p - x_{B1})} v_p + \frac{(x_A - x_p)(x_A - x_{B1})}{(x_{A1} - x_p)(x_{A1} - x_{B1})} v_{A1} + \frac{(x_A - x_{A1})(x_A - x_p)}{(x_{B1} - x_{A1})(x_{B1} - x_p)} v_{B1}. \quad (2.12b)$$

Suppose that the mesh spacing is Δ_x , that is, $A_1A = AB = BB_1 = \Delta_x$, and the distance between points A and p is $Ap = \Delta_1$, and the distance between points B and p is $Bp = \Delta_2$, then the above equations can be simplified as

$$u_A^c = \frac{2\Delta_x^2}{(\Delta_x + \Delta_1)(\Delta_x + \Delta_2)} u_p + \frac{2\Delta_1}{3(\Delta_x + \Delta_1)} u_{A1} - \frac{\Delta_1}{3(\Delta_x + \Delta_2)} u_{B1}, \quad (2.13a)$$

$$v_A^c = \frac{2\Delta_x^2}{(\Delta_x + \Delta_1)(\Delta_x + \Delta_2)} v_p + \frac{2\Delta_1}{3(\Delta_x + \Delta_1)} v_{A1} - \frac{\Delta_1}{3(\Delta_x + \Delta_2)} v_{B1}. \quad (2.13b)$$

Similarly, the velocity at B can be interpolated by the velocity at points $A1$, $B1$ and p

$$u_B^c = \frac{2\Delta_x^2}{(\Delta_x + \Delta_1)(\Delta_x + \Delta_2)} u_p - \frac{\Delta_2}{3(\Delta_x + \Delta_1)} u_{A1} + \frac{2\Delta_2}{3(\Delta_x + \Delta_2)} u_{B1}, \quad (2.14a)$$

$$v_B^c = \frac{2\Delta_x^2}{(\Delta_x + \Delta_1)(\Delta_x + \Delta_2)} v_p - \frac{\Delta_2}{3(\Delta_x + \Delta_1)} v_{A1} + \frac{2\Delta_2}{3(\Delta_x + \Delta_2)} v_{B1}. \quad (2.14b)$$

For the intersection points of immersed boundary with vertical mesh lines, we have to correct the velocity at their upper (like point D in Fig. 2) and lower (like point C in Fig. 2) adjacent Eulerian points. The velocity at point C can be computed by

$$u_C^c = \frac{(y_C - y_{C1})(y_C - y_{D1})}{(y_q - y_{C1})(y_q - y_{D1})} u_q + \frac{(y_C - y_q)(y_C - y_{D1})}{(y_{C1} - y_q)(y_{C1} - y_{D1})} u_{C1} + \frac{(y_C - y_{C1})(y_C - y_q)}{(y_{D1} - y_{C1})(y_{D1} - y_q)} u_{D1}, \quad (2.15a)$$

$$v_C^c = \frac{(y_C - y_{C1})(y_C - y_{D1})}{(y_q - y_{C1})(y_q - y_{D1})} v_q + \frac{(y_C - y_q)(y_C - y_{D1})}{(y_{C1} - y_q)(y_{C1} - y_{D1})} v_{C1} + \frac{(y_C - y_{C1})(y_C - y_q)}{(y_{D1} - y_{C1})(y_{D1} - y_q)} v_{D1}. \quad (2.15b)$$

Similarly, suppose that $D_1D = DC = CC_1 = \Delta_y$ and $Dq = \Delta_3$, $Cq = \Delta_4$, then the velocities at

points C and D can be calculated by

$$u_c^c = \frac{2\Delta_y^2}{(\Delta_y + \Delta_3)(\Delta_y + \Delta_4)} u_q - \frac{\Delta_4}{3(\Delta_y + \Delta_3)} u_{D1} + \frac{2\Delta_4}{3(\Delta_y + \Delta_4)} u_{C1}, \quad (2.16a)$$

$$v_c^c = \frac{2\Delta_y^2}{(\Delta_y + \Delta_3)(\Delta_y + \Delta_4)} v_q - \frac{\Delta_4}{3(\Delta_y + \Delta_3)} v_{D1} + \frac{2\Delta_4}{3(\Delta_y + \Delta_4)} v_{C1}, \quad (2.16b)$$

$$u_D^c = \frac{2\Delta_y^2}{(\Delta_y + \Delta_3)(\Delta_y + \Delta_4)} u_q + \frac{2\Delta_3}{3(\Delta_y + \Delta_3)} u_{D1} - \frac{\Delta_3}{3(\Delta_y + \Delta_4)} u_{C1}, \quad (2.16c)$$

$$v_D^c = \frac{2\Delta_y^2}{(\Delta_y + \Delta_3)(\Delta_y + \Delta_4)} v_q + \frac{2\Delta_3}{3(\Delta_y + \Delta_3)} v_{D1} - \frac{\Delta_3}{3(\Delta_y + \Delta_4)} v_{C1}. \quad (2.16d)$$

If the velocity at an Eulerian point is modified twice (like point $B(C)$ in Fig. 2), we then take their mean value

$$u_{B(C)}^c = \frac{1}{2}(u_B^c + u_C^c), \quad (2.17a)$$

$$v_{B(C)}^c = \frac{1}{2}(v_B^c + v_C^c). \quad (2.17b)$$

After obtaining the corrected velocity by Eqs. (2.13a)-(2.14b) and (2.16a)-(2.17b), the force density at the fluid point can be simply calculated by Eq. (2.11).

Next, we solve the governing equations at all Eulerian points again with the force density. We continue this process until the converged solution is obtained. To summarize, the basic solution procedure of present approach can be outlined below,

1. Set initial flow field and input data.
2. Use Eq. (2.2) to obtain the density distribution function at time $t = t_n$ and compute the macroscopic flow variables using Eqs. (2.8) and (2.9).
3. Use Eqs. (2.13a)-(2.14b), (2.16a)-(2.17b) to compute the corrected velocity at adjacent Eulerian points around the boundary.
4. Obtain the force density using Eq. (2.11).
5. Compute the equilibrium distribution function using Eq. (2.4).
6. Repeat Steps 2 to 5 until convergence criterion is reached.

It should be indicated that in the present approach, the no-slip boundary condition is directly implemented while it is only approximately satisfied in the conventional IBM. Meanwhile, no Dirac delta function is used, which only has the first-order of accuracy. In this work, the velocity at the boundary point is directly used to modify the velocity in the vicinity of the boundary point by the second-order Lagrange interpolation. This idea was first used in the work of Shu et al. [12], where the velocity correction is made by linear interpolation with only the first-order of accuracy. Another advantage of present method is the simple calculation of the force. We can easily compute the force density by Eq. (2.11).

3 Numerical results and discussion

3.1 Numerical test of overall accuracy

In this work, LBM is used to obtain the basic flow field. As we all know, LBM has the second-order of accuracy in time and space. The second-order Lagrange interpolation is used to correct the velocity in the vicinity of the boundary point. Although it is only used around the boundary, we have to consider its effect on the global accuracy of solution in the whole region. On the other hand, we have to indicate that a theoretical analysis of overall accuracy for present method is very difficult since the process is complicated. So in the following, we take the decaying vortex problem to numerically examine the overall accuracy. In the work of Chen et al. [17], it is used to access the numerical accuracy of their model. The solution of the problem is expressed as

$$u(x,y,t) = -U_0 \cos\left(\frac{x\pi}{L}\right) \sin\left(\frac{y\pi}{L}\right) \exp\left[-\left(\frac{2\pi^2 U_0 t}{ReL}\right)\right], \quad (3.1a)$$

$$v(x,y,t) = U_0 \sin\left(\frac{x\pi}{L}\right) \cos\left(\frac{y\pi}{L}\right) \exp\left[-\left(\frac{2\pi^2 U_0 t}{ReL}\right)\right], \quad (3.1b)$$

$$\rho(x,y,t) = \rho_0 - \frac{\rho_0 U_0^2}{4C_s^2} \left[\cos\left(\frac{2x\pi}{L}\right) + \sin\left(\frac{2y\pi}{L}\right) \right] \exp\left[-\left(\frac{4\pi^2 U_0 t}{ReL}\right)\right]. \quad (3.1c)$$

Similar to the work of Chen et al. [17], the computational domain is set as $[-L, L] \times [-L, L]$, where $L=1$ and the periodic boundary condition is applied. The single relaxation parameter is chosen as $\tau=0.65$, the Reynolds number is $Re=10$. A circular cylinder with a radius of 0.5 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 immersed body.

As the simulation is time-dependent, all the computations are up to time $U_0 t/L = 1$. Six different uniform grids $[(N+1) \times (N+1), N = 20, 40, 80, 160]$ are used in the test. The numerical error of u is measured using L_2 norm, which is defined as

$$Err = \sqrt{\frac{\sum_{N \times N} ((u - u^{exact})/U_0)^2}{N \times N}}. \quad (3.2)$$

Here u and u^{exact} represent the numerical velocity and the exact velocity. The order of accuracy can be computed by taking logarithmic for the ratio of two successive errors as

$$\text{Order of accuracy} = \log_2 \frac{Err_N}{Err_{(2N)}}, \quad (3.3)$$

where Err_N denotes the error using the mesh size of $(N+1) \times (N+1)$.

The numerical error of the velocity component u versus mesh spacing in the log scale is plotted in Fig. 3. Table 1 shows the numerical error for the decaying vortex problem given by present method. As shown in Table 1 and Fig. 3, the present method approximately has the second-order of accuracy.

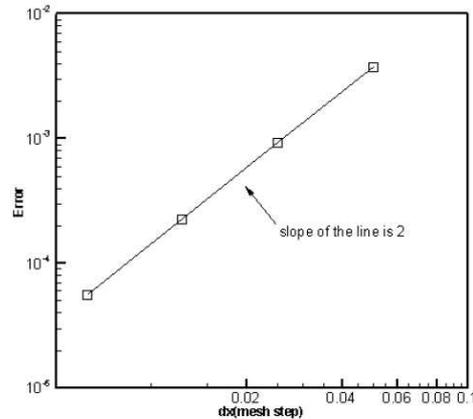


Figure 3: Numerical error of the velocity component u versus mesh spacing in the log scale for decaying vortex problem. The overall accuracy of this method is approximately 2.0.

3.2 Flows over a circular cylinder

To further examine the accuracy and efficiency of present method, numerical simulations of viscous flows past a circular cylinder are carried out. Both the steady and unsteady cases are considered. This problem has been studied extensively and there are many theoretical, experimental, and numerical results available in the literature. The problem is very attractive because the flow behavior depends on the Reynolds number

$$Re = \frac{U_{\infty} D}{\nu}. \quad (3.4)$$

Here, U_{∞} is the free stream velocity, D is the diameter of the cylinder, ν is the kinematic viscosity.

To obtain the accurate solution while keep high efficiency, the non-uniform mesh is used for the simulation, and the Taylor series expansion and least squares-based LBM (TLLBM) [18] is applied to provide the basic flow field.

In this work, the force can be directly computed from the velocity correction. The drag coefficient is defined as

$$C_d = \frac{F_D}{(1/2)\rho U_{\infty}^2 D}. \quad (3.5)$$

Here F_D is the drag force, which can be calculated by

$$F_D = - \int_{\Omega} f_x dx, \quad (3.6)$$

where f_x is the x component of force density \mathbf{f} . Similar to the drag coefficient, the lift coefficient can be defined as

$$C_l = \frac{F_L}{(1/2)\rho U_{\infty}^2 D}, \quad (3.7)$$

Table 1: Numerical error of velocity component u for the decaying vortex problem given by present method.

Mesh size	dx ($dx=dy$)	U_0	Err/U_0	Order of accuracy
41×41	0.05	0.025	3.74707535E-03	-
81×81	0.025	0.0125	9.24767769E-04	2.02
161×161	0.0125	0.00625	2.26095909E-04	2.03
321×321	0.00625	0.003125	5.57207403E-05	2.02

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

$$F_L = - \int_{\Omega} f_y d\mathbf{x}, \quad (3.8)$$

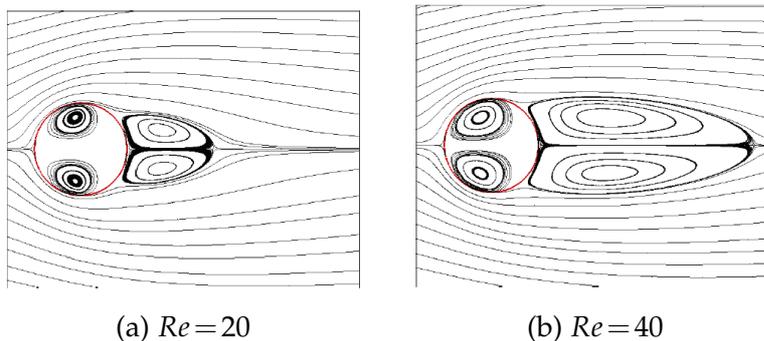
f_y is the y component of force density \mathbf{f} .

3.2.1 Steady flow past a circular cylinder

For the numerical simulation, the initial fluid density is set as $\rho=1.0$ and the initial velocity is taken as free stream velocity $U_{\infty}=0.1$. The equilibrium distribution functions are employed at far field boundaries.

For the simulation of steady flow, the Reynolds numbers of $Re=20$ and 40 are selected. The computational domain is $60D \times 40D$ with mesh size of 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 $(24D, 20D)$.

Fig. 4 shows the streamlines when flows reach the steady state for the cases of $Re=20$ and 40 . At the same time, the results using conventional IB-LBM proposed by Feng and Michaelides [6] are plotted in Fig. 5. It can be seen clearly that using present method, the streamlines, which are enclosed by the boundary of cylinder, do not penetrate the cylinder surface. In contrast, the penetration of streamlines to the cylinder surface by the conventional IB-LBM is obvious. The main reason is that the no-slip boundary condition is strictly guaranteed in the present method while it is only approximately satisfied in the conventional IBM.

Figure 4: Streamlines for the flow over a circular cylinder at $Re=20$ and $Re=40$ obtained by present method.

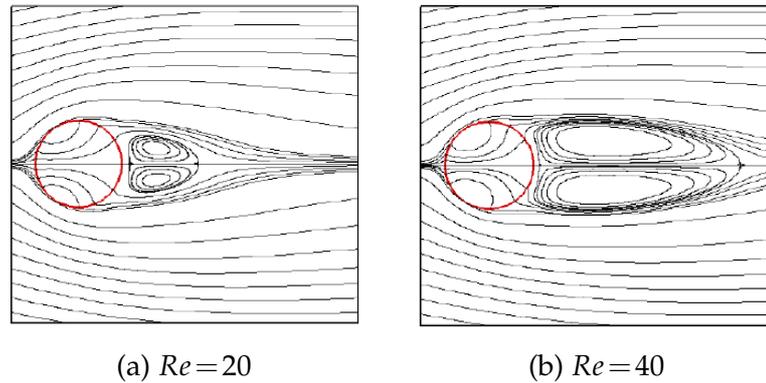


Figure 5: Streamlines for the flow over a circular cylinder at $Re = 20$ and $Re = 40$ obtained by conventional IB-LBM.

When the flow reaches steady state, there will be a pair of stationary recirculating eddies behind the cylinder. The length of the recirculating region L , which is from the nearest point of the cylinder to the end of the eddy, increases with the Reynolds number. Usually, we define the non-dimensional recirculating region as $2L/D$. In Table 2, we compare the drag coefficient and length of recirculating region with the previous results [19–21]. From the table, it can be seen that our numerical results agree well with those in the literature.

Table 2: Comparison of drag coefficient C_d and recirculation length L for steady flow over a circular cylinder at $Re = 20$ and 40.

Case	References	C_d	L
$Re = 20$	Dennis and Chang [19]	2.045	1.88
	Fornberg [20]	2.000	1.82
	He and Doolen [21]	2.152	1.842
	Present	2.072	1.86
$Re = 40$	Dennis and Chang [19]	1.522	4.69
	Fornberg [20]	1.498	4.48
	He and Doolen [21]	1.499	4.49
	Present	1.545	4.60

3.2.2 Unsteady flow past a circular cylinder

To further validate our approach, the unsteady flow around the cylinder at Reynolds numbers of $Re = 100$ and 200 is simulated. In this case, the initial perturbation in the flow field would bring the vortices to form the well-known Karman vortex street. Finally, the flow becomes periodically unsteady. The Strouhal number is defined as the dimensionless frequency with which the vortices are shed behind the body

$$St = \frac{f_q D}{U_\infty}, \quad (3.9)$$

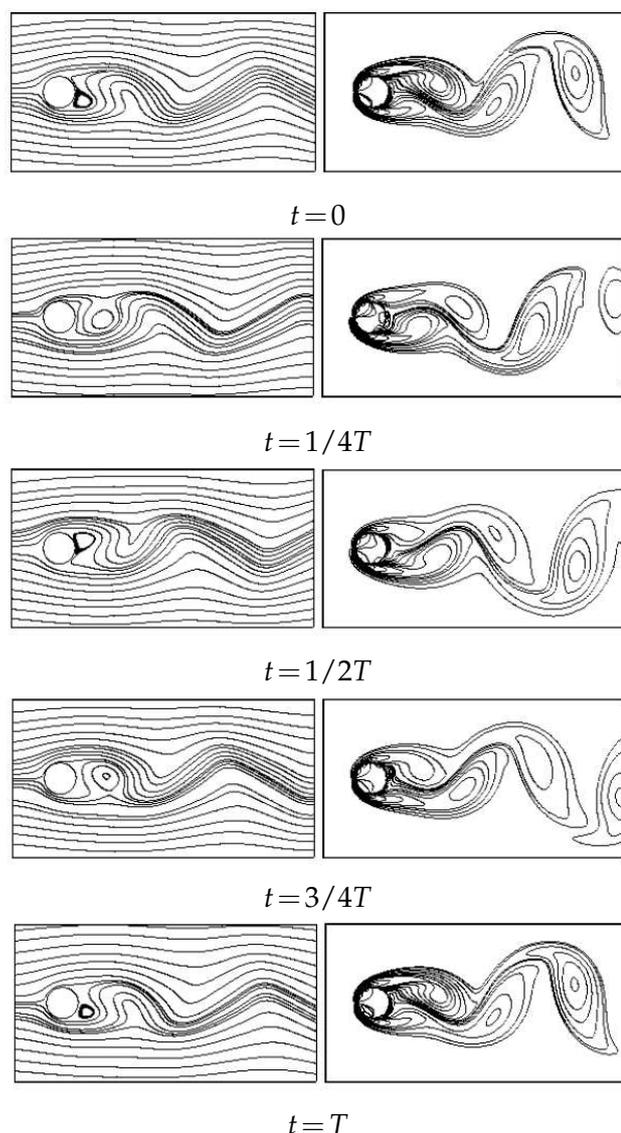


Figure 6: Vorticity contours and streamlines for the flow over a circular cylinder at $Re=100$ in a complete cycle.

where f_q is the vortex shedding frequency.

For the simulations at $Re=100$ and 200 , the computational domain is $60D \times 40D$ with the mesh size of 681×491 . The region around the cylinder is $1.2D \times 1.2D$ with a uniform mesh size of 149×149 . The cylinder is located at $(24D, 20D)$. Table 3 displays the present time-averaged drag coefficient and Strouhal number together with results from [22–25]. Good agreement is shown in this table. In particular, the Strouhal numbers computed by present method are in excellent agreement with the experimental data by Williamson [25]. The instantaneous vorticity contours of vortex shedding and the stream-

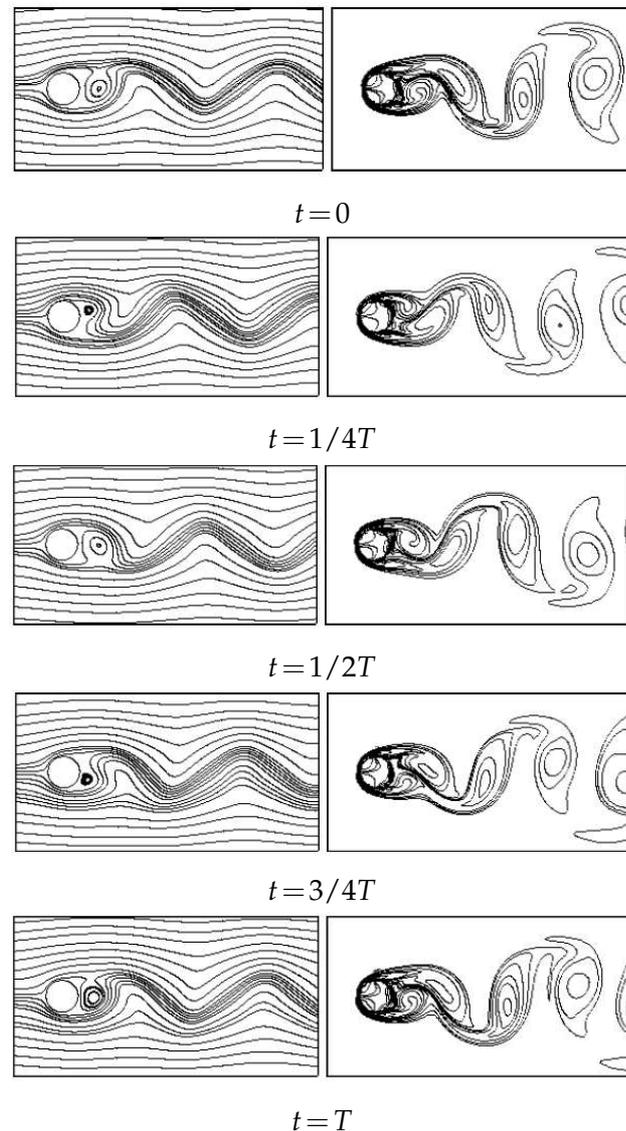


Figure 7: Vorticity contours and streamlines for the flow over a circular cylinder at $Re=200$ in a complete cycle.

lines in one complete cycle at $Re = 100$ and 200 are plotted in Figs. 6 and 7. The Karman vortex street of the flow over a circular cylinder can be clearly seen in these figures. Similar to the steady flow, the present results also do not show any streamline passing through the cylinder. This further demonstrates that the no-slip boundary condition is accurately satisfied in the present work.

Fig. 8 presents the periodic time evolutions of the drag and lift coefficients on the surface of cylinder at $Re = 100$ and 200 . It can be seen from this figure that the period of drag coefficient is different from that of lift coefficient. The period of lift coefficient is

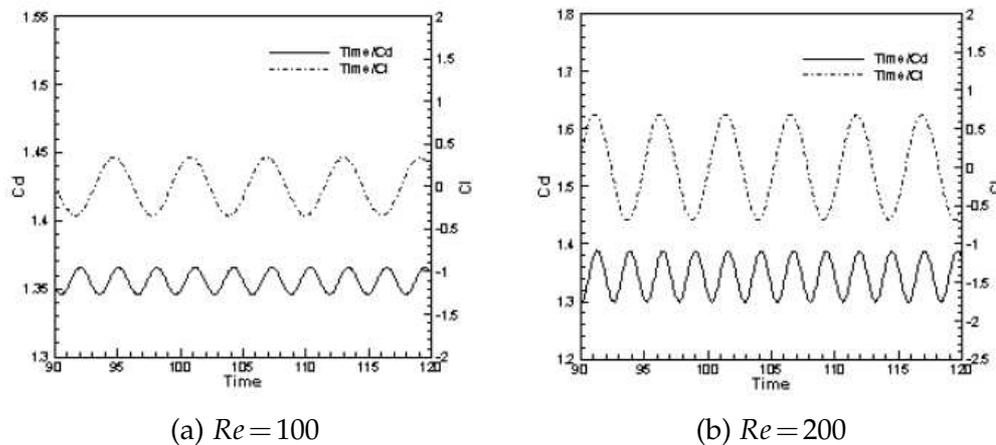


Figure 8: Evolution of drag and lift coefficients for flow over a cylinder at $Re=100$ and 200 computed by present method.

about two times the period of drag coefficient.

3.2.3 Comparison of methods for force calculation

In the work of Shu et al. [12], they used the control volume method to calculate the drag force and the lift force on the immersed body. Its process is quite tedious. In this study, we can directly calculate the drag and lift forces from the obtained velocity correction through the relationship between the velocity correction and the force density.

Fig. 9 shows the time evolution of the drag and lift coefficients on the surface of cylinder at $Re = 100$ and 200 , which are calculated by the control volume method. Table 4 shows the Comparison of the drag coefficient C_d between the present method and con-

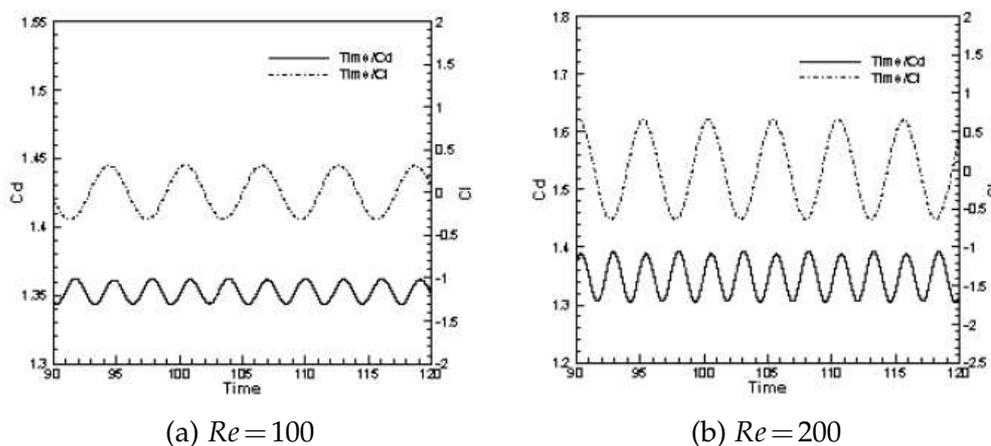


Figure 9: Evolution of drag and lift coefficients for flow over a cylinder at $Re=100$ and 200 computed by control volume method.

Table 3: Comparison of time-averaged drag coefficient C_d and Strouhal number St for unsteady flow over a circular cylinder at $Re=100$ and 200.

Case	References	C_d	St
$Re=100$	Braza et al. [22]	1.364	0.160
	Liu et al. [23]	1.350	0.164
	Ding et al. [24]	1.325	0.164
	Williamson [25]	-	0.166
	Present	1.356	0.165
$Re=200$	Braza et al. [22]	1.40	0.20
	Liu et al. [23]	1.31	0.192
	Ding et al. [24]	1.327	0.196
	Williamson [25]	-	0.197
	Present	1.345	0.195

Table 4: Comparison of time-averaged drag coefficient C_d computed by present method and control volume method.

Case	Reynolds number	C_d (present method)	C_d (control volume method)
Steady flow	$Re=20$	2.072	2.076
	$Re=40$	1.545	1.551
Unsteady flow	$Re=100$	1.356	1.354
	$Re=200$	1.345	1.349

control volume method. We can see that although the present method is very simple, its numerical results compare very well with those by the control volume method.

4 Conclusions

This paper presents a boundary condition-implemented immersed boundary-lattice Boltzmann method (IB-LBM). It overcomes two major drawbacks of conventional IB-LBM. The no-slip boundary condition is directly implemented in this work while it is only approximately satisfied in the conventional IB-LBM. The first-order Dirac delta function interpolation often adopted in the conventional IB-LBM is not applied in this work. Instead, the present work uses the second-order polynomial interpolation to implement the physical boundary condition. Another advantage of present method is the simple calculation of force on the boundary. It can be directly computed from the relationship between the velocity correction and the force density.

The accuracy and efficiency of present method are examined by its application to simulate two dimensional vortex decaying problem, steady and unsteady flows past a circular cylinder. The obtained numerical results are in good agreement with available data in the literature. In particular, the streamlines obtained by present method do not show any flow penetration to the solid boundary, which is often appeared in the results of conventional IB-LBM. The present method may have a great potential in simulation of steady and unsteady flows.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (11272153).

References

- [1] C. S. PESKIN, *Numerical analysis of blood flow in the heart*, J. Comput. Phys., 25 (1977), pp. 220–252.
- [2] D. GOLDSTEIN, R. HADLER AND L. SIROVICH, *Modeling a no-slip flow boundary with an external force field*, J. Comput. Phys., 105 (1993), 354.
- [3] M. C. LAI AND C. S. PESKIN, *An immersed boundary method with formal second-order accuracy and reduced numerical viscosity*, J. Comput. Phys., 160 (2000), pp. 705–719.
- [4] Y. H. TSENG AND J. H. FERZIGER, *A ghost-cell immersed boundary method for flow in complex geometry*, J. Comput. Phys., 192 (2003), pp. 593–623.
- [5] L. F. LIMA E SILVA, A. SILVEIRA-NETO AND J. J. R. DAMASCENO, *Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method*, J. Comput. Phys., 189 (2003), pp. 351–370.
- [6] Z. G. FENG AND E. E. MICHAELIDES, *The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems*, J. Comput. Phys., 195 (2004), pp. 602–628.
- [7] Z. G. FENG AND E. E. MICHAELIDES, *Proteus: a direct forcing method in the simulations of particulate flows*, J. Comput. Phys., 202 (2005), pp. 20–51.
- [8] E. FADLUM, R. VERZICCO, P. ORLANDI AND J. MOHD-YUSOF, *Combined immersed-boundary finite-difference methods for three dimensional complex flow simulations*, J. Comput. Phys., 161 (2000), pp. 35–60.
- [9] X. D. NIU, C. SHU, Y. T. CHEW AND Y. PENG, *A momentum exchange-based immersed boundary-lattice Boltzmann method for viscous fluid flows*, Phys. Lett. A, 354 (2006), pp. 173–182.
- [10] A. J. C. LADD, *Numerical simulations of particulate suspensions via a discretized Boltzmann equation part I: Theoretical foundation*, J. Fluid Mech., 271 (1994), pp. 285–310.
- [11] J. KIM, D. KIM AND H. CHOI, *An immersed-boundary finite-volume method for simulations of flow in complex geometries*, J. Comput. Phys., 171 (2001), pp. 132–150.
- [12] C. SHU, N. Y. LIU AND Y. T. CHEW, *A novel immersed boundary velocity correction-lattice Boltzmann method and its application to simulate flow past a circular cylinder*, J. Comput. Phys., 226 (2007), pp. 1607–1622.
- [13] J. WU AND C. SHU, *Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications*, J. Comput. Phys., 228 (2009), pp. 1963–1979.
- [14] Z. GUO, C. ZHENG AND B. SHI, *Discrete lattice effects on the forcing term in the lattice Boltzmann method*, Phys. Rev. E, 65 (2002), 046308.
- [15] Y. H. QIAN, D. D’HUMIERES AND P. LALLEMAND, *Lattice BGK model for Navier-stokes equation*, Europhys. Lett., 17 (1992), pp. 479–484.
- [16] X. HE, S. CHEN AND R. ZHANG, *A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh-Taylor instability*, J. Comput. Phys., 152 (1999), pp. 642–663.
- [17] D. J. CHEN, K. H. LIN AND C. A. LIN, *Immersed boundary method based lattice Boltzmann to simulate 2D and 3D complex geometry flows*, Int. J. Mod. Phys. C, 18 (2007), pp. 585–594.

- [18] C. SHU, X. D. NIU AND Y. T. CHEW, *Taylor series expansion- and least square-based lattice Boltzmann method: two-dimensional formulation and its applications*, Phys. Rev. E, 65 (2002), 036708.
- [19] S. C. R. DENNIS AND G. Z. CHANG, *Numerical solutions for steady flow past a circular cylinder at Reynolds number up to 100*, J. Fluid Mech., 42 (1970), pp. 471–489.
- [20] B. FORNBERG, *A numerical study of steady viscous flow past a circular cylinder*, J. Fluid Mech., 98 (1980), pp. 819–855.
- [21] X. HE AND G. DOOLEN, *Lattice Boltzmann method on curvilinear coordinates system: flow around a circular cylinder*, J. Comput. Phys., 134 (1997), pp. 306–315.
- [22] M. BRAZA, P. CHASSAING AND H. HA-MINH, *Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder*, J. Fluid Mech., 165 (1986), 79.
- [23] C. LIU, X. ZHENG AND C. H. SUNG, *Preconditioned multigrid methods for unsteady incompressible flows*, J. Comput. Phys., 139 (1998), 35.
- [24] H. DING, C. SHU, K. S. YEO AND D. XU, *Simulation of incompressible viscous flows past a circular cylinder by hybrid FD scheme and meshless least square-based finite difference method*, Comput. Methods Appl. Mech. Eng., 193 (2004), pp. 727–744.
- [25] C. H. K. WILLIAMSON, *Oblique and parallel models of vortex shedding in the wake of a circular cylinder at low Reynolds number*, J. Fluid Mech, 206 (1989), 579.