

# Simulation of incompressible viscous flows around moving objects by a variant of immersed boundary-lattice Boltzmann method

J. Wu<sup>1</sup>, C. Shu<sup>1,\*</sup>,<sup>†</sup> and Y. H. Zhang<sup>2</sup>

<sup>1</sup>*Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore*

<sup>2</sup>*Department of Mechanical Engineering, University of Strathclyde, Glasgow G1 1XJ, U.K.*

## SUMMARY

A variant of immersed boundary-lattice Boltzmann method (IB-LBM) is presented in this paper to simulate incompressible viscous flows around moving objects. As compared with the conventional IB-LBM where the force density is computed explicitly by Hook's law or the direct forcing method and the non-slip condition is only approximately satisfied, in the present work, the force density term is considered as the velocity correction which is determined by enforcing the non-slip condition at the boundary. The lift and drag forces on the moving object can be easily calculated via the velocity correction on the boundary points. The capability of the present method for moving objects is well demonstrated through its application to simulate flows around a moving circular cylinder, a rotationally oscillating cylinder, and an elliptic flapping wing. Furthermore, the simulation of flows around a flapping flexible airfoil is carried out to exhibit the ability of the present method for implementing the elastic boundary condition. It was found that under certain conditions, the flapping flexible airfoil can generate larger propulsive force than the flapping rigid airfoil. Copyright © 2009 John Wiley & Sons, Ltd.

Received 25 November 2008; Revised 23 January 2009; Accepted 28 January 2009

KEY WORDS: immersed boundary method; lattice Boltzmann method; moving objects; incompressible flow; velocity correction; non-slip condition

## 1. INTRODUCTION

In recent years, the study on interaction between fluid and moving objects receives more and more attention as moving boundary problems are often appeared in the study of fish motion, insect flight, blood flow through heart valves, and countless others. Simulation of flows around moving

\*Correspondence to: C. Shu, Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore.

<sup>†</sup>E-mail: mpeshuc@nus.edu.sg

Contract/grant sponsor: National Natural Science Foundation of China; contract/grant number: 10728206

objects accurately and efficiently puts a great challenge to numerical techniques, and is currently at the forefront in the computational fluid dynamics.

The numerical approaches for simulation of flows around moving objects can be roughly classified into two major categories, boundary conforming methods and non-boundary conforming methods. For the boundary conforming methods, the boundary points coincide with the mesh points. So, the physical boundary condition can be implemented directly. One of popular methods in this category is the use of time-dependent coordinate transformation [1–4], in which the moving physical domain is transformed into a fixed computational domain and all the numerical computations are easily performed in the computational domain. This approach is very efficient for the case where the whole object moves at the same mode. When the part of objects moves locally with a different mode, the multi-block time-dependent coordinate transformation, which could be very tedious, has to be applied. To consider the general moving boundary problem, Li *et al.* [5] proposed the moving mesh finite element algorithm, in which re-meshing and interpolation are needed. Perhaps, the most popular boundary conforming method is the arbitrary Lagrangian Eulerian (ALE) approach [6–11], which is normally applied with finite difference, finite volume, and finite element schemes. Owing to regenerating the mesh to conform to the boundary at all times, it becomes difficult for the ALE approach to solve the moving boundary problems with complex geometry, especially for the three-dimensional problem. In the category of non-boundary conforming methods, the governing equations are solved on a fixed Cartesian grid, and the boundary no longer coincides with the grid surface. The effect of boundary is accounted through the proper treatment of the solution variables at grid cells around the boundary. As compared with the boundary conforming methods, the non-boundary conforming methods eliminate the requirement of tedious grid adaptation, which makes the simulation of flows around complex boundaries undergoing movement be more straightforward.

In terms of the treatment of the boundary conditions, the non-boundary conforming methods can be further classified as Cartesian grid methods and immersed boundary methods. In the Cartesian grid methods, the boundary is treated as a sharp interface, and the grid cells around the body surface are cut by the immersed boundary. The Cartesian grid methods have been successfully applied to solve many inviscid and viscous flow problems [12–16]. However, due to irregular structures of the cut cells, the calculation of fluxes at the interface of cut cells requires complicated treatment, which may bring inconvenience and affect the computational efficiency. Recently, Zhou *et al.* [17] proposed an efficient Cartesian grid method, namely, the local domain-free discretization (DFD) method, for simulation of compressible flows around moving boundaries. In the local DFD method, the boundary information is transferred to an adjacent point to the boundary through low-order interpolation.

The immersed boundary method (IBM) may be the simplest non-boundary conforming method. It has been firstly proposed by Peskin [18] in the 1970s when he studied the blood flow in the human heart. Since then, numerous modifications and refinements have been proposed and a number of variants of this approach were proposed [19–22]. In the IBM, the flow field is represented by a set of Eulerian points, which are in fact the fixed Cartesian mesh points, and the boundary of immersed object is represented by a set of Lagrangian points. The basic idea of IBM is to treat the physical boundary as deformable with high stiffness. A small distortion of the boundary will yield a force that tends to restore the boundary into its original shape. The balances of such forces are distributed into the Eulerian points and the Navier–Stokes (N–S) equations with a body force are solved on the whole domain including exterior and interior of the object. Usually, in the IBM, the flow field is obtained by solving the N–S equations. Recently, as an alternative computational

technique to the N-S solver, the lattice Boltzmann method (LBM) [23] has been proven to be an efficient approach for simulation of flow field. Based on the kinetic theory, the LBM studies the dynamics of fictitious particles by using the density distribution functions. It contains the collision and streaming processes. The macroscopic variables such as density and velocity can be calculated through the density distribution functions from the conservation laws of mass and momentum. The major advantage of LBM is its simplicity, easy implementation, algebraic operation, and intrinsic parallel nature. Like the IBM, the standard LBM is usually applied on the Cartesian mesh. Owing to this common feature, it is desirable to combine these two methods together. Many efforts [24–26] have been made in this aspect.

One of key issues in the application of IBM is the computation of restoring force. Basically, there are three ways. The popular way is the penalty method [18], in which Hook's law is applied and the spring parameter needs to be specified by the user. The second way is called the direct forcing method, which has been introduced firstly by Fadlun *et al.* [27]. This way directly applies the momentum equations at the boundary points to compute the force density. The third way was proposed by Niu *et al.* [28], in which the momentum exchange at the boundary is used to compute the force. It is noted that all the three ways compute the restoring force explicitly. As pointed out by Shu *et al.* [29], the pre-calculated restoring force cannot guarantee that the corrected velocity field due to presence of immersed boundary satisfies the non-slip condition at the boundary point. As a result, obvious flow penetration to the immersed boundary can be observed in the IBM results. Flow penetration implies mass exchange across the boundary. As we know, mass exchange would bring the momentum exchange, leading to a numerical force. Clearly, this force error will affect the accuracy of lift and drag forces acting on the immersed object. This greatly limits the application of IBM to the moving boundary problems.

To overcome the flow penetration problem in IBM results, a variant of immersed boundary-lattice Boltzmann method (IB-LBM) is presented in this work, where the restoring force is not pre-calculated and the flow field is obtained by the lattice Boltzmann method. In the present work, the restoring force is determined by enforcing the non-slip condition on the boundary. Since the non-slip condition is accurately satisfied, no flow penetration can be found in the present results. Owing to this improvement, it is expected that the present approach can be well applied to simulate flows around moving objects. The flows around a moving circular cylinder, a rotationally oscillating cylinder, and an elliptic flapping wing are chosen to validate the present approach. The obtained results agree very well with available data in the literature. Furthermore, the flow around the flapping flexible airfoil is simulated to exhibit the ability of the present approach for implementing the elastic boundary condition.

## 2. NUMERICAL METHOD

### 2.1. Conventional immersed boundary method

In the immersed boundary method, the effect of boundary to the surrounding fluids is through a force density exerting on them. The governing equations of immersed boundary method for the viscous incompressible flows can be written as

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

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

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

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

where  $\mathbf{x}$ ,  $\mathbf{u}$ , and  $\mathbf{f}$  are the Eulerian coordinates, fluid velocity, and force density acting on the fluid, respectively,  $p$  is the fluid pressure,  $\rho$  is the fluid density, and  $\mu$  is the dynamic viscosity.  $\mathbf{X}$  and  $\mathbf{F}$  represent Lagrangian coordinates and boundary force density.  $\delta(\mathbf{x} - \mathbf{X}(s, t))$  is a Dirac delta function. Equations (1) and (2) are the N-S equations with external force. Equations (3) and (4) describe the interaction between the immersed boundary and the fluid flow by distributing the boundary force at the Lagrangian points to the Eulerian points and interpolating the velocity at the Eulerian points to Lagrangian points. The calculation of boundary force density  $\mathbf{F}$ , which is also called restoring force, is critical in the IBM. Using Hooke's law, it can be determined by

$$\mathbf{F}(s, t) = -k\Delta\zeta = -k(\mathbf{V}_{\text{fluid}}\Delta t - \mathbf{V}_{\text{wall}}\Delta t) \quad (5)$$

where  $\mathbf{V}_{\text{fluid}}$  is the fluid velocity at the boundary point interpolated from the surrounding fluid (Eulerian) points,  $\mathbf{V}_{\text{wall}}$  is the boundary velocity of the object,  $k$  is the spring coefficient. Note that the boundary force density can also be computed by the direct forcing method [27] and the momentum exchange method [28]. The basic solution process of IBM can be summarized as follows:

- (1) Set force density  $\mathbf{f}$  as zero at beginning. Solve Equations (1) and (2) to get flow variables at Eulerian points.
- (2) Interpolate velocity at Eulerian points to the boundary (Lagrangian) points by using Equation (4).
- (3) Use interpolated velocity given in step (2) and the boundary velocity of the object to compute the boundary force density  $\mathbf{F}$  by using Equation (5).
- (4) Compute the force density  $\mathbf{f}$  at Eulerian points by using Equation (3).
- (5) Solve Equations (1) and (2) with the force density  $\mathbf{f}$  to get the corrected velocity field at Eulerian points.
- (6) Go back to step (2) until the convergence criterion is satisfied.

It should be noted that in the above process, when the boundary force density  $\mathbf{F}$ , and therefore the force density  $\mathbf{f}$ , is computed explicitly, the new (corrected) velocity given in step (5) may not satisfy the non-slip boundary condition. So, IBM needs to continue the process until convergence state is reached, and hopes that at the converged state, the non-slip condition can be satisfied. However, we have to indicate that in the whole process, there is no guarantee to satisfy the non-slip boundary condition. Indeed, it is only approximately satisfied. This could be the major reason to cause flow penetration to the solid body in the conventional IBM results. As shown in the following section, we will present a variant of IB-LBM to enforce the non-slip boundary condition.

## 2.2. A Variant of IB-LBM method

Equations (1) and (2) are the governing equations for the flow field. In the lattice Boltzmann context, they can be replaced by the lattice Boltzmann equation. In this work, the form of lattice

Boltzmann equation proposed by Guo *et al.* [30] is adopted, which can be written as

$$f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \delta t, t + \delta t) - f_\alpha(\mathbf{x}, t) = -\frac{1}{\tau}(f_\alpha(\mathbf{x}, t) - f_\alpha^{\text{eq}}(\mathbf{x}, t)) + F_\alpha \delta t \quad (6)$$

$$F_\alpha = \left(1 - \frac{1}{2\tau}\right) w_\alpha \left(\frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^4} \mathbf{e}_\alpha\right) \cdot \mathbf{f} \quad (7)$$

$$\rho \mathbf{u} = \sum_\alpha \mathbf{e}_\alpha f_\alpha + \frac{1}{2} \mathbf{f} \delta t \quad (8)$$

where  $f_\alpha$  is the distribution function,  $f_\alpha^{\text{eq}}$  is its corresponding equilibrium state,  $\tau$  is the single relaxation time,  $\mathbf{e}_\alpha$  is the particle velocity,  $w_\alpha$  are coefficients that depend on the selected particle, and  $\mathbf{f}$  is the external force density. For the popular D2Q9 model [31], the particle velocity set is given by

$$\mathbf{e}_\alpha = \begin{cases} 0, & \alpha = 0 \\ (\cos[(\alpha-1)\pi/2], \sin[(\alpha-1)\pi/2])c, & \alpha = 1, 2, 3, 4 \\ \sqrt{2}(\cos[(\alpha-5)\pi/2 + \pi/4], \sin[(\alpha-5)\pi/2 + \pi/4])c, & \alpha = 5, 6, 7, 8 \end{cases} \quad (9)$$

where  $c = \delta x / \delta t$ ,  $\delta x$  and  $\delta t$  are the lattice spacing and time step, respectively. The corresponding equilibrium distribution function is

$$f_\alpha^{\text{eq}}(\mathbf{x}, t) = \rho w_\alpha \left[ 1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2 - (c_s |\mathbf{u}|)^2}{2c_s^4} \right] \quad (10)$$

with  $w_0 = \frac{4}{9}$ ,  $w_1 = w_2 = w_3 = w_4 = \frac{1}{9}$  and  $w_5 = w_6 = w_7 = w_8 = \frac{1}{36}$ .  $c_s = c / \sqrt{3}$  is the sound speed of this model. The relation between the relaxation time and the kinematic viscosity of fluid is

$$\nu = (\tau - \frac{1}{2}) c_s^2 \delta t$$

Note that this relationship is given from the Chapman–Enskog expansion [31] in such a way that the macroscopic flow variables such as density and velocity obtained in the LBM can satisfy N-S equations. If we define the intermediate velocity  $\mathbf{u}^*$  as

$$\rho \mathbf{u}^* = \sum_\alpha \mathbf{e}_\alpha f_\alpha \quad (11)$$

and the velocity correction as

$$\rho \delta \mathbf{u} = \frac{1}{2} \mathbf{f} \delta t \quad (12)$$

then Equation (8) can be written as

$$\mathbf{u} = \mathbf{u}^* + \delta \mathbf{u} \quad (13)$$

In the conventional IB-LBM,  $\mathbf{f}$  is computed explicitly by Equation (5) or the direct forcing method [25, 27] or the momentum exchange method [28]. When the corrected velocity field is obtained from Equation (13), there is no guarantee that the velocity at the boundary point interpolated from the corrected velocity field satisfies the non-slip boundary condition. To overcome this drawback,

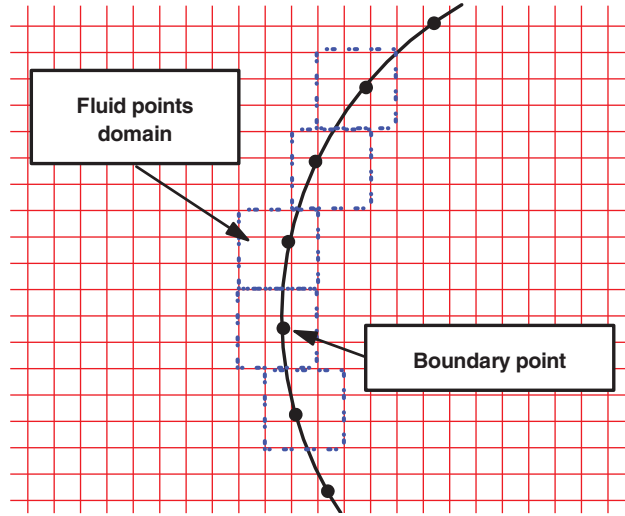


Figure 1. Configuration of boundary points and their surrounding fluid points.

we have to consider the force density  $\mathbf{f}$  as unknown, which is determined in such a way that the velocity at the boundary point interpolated from the corrected velocity field satisfies the non-slip boundary condition. As shown in Figure 1, the velocity correction  $\delta\mathbf{u}$  at Eulerian points is distributed from the velocity correction at the boundary (Lagrangian) points. In the IBM, the boundary of the object is represented by a set of Lagrangian points  $\mathbf{X}_B(s_l, t)$ ,  $l = 1, 2, \dots, m$ . Here, we can set an unknown velocity correction vector  $\delta\mathbf{u}_B^l$  at every Lagrangian point. The velocity correction  $\delta\mathbf{u}$  at the Eulerian point can be obtained by the following Dirac delta function interpolation:

$$\delta u(\mathbf{x}, t) = \int_{\Gamma} \delta\mathbf{u}_B(\mathbf{X}_B, t) \delta(\mathbf{x} - \mathbf{X}_B(s, t)) ds \quad (14)$$

In the actual implementation,  $\delta(\mathbf{x} - \mathbf{X}_B(s, t))$  is smoothly approximated by a continuous kernel distribution

$$\delta(\mathbf{x} - \mathbf{X}_B(s, t)) = D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) = \delta(x_{ij} - X_B^l) \delta(y_{ij} - Y_B^l) \quad (15)$$

where  $\delta(r)$  is proposed by Peskin [32] as

$$\delta(r) = \begin{cases} \frac{1}{4} \left( 1 + \cos\left(\frac{\pi|r|}{2}\right) \right), & |r| \leq 2 \\ 0, & |r| > 2 \end{cases} \quad (16)$$

Using Equation (15), the velocity correction at Eulerian points can be expressed as

$$\delta\mathbf{u}(\mathbf{x}_{ij}, t) = \sum_l \delta\mathbf{u}_B^l(\mathbf{X}_B^l, t) D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta s_l \quad (l = 1, 2, \dots, m) \quad (17)$$

where  $\Delta s_l$  is the arc length of the boundary element. In order to satisfy the no-slip boundary condition, the fluid velocity at the boundary point must be equal to the boundary velocity at the same position

$$\mathbf{U}_B^l(\mathbf{X}_B^l, t) = \sum_{i,j} \mathbf{u}(\mathbf{x}_{ij}, t) D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta x \Delta y \quad (18)$$

Here,  $\mathbf{U}_B^l$  is the boundary velocity;  $\mathbf{u}$  is the fluid velocity, which is corrected by the velocity correction  $\delta \mathbf{u}$

$$\mathbf{u}(\mathbf{x}_{ij}, t) = \mathbf{u}^*(\mathbf{x}_{ij}, t) + \delta \mathbf{u}(\mathbf{x}_{ij}, t) \quad (19)$$

where  $\mathbf{u}^*$  is the intermediate fluid velocity obtained from Equation (11). Note that the unknowns in Equations (18) and (19) are the velocity corrections at the boundary points,  $\delta \mathbf{u}_B^l$ . Substituting Equations (19) and (17) into Equation (18) gives

$$\begin{aligned} \mathbf{U}_B^l(\mathbf{X}_B^l, t) &= \sum_{i,j} \mathbf{u}^*(\mathbf{x}_{ij}, t) D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta x \Delta y \\ &+ \sum_{i,j} \sum_l \delta \mathbf{u}_B^l(\mathbf{X}_B^l, t) D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta s_l D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta x \Delta y \end{aligned} \quad (20)$$

Equation system (20) can be further rewritten as the following matrix form:

$$\mathbf{A}\mathbf{X} = \mathbf{B} \quad (21)$$

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

$$\Delta \mathbf{u}_l = \mathbf{U}_B^l(\mathbf{X}_B^l, t) - \sum_{i,j} \mathbf{u}^*(\mathbf{x}_{ij}, t) D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta x \Delta y \quad (l = 1, 2, \dots, m) \quad (22)$$

Note that the elements of matrix  $\mathbf{A}$  are only related to the boundary points and their nearby Eulerian points (see Figure 1). After obtaining the velocity correction at the boundary point by solving equation system (21), the velocity correction and the corrected velocity at Eulerian points are then calculated by Equations (17) and (19). In our simulation, the density and pressure are calculated by

$$\rho = \sum_{\alpha} f_{\alpha}, \quad P = c_s^2 \rho \quad (23)$$

With velocity correction  $\delta \mathbf{u}$ , the force density  $\mathbf{f}$  can be simply calculated by Equation (12) as

$$\mathbf{f} = 2\rho \delta \mathbf{u} / \delta t \quad (24)$$

Equation (24) can be applied at the boundary points to compute the lift and drag forces. The basic procedure of present IB-LBM is outlined as follows:

- *Step 1:* Set initial values, compute the elements of matrix  $\mathbf{A}$ , and get its inverse matrix  $\mathbf{A}^{-1}$ .
- *Step 2:* Use Equation (6) to get the density distribution function at time level  $t = t_n$  (initially setting  $F_{\alpha} = 0$ ) and compute the macroscopic variables using Equations (11) and (23).
- *Step 3:* Solve equation system (21) to get the velocity corrections at all boundary points and use Equation (17) to get velocity corrections at Eulerian points.
- *Step 4:* Correct the fluid velocity at Eulerian points using Equation (19) and obtain the force density using Equation (24).
- *Step 5:* Compute the equilibrium distribution function using Equation (10).
- *Step 6:* Repeat Step 2 to Step 5 until convergence is reached.

## 3. RESULTS AND DISCUSSION

## 3.1. Steady flow over a stationary circular cylinder

To demonstrate that the present approach has no flow penetration to the boundary of immersed object and provides more accurate forces acting on the object, the steady flow over a stationary circular cylinder is selected for simulation. This problem has been studied extensively and there are numerous theoretical, experimental, and numerical results available in the literature. Depending on the Reynolds number, different kind of flow behaviors can be characterized. Here, the Reynolds number is defined as

$$Re = \frac{U_\infty D}{\nu} \quad (25)$$

where  $U_\infty$  is the free stream velocity,  $D$  is the diameter of cylinder, and  $\nu$  is the kinematic viscosity.

It has been pointed out by Lai and Peskin [33] that the drag force arises from two sources: the shear stress and the pressure distribution along the body. The drag coefficient is defined as

$$C_d = \frac{F_D}{(\frac{1}{2})\rho U_\infty^2 D} \quad (26)$$

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

$$F_D = - \int_{\Omega} f_x d\mathbf{x} \quad (27)$$

where  $f_x$  stands for the  $x$ -component of force density  $\mathbf{f}$  at the boundary point. In our proposed IB-LBM, the boundary forces could be calculated directly from the velocity corrections by using Equation (24). It is a simple way for the force computation. There are some other ways to compute forces. For example, the control volume method could be employed. It is accomplished by integrating the momentum equations over a control volume  $\Omega$

$$F_D = - \left\{ \frac{d}{dt} \left( \int_{\Omega} u_1 d\Omega \right) + \int_S \left[ u_1 \vec{u} \cdot \vec{n} + p n_1 - \mu \left( \frac{\partial u_1}{\partial x_i} + \frac{\partial u_i}{\partial x_1} \right) \right] ds \right\} \quad (28)$$

where  $S$  is the control surface,  $\vec{n}$  is the normal vector to the control surface,  $\vec{n} = (n_1, n_2)$ , the subscripts 1 and 2 denote the  $x$ -direction and  $y$ -direction, respectively. Compared with our method, the control volume method needs to evaluate derivatives of flow variables, which may wipe out the simplicity of IB-LBM.

The simulations at  $Re=20$  and  $40$  are carried out. The computational domain is set by  $50D \times 50D$ . The cylinder is located at  $(20D, 25D)$ . The free stream velocity is taken as  $U_\infty=0.1$  and the fluid density is  $\rho=1.0$ . The computation starts with the given free stream velocity. At the far-field boundaries, the equilibrium distribution functions are used to implement the boundary condition. To well capture the accurate solution near the cylinder surface, fine grid is used around the cylinder. On the other hand, considering the computational efficiency, coarse grid is good enough for the region far away from the cylinder. To balance these two, the non-uniform mesh is used in the present simulation. As the standard lattice Boltzmann method is only applicable on the uniform mesh, in this work, we adopt the Taylor series expansion and least-squares-based lattice Boltzmann method (TLLBM) [34], which can be well applied on the non-uniform mesh.



Table I. Grid-dependent study for flow over a cylinder at  $Re=20$ .

Total mesh size	Mesh spacing of uniform mesh around cylinder	$C_d$	$L$
$301 \times 251$	$\frac{1}{50}$	2.084	0.96
$351 \times 301$	$\frac{1}{60}$	2.079	0.95
$401 \times 351$	$\frac{1}{70}$	2.075	0.94
$451 \times 401$	$\frac{1}{80}$	2.072	0.92

To investigate the convergence of present method with respect to the numerical resolution (mesh size), the cases with four different mesh sizes at  $Re=20$  were simulated. Table I compares the calculated length of recirculation bubbles  $L$  (based on the diameter of cylinder  $D$ ) and drag coefficient  $C_d$  under four different mesh sizes. Clearly, as mesh size increases, numerical results tend to be grid-independent.

In the following simulation for  $Re=20$  and 40, a very fine uniform mesh size of  $97 \times 97$  is used in a small region around the cylinder ( $1.2D \times 1.2D$ ). The streamlines are plotted in Figure 2 for the case of  $Re=40$ . As shown in the figure, a pair of symmetric recirculation bubbles appears in the wake of the cylinder. At the same time, we can observe two pairs of weak vortices enclosed inside the cylinder. It means that the flow inside the cylinder has been occluded by the boundary. This is an ideal case as it ensures no mass exchange between interior and exterior of the cylinder surface. As shown clearly in Figure 2(b) (near wall flow field), the streamlines outside and inside the cylinder attach the cylinder surface completely. To the best of our knowledge, this is the first such promising result obtained by IBM and its various versions. The good performance of present results is indeed due to enforcement of non-slip boundary condition to prevent flow penetration to the boundary.

Table II compares the values of  $L$  and  $C_d$  for two cases with previous data [35–38]. Also shown in the table are the results obtained by conventional IB-LBM [28]. From the table, we can see clearly that the present drag coefficients are closer to the experimental data and numerical results obtained by body-fitted N-S solvers than the conventional IB-LBM results. This is also because the non-slip boundary condition is enforced in the present method and flow penetration is avoided. As a consequence, the boundary force could be computed more accurately. This is of underlying importance for moving boundary flow problems.

### 3.2. Flow over a moving circular cylinder

For the case presented above, the cylinder is stationary. To investigate the capability of the present method for modeling moving boundary flow problems, the simulation of flow over a moving cylinder is carried out. To ease the simulation, the uniform mesh is used. For making comparison, the flow over a stationary cylinder is also simulated on the same mesh. The computational domain is set by  $32D \times 32D$  with a mesh size of  $1281 \times 1281$ . The stationary cylinder is located at  $(8D, 16D)$ . The moving cylinder moves toward the left from the position of  $(30D, 16D)$ . The Reynolds number for both cases are taken as  $Re=40$ .

To compare the results of moving cylinder case with that of stationary cylinder case, we can adjust the frame of reference. It can be easily implemented by adding an opposite velocity  $U_\infty$  onto

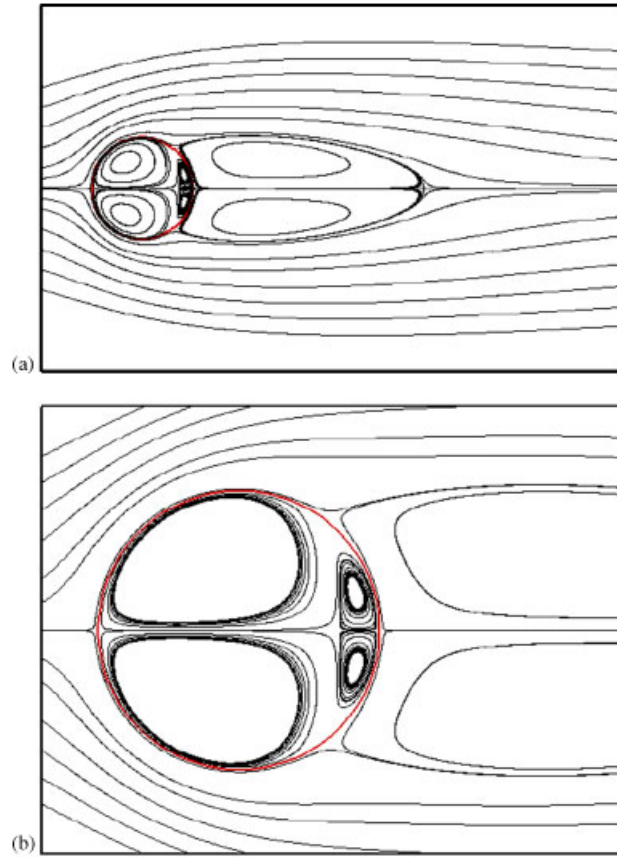


Figure 2. Streamlines for flow over a stationary cylinder at  $Re=40$ : (a) global view and (b) near wall flow field.

the velocity in the Eulerian mesh. The adjusted streamlines are shown in Figure 3. For comparison, the streamlines of stationary cylinder case are also shown in Figure 3. It is apparent that the results of both cases have good agreement with each other. The good agreement of both results can be further confirmed by Figure 4, which compares the pressure profiles on the surface of cylinder for two cases. Here, the pressure distribution is calculated by

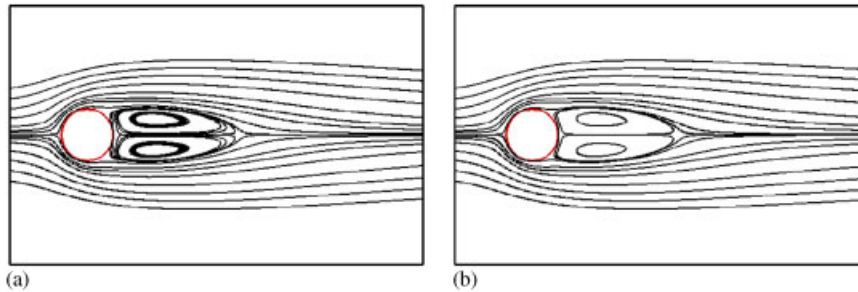
$$C_p = \frac{P_B - P_\infty}{(\frac{1}{2})\rho U_\infty^2} \quad (29)$$

where  $P_B$  and  $P_\infty$  are, respectively, the pressure on the boundary of cylinder and far-field pressure. Similar to Equation (18), the pressure  $P_B$  could be calculated by

$$P_B = \sum_{i,j} P(\mathbf{x}_{ij}) D_{ij} (\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta x \Delta y \quad (30)$$

Table II. Comparison of drag coefficients and length of recirculation zone.

Case	Authors	$C_d$	$L$
$Re=20$	Dennise and Chang [35]	2.05	0.94
	Nieuwstadt and Keller [36]	2.053	0.893
	Fornberg [37]	2.000	0.94
	Shukla <i>et al.</i> [38]	2.07	0.92
	Niu <i>et al.</i> [28]	2.144	0.945
	Present	2.072	0.92
$Re=40$	Dennise and Chang [35]	1.52	2.35
	Nieuwstadt and Keller [36]	1.54	2.18
	Fornberg [37]	1.498	2.25
	Shukla <i>et al.</i> [38]	1.55	2.34
	Niu <i>et al.</i> [28]	1.589	2.26
	Present	1.554	2.3

Figure 3. Streamlines for moving and stationary cylinder cases at  $Re=40$ : (a) moving cylinder case and (b) stationary cylinder case.

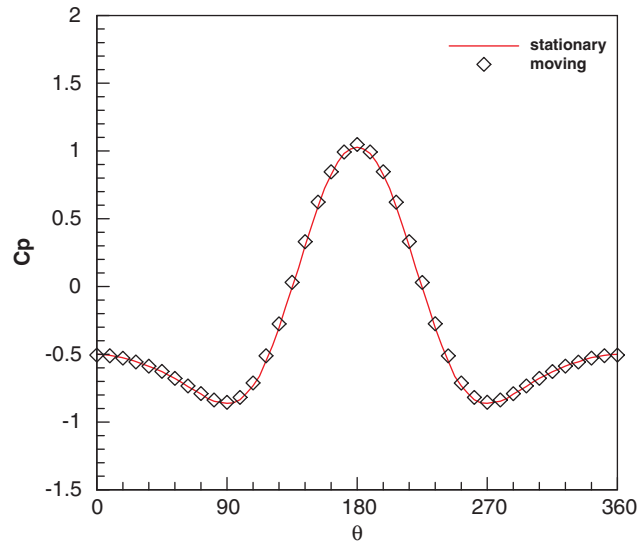
Through this example, on one hand, we demonstrate the capability of the present method for simulation of flows around a moving object; on the other hand, we show Galilean invariance of the present approach.

### 3.3. Flow over a rotationally oscillating cylinder

Vortex shedding in the near wake behind a bluff body could produce periodically oscillating drag and lift forces. These fluctuating forces would produce structural vibrations, acoustic noise, or resonance. Hence, it is very important from practical engineering perspective to control vortex shedding appropriately. Many attempts for such control have been made recently. One of simple attempts is the rotary control of cylinder wake [39, 40]. Here, we apply our developed method to simulate such moving boundary flow problem.

The control of motion has the form of the rotary oscillation of the cylinder with the instantaneous rotational velocity given by

$$\gamma(t) = \gamma_0 \sin(2\pi f t) = A \frac{2U_\infty}{D} \sin\left(2\pi St_f \frac{U_\infty}{D} t\right) \quad (31)$$

Figure 4. Pressure distribution on the surface of cylinder at  $Re=40$ .

where  $U_\infty$  and  $D$  represent the free stream velocity and diameter of cylinder, respectively.  $f$  and  $\gamma_0$  are the frequency and the rotation amplitude, respectively, which can be expressed in terms of non-dimensional parameters: the Strouhal number  $St_f = fD/U_\infty$  and the normalized amplitude  $A = \gamma_0 D/(2U_\infty)$ . These two parameters are sufficient to characterize the control.

The computational domain is set by  $50D \times 40D$  with a mesh size of  $551 \times 301$  and the cylinder is located at  $(20D, 20D)$ . The region around the cylinder is  $1.2D \times 1.2D$  with a very fine uniform mesh size of  $121 \times 121$ . The whole domain is discretized on the non-uniform mesh. Hence, the TLLBM [34] is applied.

In the present simulation,  $A=2.0$  is selected. Based on the free stream velocity  $U_\infty$  and diameter of cylinder  $D$ , the Reynolds number is taken as  $Re=100$ . To verify our method for this problem, the case of  $St_f=0.163$  is simulated first. At  $Re=100$ , the natural vortex shedding ( $St_n$ ) for flow over a stationary cylinder is about 0.163. As indicated by Choi *et al.* [39], when  $St_f=St_n$ , the interesting vortex shedding pattern would happen, which is shown in Figure 5. Note that the result of Choi *et al.* [39] is also included in Figure 5 for comparison. It is clear from the figure that the present result compares well with that of Choi *et al.* [39]. We also simulate the case of  $St_f=0.4$  to compare the forces exerted on the cylinder with that of Choi *et al.* [39]. The computational domain of their simulation is  $0.5D \leq r < 50D$  and  $0 \leq \theta < 2\pi$  and the O-type grid with a mesh size of  $241 \times 241$  is used. The obtained time-averaged coefficient and maximum amplitude of lift coefficient fluctuation are 1.302 and 0.321, respectively. The corresponding values of Choi *et al.* [39] are 1.231 and 0.299. The small differences between the present results and those of Choi *et al.* [39] may be caused by the use of IBM. Since the delta function, which is employed to relate the information between immersed boundary and flow field, only has the first order of accuracy, the computed forces in IBM are a little bit larger. In addition, the computational domain of Choi *et al.* [39] is much bigger than that of the present simulation. This may also induce the differences between two numerical results.

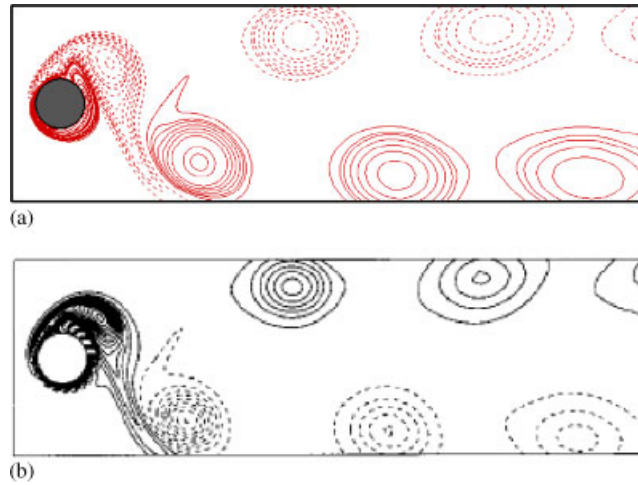


Figure 5. Comparison of vorticity contours when  $St_f = St_n$  at  $Re = 100$ : (a) result of present simulation and (b) result of Choi *et al.* [39].

After verification of the method, the cases for five different forcing frequencies are simulated. They are  $St_f = 0.1, 0.16, 0.3, 0.7$ , and  $0.9$ . Figure 6 illustrates the instantaneous vorticity contours obtained from the numerical simulations at the same time levels. As a reference, the flow pattern of stationary cylinder case is also included. The similar vorticity contours have been obtained by Protas and Wesfreid [40]. Figure 6(b) shows that the vortices with  $St_f = 0.1$  have the wavelength about twice as big as that in the stationary case (shown in Figure 6(a)). In Figure 6(c), the cylinder releases two arrays of regular vortices per half-cycle. This flow pattern happens when the forcing frequency is close to the natural shedding frequency. In Figure 6(d), the wake structure is synchronized with the rotary oscillation forcing and moves downstream without coalescence. This feature can be utilized to stabilize the flow in the wake of the cylinder. In Figure 6(e), the near-wake structure also becomes synchronized with the forcing. However, it becomes unstable and merges into a large-scale vortex some distance downstream. In Figure 6(f), the oscillation forcing only generates small-scale vortices in the shear layers near the cylinder, and most of the wake resembles that behind the stationary cylinder.

The flow behind a rotationally oscillating cylinder can be classified into two categories: lock-on and non-lock-on. According to Choi *et al.* [39], the cases for  $St_f$  being equal to  $0.1, 0.16$ , and  $0.3$  belong to lock-on and the remaining cases belong to non-lock-on. As pointed out in [39], one of the characteristics for the non-lock-on region is the occurrence of modulation phenomenon. The modulation frequency is very low as compared with the forcing and vortex shedding frequencies. Figure 7 shows the time histories of the lift and drag coefficients for non-lock-on cases. Figure 8 shows the variations of the time-averaged drag coefficients and the maximum amplitude of the lift coefficient fluctuations due to rotary oscillation forcing. In the present simulation, the mean drag coefficient for the stationary cylinder is about  $1.361$ . It is shown in Figure 8(a) that the mean drag decreases markedly with increasing  $St_f$  in the lock-on region, and it increases gradually with increasing  $St_f$  in the non-lock-on region. The local minimum of mean drag exists near the boundary between the lock-on and non-lock-on regions. A similar trend is found for the maximum amplitude of the lift coefficient fluctuations in Figure 8(b).

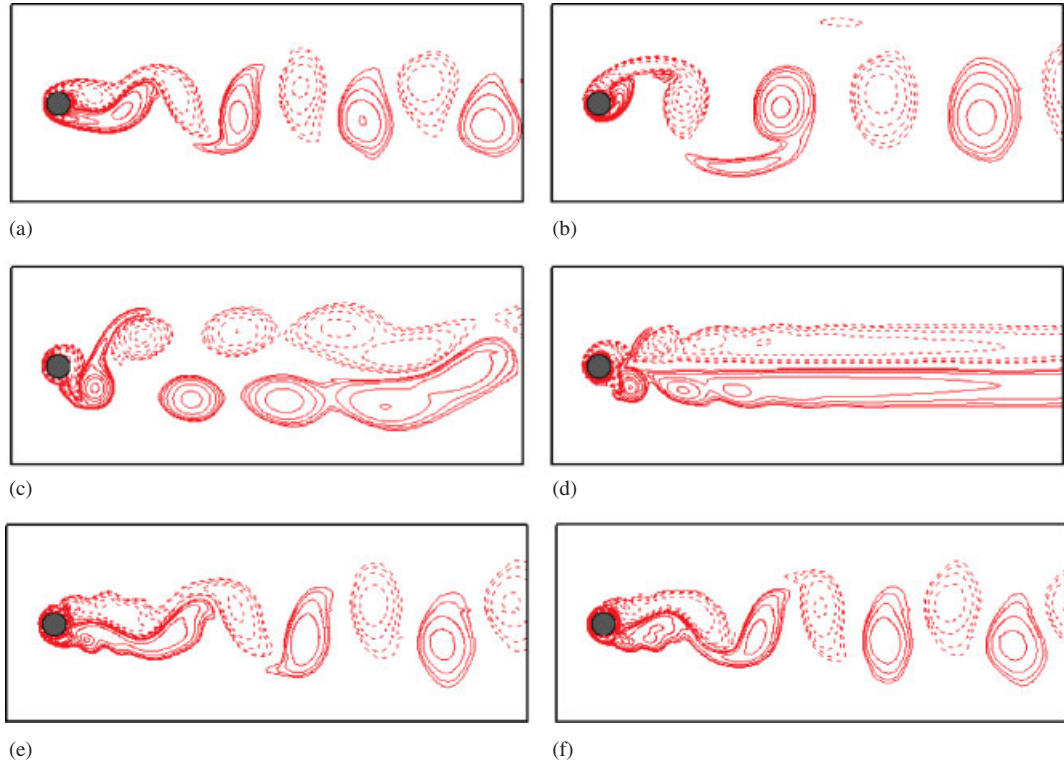


Figure 6. Vorticity contours at  $Re = 100$  for flow around a rotationally oscillating cylinder (solid lines: positive; dashed lines: negative): (a) stationary; (b)  $St_f = 0.1$ ; (c)  $St_f = 0.16$ ; (d)  $St_f = 0.3$ ; (e)  $St_f = 0.7$ ; and (f)  $St_f = 0.9$ .

### 3.4. Unsteady flows at low Reynolds number flapping flight

The flapping insect flight has fascinated physicists and biologists for more than one century. As compared with the fixed wing, flapping wing demonstrates attractive lift enhancement due to unsteady effects, which is very important for insect flight. Using the proposed method, we will simulate unsteady flows arising from flapping of a single elliptical wing to display the flow patterns associated with wing translation and rotation, as well as stroke reversal. The obtained fluid forces on the object are compared with those obtained numerically and experimentally in References [41, 42].

In the current simulation, the elliptical wing of aspect ratio 10 follows a prescribed sinusoidal translational and rotational motion. Specially, the wing sweeps in the horizontal plane and pitches about its center

$$x(t) = \frac{A_0}{2} \cos(2\pi f t), \quad y(t) = 0 \quad (32)$$

$$\alpha(t) = \alpha_0 + \beta \sin(2\pi f t + \varphi) \quad (33)$$

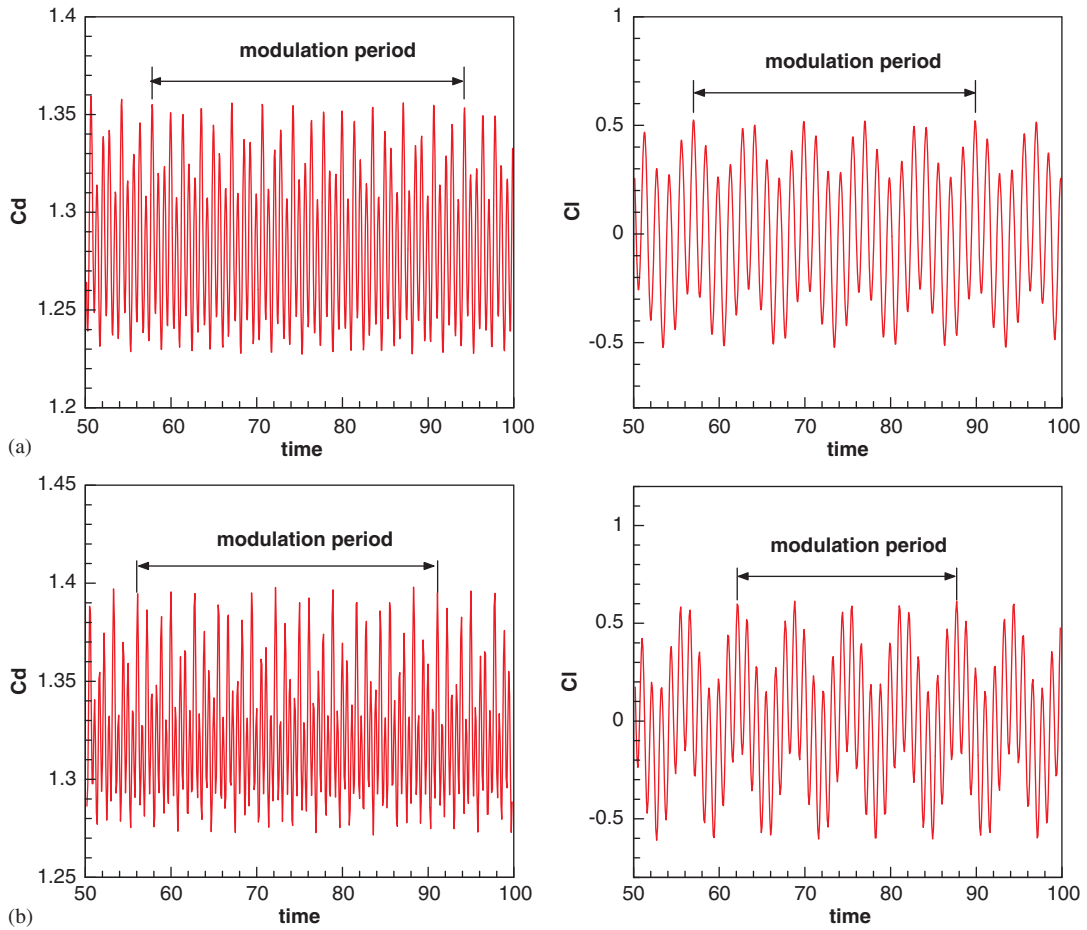


Figure 7. Time histories of the lift and drag coefficients for two non-lock-on cases: (a)  $St_f = 0.7$  and (b)  $St_f = 0.9$ .

where  $x(t), y(t)$  is the position of the center of wing,  $\alpha(t)$  is the wing orientation that is measured counterclockwise relative to the positive  $x$ -axis. The parameters also include the stroke amplitude  $A_0$ , the initial angle of attack  $\alpha_0$ , the amplitude of pitching angle of attack  $\beta$ , the frequency  $f$ , and the phase difference  $\phi$  between  $x(t)$  and  $a(t)$ . For the motion simulated in this work, the Reynolds number can be defined as  $Re = U_{\max} c / \nu = \pi f A_0 c / \nu$ , where  $U_{\max}$  is the maximum wing velocity and  $c$  is the chord. In current simulation,  $A_0/c = 2.8$  with  $Re = 75$  is used. Other parameters  $\alpha_0, \beta$ , and  $f$  are fixed to be  $\pi/2, \pi/4$ , and 0.25 Hz, respectively. Based on previous investigations, it is well known that the phase difference  $\phi$  between rotation and translation is a critical parameter in force generation. Here, two phase differences are selected:  $\phi = \pi/4$  and  $-\pi/4$ , corresponding to the advanced and delayed rotation, respectively. To make a fair comparison, all the above parameters are chosen the same as those used by Wang *et al.* [41] and Eldredge [42]. The computational domain of  $20c \times 20c$  is discretized by a non-uniform mesh. The region around the center of domain for the wing motion is  $3.5c \times 1.2c$  with a very fine uniform mesh size of  $281 \times 97$ .

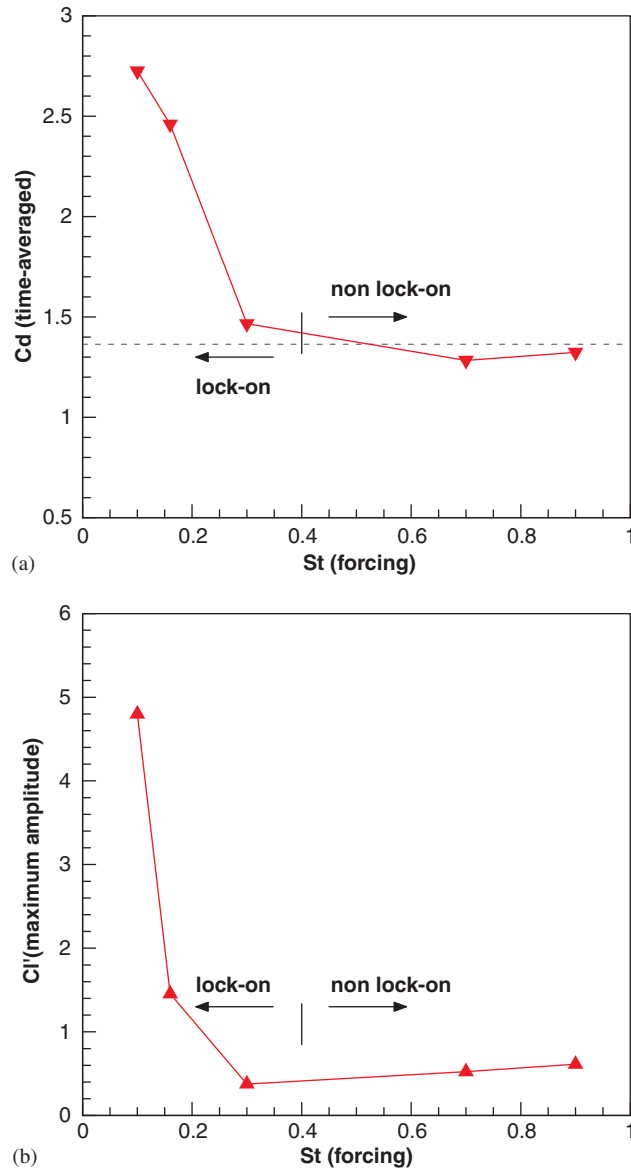


Figure 8. Variations of drag and lift coefficients versus  $St_f$ : (a) time-averaged drag coefficients (dash line: the value for stationary cylinder case) and (b) maximum amplitude of the lift coefficient fluctuation.

Figure 9 shows four snapshots of vorticity contours given in one cycle for advanced rotation of  $\phi = \pi/4$ . They are very similar to those obtained by Eldredge [42], which gave the physical interpretation.

The time histories of lift and drag coefficients are plotted in Figure 10. The previous numerical and experimental results [41, 42] are also included for comparison. All forces are normalized by the maxima of the corresponding quasi-steady forces that are mentioned in [41]:  $C_L = 1.2 \sin(2\alpha)$ ,



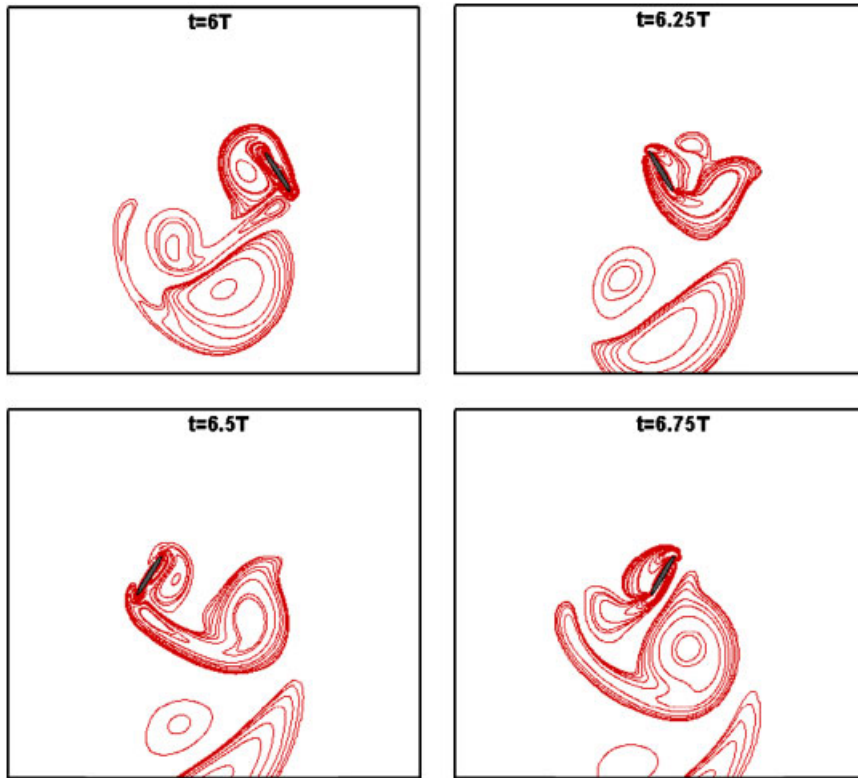
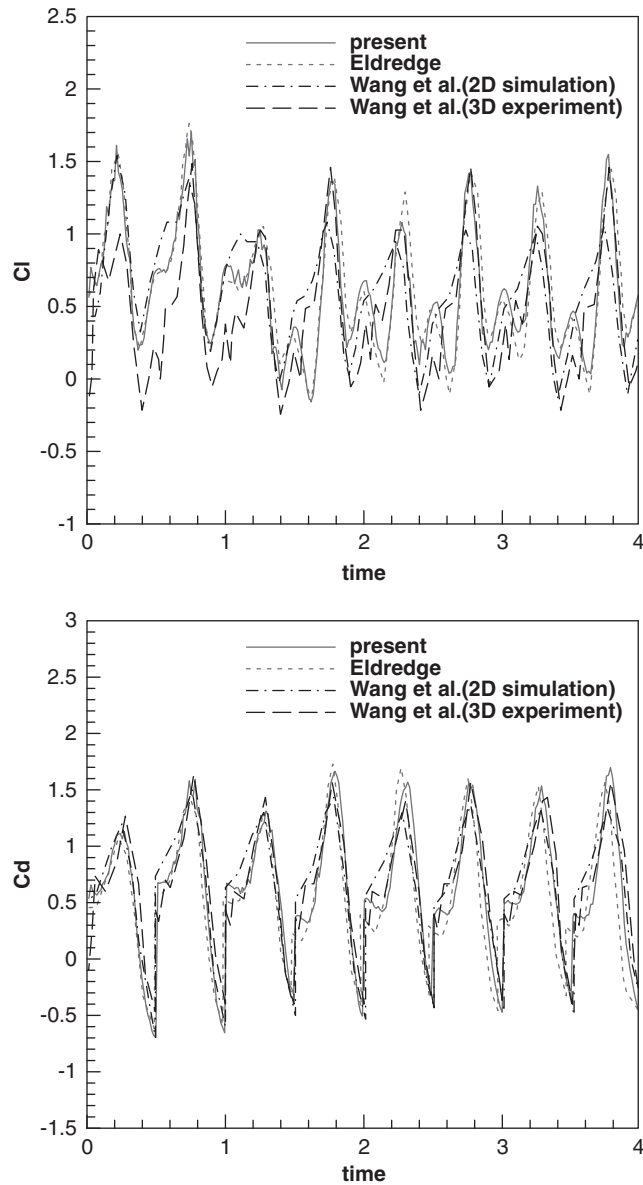


Figure 9. Vorticity contours in one cycle for  $\phi = \pi/4$ .

$C_D = 1.4 - \cos(2\alpha)$ , where  $\alpha$  is the angle of attack. From Figure 10, we can see that the present lift coefficient matches very well with the result of Eldredge [42], and the peak values are in good agreement with those from Wang *et al.* [41] in the whole four strokes. In the later strokes, the peak values of lift also compare well with experimental results of [41]. The present drag coefficients also agree very well with those given in [41, 42]. Figure 11 depicts the vorticity contours for the delayed rotation case of  $\phi = -\pi/4$ . By comparing the results of  $\phi = \pi/4$  with those of  $\phi = -\pi/4$ , the sensitivity to the kinematics is obvious. Similar to the advanced rotation case, the behavior of leading edge vortex is recaptured and trailing edge vortex shedding is also observed. The instantaneous lift and drag coefficients are plotted in Figure 12 and are compared with those of Wang *et al.* [41]. It can be found that the agreement is reasonable. By comparing Figure 10 with Figure 12, we can find that the characteristics of lift and drag are very different. The time-averaged lift and drag coefficients are (0.526, 0.567) and (0.101, 0.721), respectively, for advanced and delayed rotation cases. From these values, we can conclude that the phase difference  $\phi$  affects the force generation significantly, especially for the lift production.

### 3.5. Flow over a flapping flexible airfoil

The numerical solution of fluid flow problems with thin flexible moving objects is motivated by the wide range of potential applications in biology and physiology. In comparison with

Figure 10. Time histories of lift and drag coefficients for  $\phi = \pi/4$ .

rigid airfoils, the flapping flexible airfoils may generate better aerodynamic performance under certain conditions. Recently, Miao and Ho [43] investigated the effect of flexure on aerodynamic propulsive efficiency of flapping flexible airfoil. To exhibit the capability of our method for modeling such kind of problems, the simulation of a single flapping flexible airfoil is carried out.

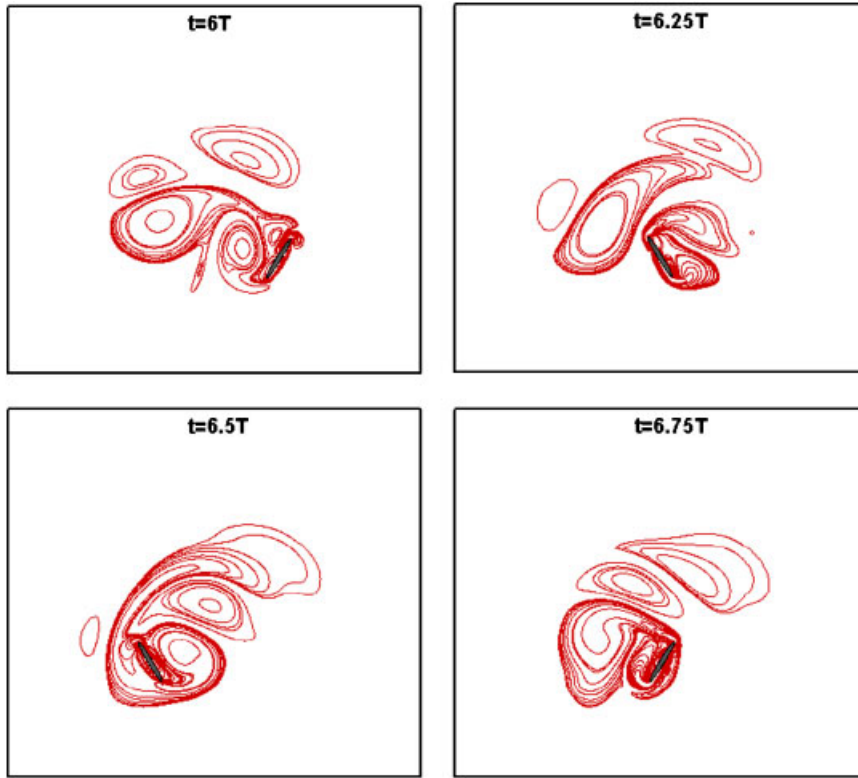


Figure 11. Vorticity contours in one cycle for  $\phi = -\pi/4$ .

In our simulation, a single NACA0012 airfoil with flexural deformation executes plunge motion, which means that the airfoil makes the cross flow oscillation. The plunge motion of airfoil shown in Figure 13 can be expressed by

$$h = h_0 c \cos(\omega t) \quad (34)$$

where  $h$  means the instantaneous position of the airfoil,  $h_0$  is the dimensionless stroke amplitude,  $c$  denotes the chord length of the airfoil, and  $\omega$  is the flapping frequency.

Inspired by the hydrodynamics of fish-like swimming, the profile of the flexible airfoil varying over time can be expressed by

$$y = -a_0 c x^2 \cos(\omega t + \phi) \quad (35)$$

where  $a_0$  is the dimensionless flexure amplitude and  $\phi$  denotes the phase angle. In the above equation, the  $x - y$  local frame refers to the body coordinate system and  $x$  is in the range of  $[0, c]$  with  $x=0$  corresponding to the head of airfoil and  $x=c$  to the tail.

In the study of Miao and Ho [43], they found that the thrust-indicative wake structures were generated only when  $a_0$  is less than 0.5. It means that when  $a_0$  is larger than 0.5, the flexible structures could not enhance aerodynamic performance. In the current simulation, the parameters for controlling the airfoil motion are chosen as:  $h_0=0.4$ ,  $a_0=0.3$ ,  $\phi=\pi/2$ , and  $\omega=0.2, 0.3, 0.4, 0.5$ .

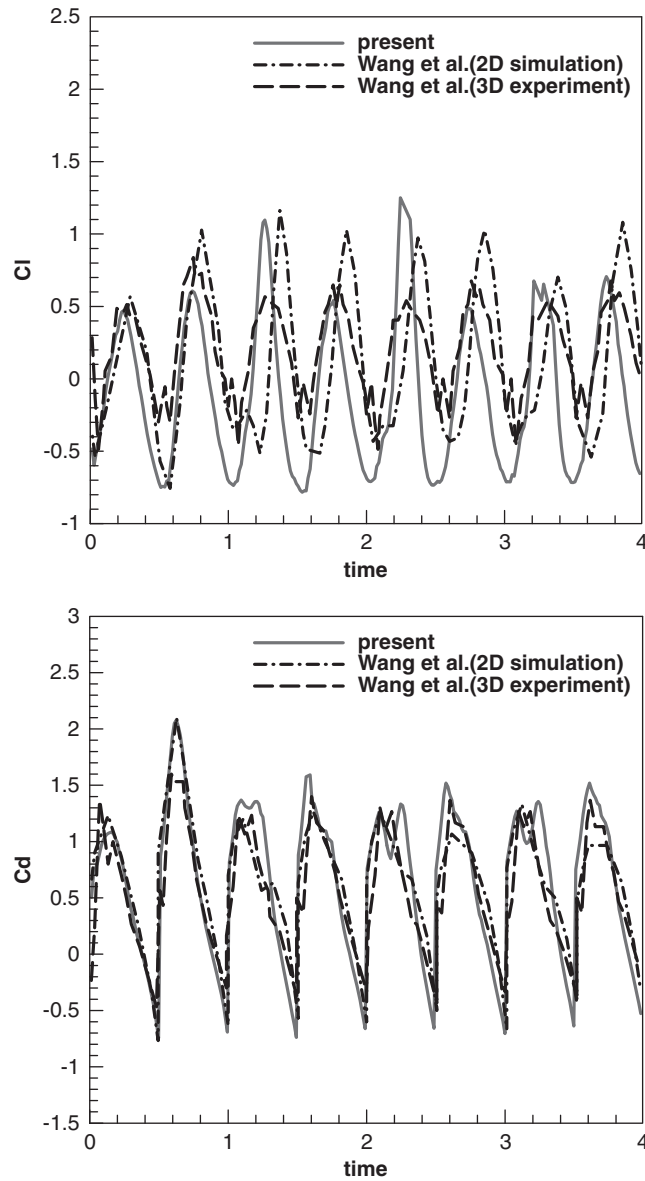


Figure 12. Time histories of lift and drag coefficients for  $\phi = -\pi/4$ .

The Reynolds number based on the chord length  $c$  is  $Re = 500$ . A non-uniform mesh is used in the whole computational domain of  $30c \times 24c$ . In the small region around the airfoil ( $1.1c \times 1.5c$ ) where airfoil is moved, a very fine uniform mesh size of  $221 \times 301$  is applied.

The instantaneous vorticity contours in one cycle and evolution of drag coefficient are plotted in Figures 14 and 15, respectively. For comparison, the results of flapping rigid airfoil with the same

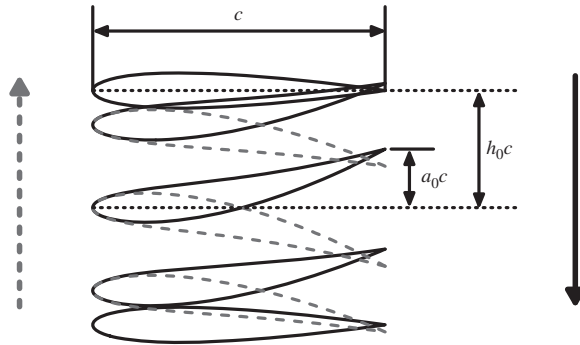


Figure 13. Plunge and deflection motion of a single flexible airfoil (solid line: downstroke motion; dashed line: upstroke motion).

stroke amplitude are also included. In both cases, the flapping frequency  $\omega$  is 0.4. From Figure 14, we can observe the vortex being shed from the trailing edge of airfoil due to oscillation of the airfoil. Subsequently, the periodic vortex shedding induces the drag coefficient to vary periodically, which can be observed in Figure 15. We can also find that the drag coefficient becomes negative partly, which implies generation of a propulsive force. Figure 14 reveals that the flow patterns of two cases are almost the same except that the wavelength of vortex for rigid airfoil is a bit longer than that of flexible airfoil. One reason may be that the flexible airfoil makes the vortex shed more easily, implying that the vortex shedding becomes faster. Additionally, it is noticeable in Figure 15 that the maximum of negative drag coefficient for flexible airfoil is much larger than that of rigid airfoil, and that more negative drag region appears for flapping flexible airfoil. Therefore, it may be concluded that the flexure could effectively augment the propulsive force for flapping airfoil.

Figure 16 presents the evolution of the pressure contours around the rigid and flexible airfoils during one cycle. The corresponding results for the lift coefficients are plotted in Figure 17. As can be seen from Figure 16, the pressure in the wake of flexible airfoil is higher than that of rigid airfoil. As a result, the pressure contributes less drag force for the flexible airfoil case. This is in line with the findings in [43].

To demonstrate the effect of flapping frequency on the generation of thrust force, the evolution of drag coefficients with four different flapping frequencies is depicted in Figure 18. It is clear that the maximum negative drag coefficient increases with the flapping frequency. Higher frequency implies more generation of propulsive force.

Let  $T$  denote the flapping period, the period-averaged thrust force  $\bar{F}_x$  can be calculated by

$$\bar{F}_x = \frac{1}{T} \int_0^T F_x(t) dt \quad (36)$$

where  $F_x(t)$  represents the instantaneous thrust force on the airfoil, which is equal to the negative drag force. Therefore, we can define the period-averaged thrust power coefficient  $\xi$  as

$$\xi = \frac{\bar{F}_x U_\infty}{(\frac{1}{2} \rho U_\infty^2 c) U_\infty} = -\frac{1}{T} \int_0^T C_d dt \quad (37)$$

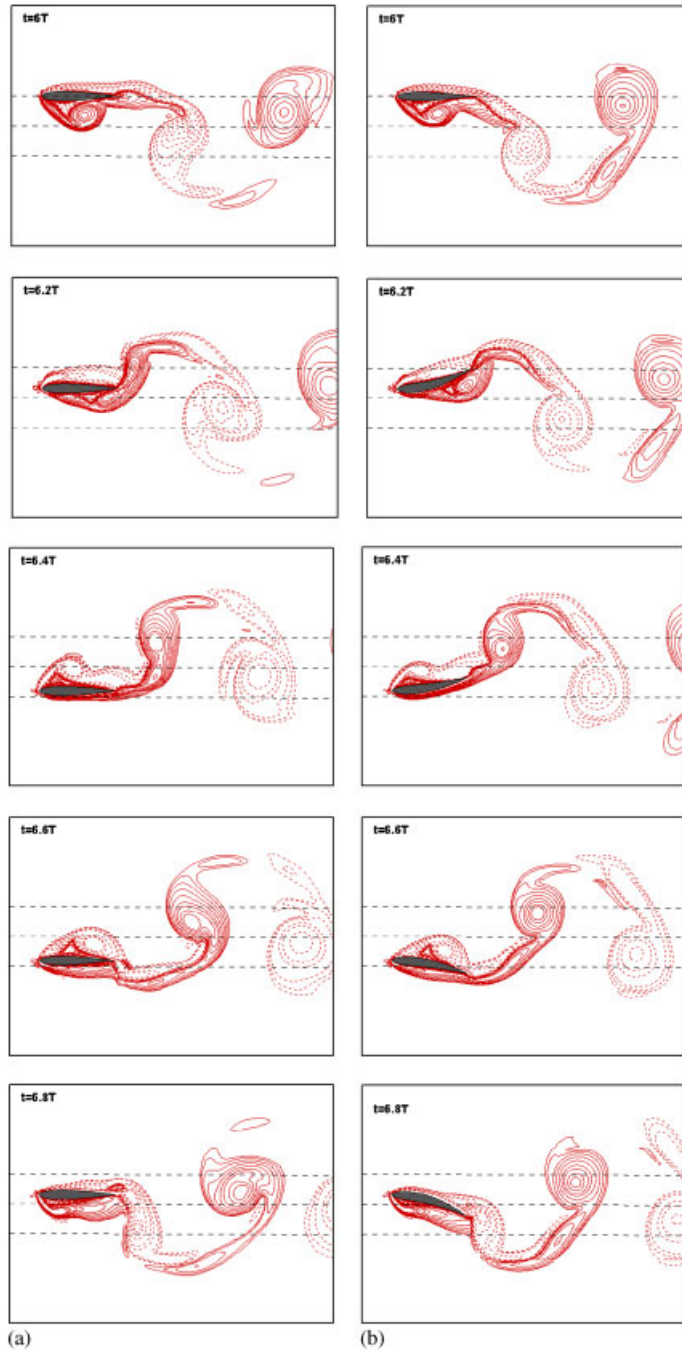


Figure 14. Instantaneous vorticity contours for flapping airfoil in one cycle:  
(a) rigid airfoil and (b) flexible airfoil.

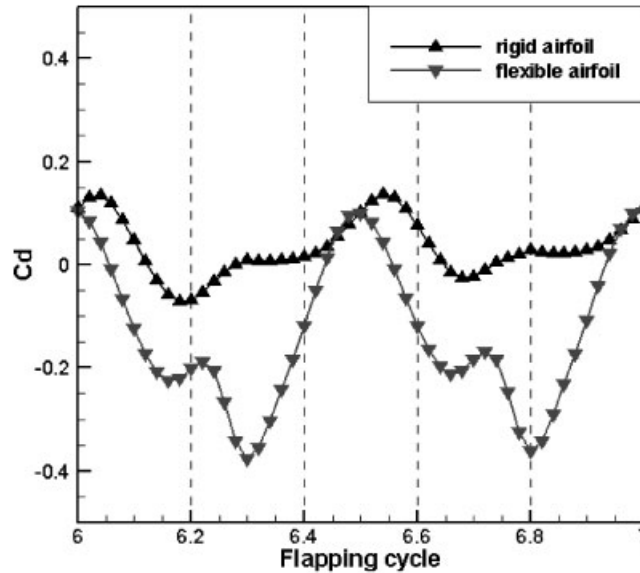


Figure 15. The evolution of drag coefficients for rigid and flexible airfoils.

Figure 19 shows the variation of the thrust power coefficient with the flapping frequency. It can be seen that the thrust power coefficient is greater at larger flapping frequency. Hence, same conclusion with that from Figure 18 can be obtained.

#### 4. CONCLUSIONS

A new version of immersed boundary-lattice Boltzmann method (IB-LBM) is presented in this paper for simulating incompressible viscous flows around moving objects. In the conventional IB-LBM, the force density is calculated explicitly by Hook's law or the direct forcing method or the momentum exchange method. Therefore, the non-slip boundary condition is only approximately satisfied. In this study, the force density, which can be recognized as the velocity correction, is set as unknown. It is solved by enforcing the non-slip boundary condition. Moreover, the lift and drag forces on the moving object can be easily calculated via the velocity correction on the boundary points. The lattice Boltzmann equation with a force density term is adopted in this work to obtain the flow field at the Eulerian points.

To show that the present method does not have any flow penetration to the solid boundary and provides better results for the forces acting on the object, the steady flow over a stationary circular cylinder was first simulated. Numerical results do confirm our expectation. For moving boundary flow problems, the simulation of flows around a moving circular cylinder, a rotationally oscillating cylinder, and an elliptic flapping wing was performed. The obtained numerical results are compared well with available data in the literature. In addition, to demonstrate the capability of the present method for solving elastic boundary flow problems, the flow around a flexible flapping airfoil was simulated. It was found that as compared with the flapping rigid airfoil, the flapping flexible

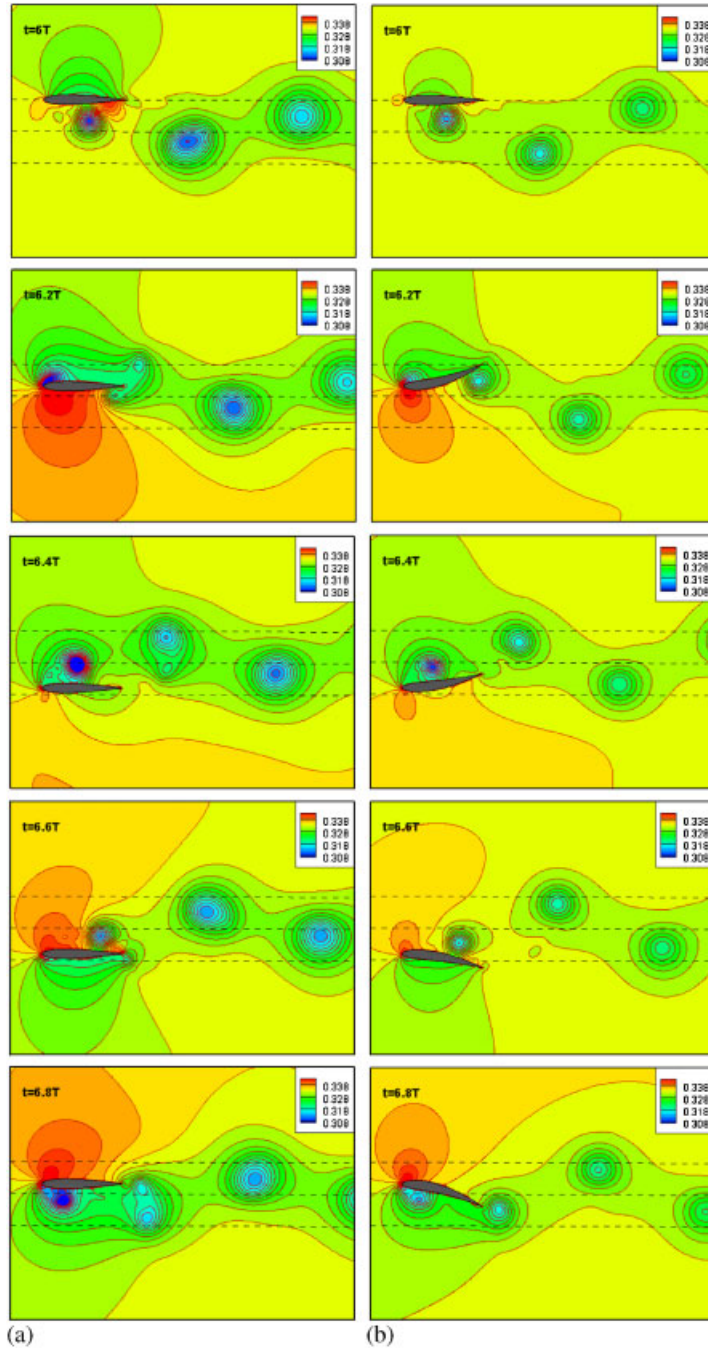


Figure 16. The pressure contours for flapping airfoil in one cycle: (a) rigid airfoil and (b) flexible airfoil.



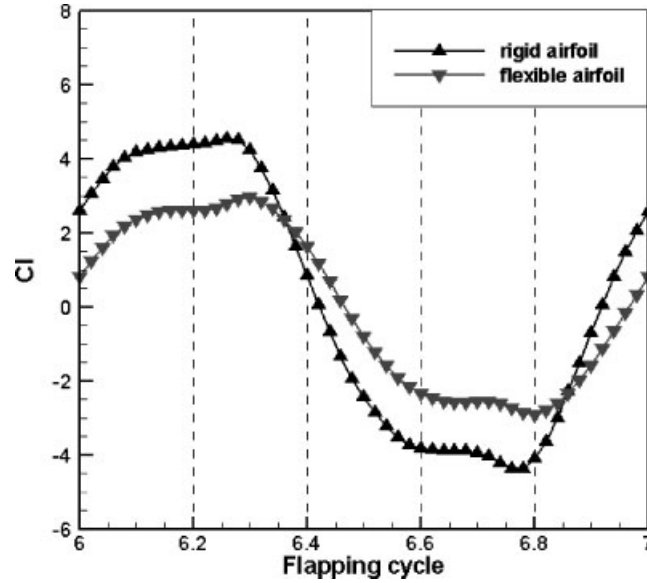


Figure 17. The evolution of lift coefficients for rigid and flexible airfoils.

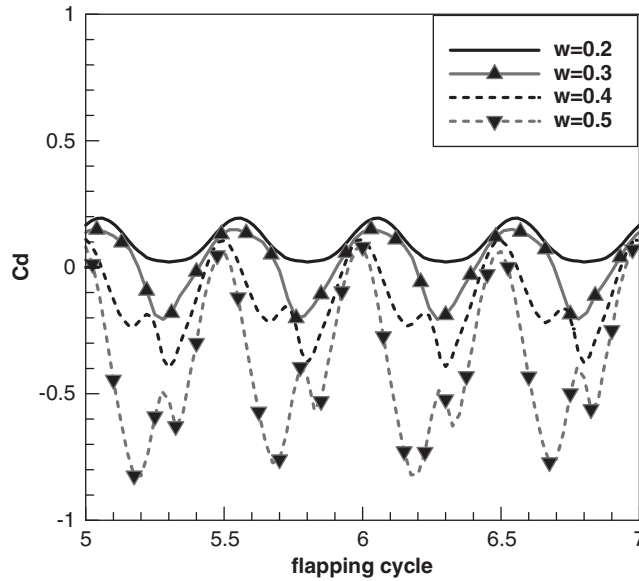


Figure 18. The evolution of drag coefficients with different frequencies.

airfoil can easily generate the propulsive force. Through numerical experiments, it is believed that the present method has a potential to effectively simulate incompressible viscous flows around moving objects. Owing to simplicity of the approach, the present method could be easily extended

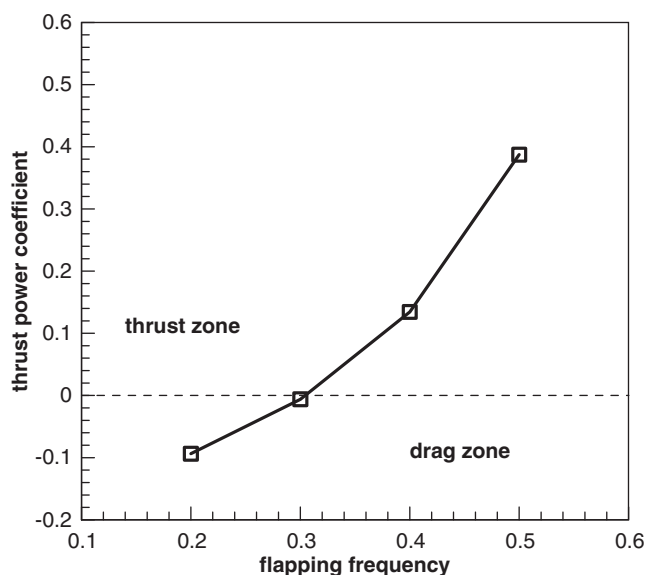


Figure 19. Effect of flapping frequency on thrust power coefficient.

to simulate three-dimensional incompressible viscous flows around complex geometries and/or moving objects.

#### ACKNOWLEDGEMENTS

This work was partially supported by the National Natural Science Foundation of China (10728206).

#### REFERENCES

1. Tsiveriotis K, Brown RA. Boundary-conforming mapping applied to computations of highly deformed solidification interfaces. *International Journal for Numerical Methods in Fluids* 1992; **14**:981–1003.
2. Mateescu D, Mekanik A, Païdoussis MP. Analysis of 2-D and 3-D unsteady annular flows with oscillating boundaries, based on a time-dependent coordinate transformation. *Journal of Fluids and Structures* 1996; **10**: 57–77.
3. Dimakopoulos Y, Tsamopoulos J. A quasi-elliptic transformation for moving boundary problems with large anisotropic deformations. *Journal of Computational Physics* 2003; **192**:494–522.
4. Luo H, Bewley TR. On the contravariant form of the Navier–Stokes equations in time-dependent curvilinear coordinate systems. *Journal of Computational Physics* 2004; **199**:355–375.
5. Li R, Tang T, Zhang PW. A moving mesh finite element algorithm for singular problems in two and three space dimensions. *Journal of Computational Physics* 2002; **177**:365–393.
6. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *Journal of Computational Physics* 1974; **14**:227–253.
7. Hu HH, Patankar NA, Zhu MY. Direct numerical simulations of fluid–solid systems using arbitrary Lagrangian–Eulerian technique. *Journal of Computational Physics* 2001; **169**:427–462.
8. Sarrate J, Huerta A, Donea J. Arbitrary Lagrangian–Eulerian formulation for fluid–rigid body interaction. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3171–3188.
9. Anderson RW, Elliott NS, Pember RB. An arbitrary Lagrangian–Eulerian method with adaptive mesh refinement for the solution of the Euler equations. *Journal of Computational Physics* 2004; **199**:598–617.

10. Li J, Hesse M, Ziegler J, Woods AW. An arbitrary Lagrangian Eulerian method for moving-boundary problems and its application to jumping over water. *Journal of Computational Physics* 2005; **208**:289–314.
11. Chew CS, Yeo KS, Shu C. A generalized finite difference (GFD) ALE scheme for incompressible flows around moving solid bodies on hybrid mesh free-Cartesian grids. *Journal of Computational Physics* 2006; **218**:510–548.
12. Russell D, Wang ZJ. A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow. *Journal of Computational Physics* 2003; **191**:177–205.
13. Quirk JJ. An alternative to unstructured grids for computing gas dynamic flows around arbitrary complex two-dimensional bodies. *Computers and Fluids* 1994; **23**:125–142.
14. Udaykumar HS, Shyy W, Rao MM. Elafint: a mixed Eulerian–Lagrangian method for fluid flows with complex and moving boundaries. *International Journal for Numerical Methods in Fluids* 1996; **22**:691–705.
15. Ye T, Mittal R, Udaykumar HS, Shyy W. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics* 1999; **156**:209–240.
16. Kirkpatrick MP, Armfield SW, Kent JH. A representation of curved boundaries for the solution of the Navier–Stokes equations on a staggered three-dimensional Cartesian grid. *Journal of Computational Physics* 2003; **184**:1–36.
17. Zhou CH, Shu C, Wu YZ. Extension of domain-free discretization method to simulate compressible flows over fixed and moving bodies. *International Journal for Numerical Methods in Fluids* 2007; **53**:175–199.
18. Peskin CS. Numerical analysis of blood flow in the heart. *Journal of Computational Physics* 1977; **25**:220–252.
19. Goldstein D, Handler R, Sirovich L. Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics* 1993; **105**:354–366.
20. Goldstein D, Handler R, Sirovich L. Direct numerical simulation of turbulent flow over a modeled riblet covered surface. *Journal of Fluid Mechanics* 1995; **302**:333–376.
21. Kim J, Kim D, Choi H. An immersed boundary finite volume method for simulations of flow in complex geometries. *Journal of Computational Physics* 2001; **171**:131–150.
22. Lima ALF, Silva E, Silveira-Neto A, Damasceno JJR. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. *Journal of Computational Physics* 2003; **189**:351–370.
23. Chen S, Doolen GD. Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics* 1998; **30**:329–364.
24. Feng ZG, Michaelides EE. The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. *Journal of Computational Physics* 2004; **195**:602–628.
25. Feng ZG, Michaelides EE. Proteus: a direct forcing method in the simulations of particulate flows. *Journal of Computational Physics* 2005; **202**:20–51.
26. Peng Y, Shu C, Chew YT, Niu XD, Lu XY. Application of multi-block approach in the immersed boundary-Lattice Boltzmann method for viscous fluid flows. *Journal of Computational Physics* 2006; **218**:460–478.
27. Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics* 2000; **161**:35–60.
28. Niu XD, Shu C, Chew YT, Peng Y. A momentum exchanged-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows. *Physics Letters A* 2006; **354**:173–182.
29. Shu C, Liu NY, Chew YT. A novel immersed boundary velocity correction-Lattice Boltzmann method and its application to simulate flow past a circular cylinder. *Journal of Computational Physics* 2007; **226**:1607–1622.
30. Guo Z, Zheng C, Shi B. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Physical Review E* 2002; **65**:046308.
31. Qian YH, d’Humières D, Lallemand P. Lattice BGK model for Navier–Stokes equation. *Europhysics Letters* 1992; **17**:479–484.
32. Peskin CS. The immersed boundary method. *Acta Numerica* 2002; **11**:479–517.
33. Lai MC, Peskin CS. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics* 2000; **160**:705–719.
34. Shu C, Niu XD, Chew YT. Taylor-series expansion and least-squares-based lattice Boltzmann method: two-dimensional formulation and its applications. *Physical Review E* 2002; **65**:036708.
35. Dennise SCR, Chang GZ. Numerical solutions for steady flow past a circular cylinder at Reynolds number up to 100. *Journal of Fluid Mechanics* 1970; **42**:471–489.
36. Nieuwstadt F, Keller HB. Viscous flow past circular cylinders. *Computers and Fluids* 1973; **1**:59–71.
37. Fornberg B. A numerical study of steady viscous flow past a circular cylinder. *Journal of Fluid Mechanics* 1980; **98**:819–855.
38. Shukla RK, Tatineni M, Zhong X. Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier–Stokes equations. *Journal of Computational Physics* 2007; **224**:1064–1094.

39. Choi S, Choi H, Kang S. Characteristics of flow over a rotationally oscillating cylinder at low Reynolds number. *Physics of Fluids* 2002; **14**:2767–2777.
40. Protas B, Wesfreid JE. Drag force in the open-loop control of the cylinder wake in the laminar regime. *Physics of Fluids* 2002; **14**:810–826.
41. Wang ZJ, Birch JM, Dickinson MH. Unsteady forces and flows in low Reynolds number hovering flight: two-dimensional computations vs robotic wing experiments. *Journal of Experimental Biology* 2004; **207**:449–460.
42. Eldredge JD. Numerical simulation of the fluid dynamics of 2D rigid body motion with the vortex particle method. *Journal of Computational Physics* 2007; **221**:626–648.
43. Miao JM, Ho MH. Effect of flexure on aerodynamic propulsive efficiency of flapping flexible airfoil. *Journal of Fluids and Structures* 2006; **22**:401–419.