Simulation of Incompressible Viscous Flows around Moving Objects by A Variant of Immersed Boundary-Lattice Boltzmann Method

J. Wu, C. Shu[1]

Department of Mechanical Engineering, National University of Singapore

10 Kent Ridge Crescent, Singapore 119260

and

Y. H. Zhang

Department of Mechanical Engineering,

University of Strathclyde, Glasgow, G1 1XJ, UK

Abstract

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 the 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 present method for moving objects is well demonstrated through its application to

---

[1] Corresponding author. Tel: (65) 6516 6476. Fax: (65) 6779 1459. Email: mpeshuc@nus.edu.sg

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 present method for implementing the elastic boundary condition. It was found that the flapping flexible airfoil can generate larger propulsive force than the flapping rigid airfoil.

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 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. Due to regenerating the mesh to conform to the boundary at all times, it becomes difficult for 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 to 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. Like the IBM, the

standard LBM is usually applied on the Cartesian mesh. Due to this common feature, it is desirable to combine these two methods together. Many efforts [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 the 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 has been 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. 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. Due 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 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

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

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

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

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

where $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{f}$ are the Eulerian coordinate, fluid velocity and force density acting on the fluid, respectively, p is the fluid pressure, ρ is the fluid density and μ 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)-(2) are the N-S equations with external force. Equations (3)-(4) describe the interaction between the immersed boundary and the fluid flow by distributing the boundary force at the Lagrangian points to Eulerian points and interpolating the velocity at the Eulerian points to Lagrangian points. The calculation of boundary force density **F,** which is also called restoring force, is critical in the IBM. Using Hooke's law, it can be determined by

$$\mathbf{F}(s,t) = -k\Delta\xi = -k\left(\mathbf{V}_{fluid}\Delta t - \mathbf{V}_{wall}\Delta t\right) \qquad (5)$$

where $\mathbf{V}_{fluid}$ is the fluid velocity at the boundary point interpolated from the surrounding fluid (Eulerian) points, $\mathbf{V}_{wall}$ 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 $\mathbf{F}$, 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 immersed boundary-lattice Boltzmann (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

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

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

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

where f_α is the distribution function, f_α^{eq} is its corresponding equilibrium state, τ is the single relaxation time, $\mathbf{e}_\alpha$ is the particle velocity, w_α are coefficients which 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

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

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

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

with $w_0 = 4/9$, $w_1 = w_2 = w_3 = w_4 = 1/9$ and $w_5 = w_6 = w_7 = w_8 = 1/36$. $c_s = c/\sqrt{3}$ is the sound speed of this model. The relation between the relaxation time and the kinematic viscosity of fluid is $\upsilon = \left(\tau - \frac{1}{2}\right)c_s^2 \delta t$. If we define the intermediate velocity $\boldsymbol{u}^*$ as

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

and the velocity correction as

$$\rho \delta \mathbf{u} = \frac{1}{2} \mathbf{f} \delta t \tag{12}$$

then equation (8) can be written as

$$\mathbf{u} = \mathbf{u}^* + \delta\mathbf{u} \tag{13}$$

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, 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 Fig. 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 $\mathbf{X}_B(s_l, t), l = 1, 2, \cdots, m$. Here, we can set an unknown velocity correction vector $\delta\mathbf{u}_B^l$ at every Lagrangian point. The velocity correction $\delta\mathbf{u}$ at the Eulerian point can be obtained by the following Dirac delta function interpolation

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

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

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

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

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

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

$$\delta\mathbf{u}\left(\mathbf{x}_{ij},t\right)=\sum_{l}\delta\mathbf{u}_{B}^{l}\left(\mathbf{X}_{B}^{l},t\right)D_{ij}\left(\mathbf{x}_{ij}-\mathbf{X}_{B}^{l}\right)\Delta s_{l}\qquad\left(l=1,2,\cdots,m\right)\tag{17}$$

where Δs_{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

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

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

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

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

$$\begin{aligned}\mathbf{U}_{B}^{l}\left(\mathbf{X}_{B}^{l},t\right)=&\sum_{i,j}\mathbf{u}^{*}\left(\mathbf{x}_{ij},t\right)D_{ij}\left(\mathbf{x}_{ij}-\mathbf{X}_{B}^{l}\right)\Delta x\Delta y\\&+\sum_{i,j}\sum_{l}\delta\mathbf{u}_{B}^{l}\left(\mathbf{X}_{B}^{l},t\right)D_{ij}\left(\mathbf{x}_{ij}-\mathbf{X}_{B}^{l}\right)\Delta s_{l}D_{ij}\left(\mathbf{x}_{ij}-\mathbf{X}_{B}^{l}\right)\Delta x\Delta y\end{aligned}\tag{20}$$

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

$$\mathbf{AX}=\mathbf{B}\tag{21}$$

where $\mathbf{X}=\left\{\delta\mathbf{u}_{B}^{1},\delta\mathbf{u}_{B}^{2},\cdots,\delta\mathbf{u}_{B}^{m}\right\}^{T}$; $\mathbf{B}=\left\{\Delta\mathbf{u}_{1},\Delta\mathbf{u}_{2},\cdots,\Delta\mathbf{u}_{m}\right\}^{T}$ with

$$\Delta\mathbf{u}_{l}=\mathbf{U}_{B}^{l}\left(\mathbf{X}_{B}^{l},t\right)-\sum_{i,j}\mathbf{u}^{*}\left(\mathbf{x}_{ij},t\right)D_{ij}\left(\mathbf{x}_{ij}-\mathbf{X}_{B}^{l}\right)\Delta x\Delta y\quad\left(l=1,2,\cdots,m\right)\tag{22}$$

Note that the elements of matrix $\mathbf{A}$ are only related to the boundary points and their nearby Eulerian points (See Fig. 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

$$\rho = \sum_\alpha f_\alpha, \qquad P = c_s^2 \rho \qquad\qquad (23)$$

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

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

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 macro 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

$$\text{Re} = \frac{U_\infty D}{\upsilon} \tag{25}$$

where U_∞ is the free stream velocity, D is the diameter of cylinder, and υ 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

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

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

$$F_D = -\int_\Omega f_x d\mathbf{x} \tag{27}$$

f_x stands for the x-component of force density $\mathbf{f}$ at the boundary point.

The simulations at $Re = 20$ and 40 are carried out. The computational domain is set by $50D \times 50D$ with a mesh size of 451×401. 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 should be used around 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. The region around the cylinder is $1.2D \times 1.2D$ with a very fine uniform mesh size of 97×97.

The streamlines are plotted in Fig. 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. 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 1 compares the length of recirculation bubbles L (based on the diameter of cylinder D) and drag coefficients C_d for two cases with previous data [35-37]. 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 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 1281×1281. The stationary cylinder is located at $(8D, 16D)$. The moving cylinder moves towards the left from the position of $(30D, 16D)$. The Reynolds number for both case 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_{∞} onto the velocity in the Eulerian mesh. The adjusted streamlines are shown in Fig. 3. For comparison, the streamlines of stationary cylinder case are also shown in Fig. 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 Fig. 4, which compares the pressure profiles on the surface of cylinder for two cases. Through this example, on one hand, we demonstrate the capability of present method for simulation of flows around a moving object; on the other hand, we show Galilean invariance of 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, acoustics 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 [38, 39]. 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

$$\gamma(t) = \gamma_0 \sin(2\pi ft) = A\frac{2U_\infty}{D}\sin\left(2\pi \mathrm{St}_f \frac{U_\infty}{D}t\right) \tag{28}$$

where U_∞ and D represent the free stream velocity and diameter of cylinder, respectively. f and γ_0 are the frequency and the rotation amplitude respectively, which can be expressed in terms of non-dimensional parameters: the Strouhal number $\mathrm{St}_f = fD/U_\infty$ 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 $50D \times 40D$ 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 121×121. The whole domain is discretized on the non-uniform mesh. Hence, the TLLBM [34] is applied.

In present simulation, $A = 2.0$ is selected. Based on the free stream velocity U_∞

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 firstly. 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. [38], when $St_f = St_n$, the interesting vortex shedding pattern would happen, which is shown in Fig. 5. Note that the result of Choi et al. [38] is also included in Fig. 5 for comparison. It is clear from the figure that the present result compares well with that of Choi et al [38]. We also simulate the case of $St_f = 0.4$ to compare the forces exerted on the cylinder with that of Choi et al [38]. The obtained time-averaged coefficient and maximum amplitude of lift coefficient fluctuation are 1.302 and 0.321 respectively, which have good agreement with 1.231 and 0.299 from [38].

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 [39]. Fig. 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 Fig. 6 (a)). In Fig. 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 Fig. 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 Fig. 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 Fig. 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. [38], 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 [38], one of the characteristics for the non lock-on region is the occurrence of modulation phenomenon. The modulation frequency is very low as compared to 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 the rotary oscillation forcing. In the present simulation, the mean drag coefficient for the stationary cylinder is about 1.361. It is shown in Fig. 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. The similar trend is found for the maximum amplitude of the lift coefficient fluctuations in Fig. 8 (b).

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 to 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 [40] and [41].

In 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

$$x(t) = \frac{A_0}{2}\cos(2\pi ft), \qquad y(t) = 0 \tag{29}$$

$$\alpha(t) = \alpha_0 + \beta\sin\left(2\pi ft + \phi\right) \tag{30}$$

where $x(t), y(t)$ is the position of the center of wing, $\alpha(t)$ is the wing orientation which is measured counterclockwise relative to the positive x-axis. The parameters also include the stroke amplitude A_0, the initial angle of attack α_0, the amplitude of pitching angle of attack β, the frequency f and the phase difference ϕ between $x(t)$ and $a(t)$. For the motion simulated in this work, the Reynolds number can be defined as $\mathrm{Re} = U_{\max}c/\upsilon = \pi f A_0 c/\upsilon$, 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 α_0, β 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 ϕ

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 above parameters are chosen the same as those used by Wang et al. [40] and Eldredge [41]. 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 281×97.

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 [41], which gave the physical interpretation.

The time histories of lift and drag coefficients are plotted in Fig. 10. The previous numerical and experimental results [40, 41] are also included for comparison. All forces are normalized by the maxima of the corresponding quasi-steady forces which are mentioned in [40]: $C_L = 1.2\sin(2\alpha)$, $C_D = 1.4 - \cos(2\alpha)$, where α is the angle of attack. From Fig. 10, we can see that the present lift coefficient matches very well with the result of Eldredge [41], and the peak values are in good agreement with those from Wang et al. [40] in the whole four strokes. In the later strokes, the peak values of lift also compare well with experimental results of [40]. The present drag coefficients also agree very well with those given in [40-41]. 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 Fig. 12 and are compared with those of Wang et al [40]. It can be found that the agreement is reasonable. By comparing Fig. 10 with Fig. 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 ϕ 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. To exhibit the capability of our method for modeling such problems, the simulation of a single flapping flexible airfoil is carried out.

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 Fig. 13 can be expressed by

$$h = h_0 c \cos(\omega t) \tag{31}$$

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 ω is the flapping frequency.

Inspired by the hydrodynamics of fishlike swimming, the profile of the flexible

airfoil varying over time can be expressed by

$$y = -a_0 c x^2 \cos(\omega t + \phi) \tag{32}$$

where a_0 is the dimensionless flexure amplitude and ϕ 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 current study, 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. 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×24c. In the small region around the airfoil (1.1c×1.5c) where airfoil is moved, a very fine uniform mesh size of 221×301 is applied.

The instantaneous vorticity contours in one cycle and evolution of drag coefficient are plotted in Fig. 14 and 15, respectively. For comparison, the results of flapping rigid airfoil with the same stroke amplitude are also included. In both cases, the flapping frequency ω is 0.4. From Fig. 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 found in Fig. 15. We can also find that the drag coefficient becomes negative partly, which implies generation of a propulsive force. Fig. 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 Fig. 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 Fig. 17. As can be seen from Fig. 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 [42].

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 Fig. 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

$$\bar{F}_x = \frac{1}{T}\int_0^T F_x(t)\,dt \tag{33}$$

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 ξ as

$$\xi = \frac{\bar{F}_x U_\infty}{\left(\frac{1}{2}\rho U_\infty^2 c\right)U_\infty} = -\frac{1}{T}\int_0^T C_d\,dt \tag{34}$$

Fig. 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 Fig. 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 the 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 on 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 present method for solving elastic boundary flow problem, the flow around a flexible flapping airfoil was simulated. It was found that as compared with the flapping rigid

airfoil, the flapping flexible 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.

References

1. K. Tsiveriotis and R. A. Brown, Boundary-Conforming Mapping Applied to Computations of Highly Deformed Solidification Interfaces, *Int. J. Numer. Methods Fluids*, **14**, 981-1003 (1992).

2. D. Mateescu, A. Mekanik, and M. P. Païdoussis, Analysis of 2-D and 3-D Unsteady Annular Flows with Oscillating Boundaries, Based on a Time-Dependent Coordinate Transformation, *J. Fluids Struct.* **10**, 57–77 (1996).

3. Y. Dimakopoulos and J. Tsamopoulos, A Quasi-elliptic Transformation for Moving Boundary Problems with Large Anisotropic Deformations, *J. Comput. Phys*. **192**, 494–522 (2003).

4. H. Luo and T. R. Bewley, On the Contravariant Form of the Navier-Stokes Equations in Time-dependent Curvilinear Coordinate Systems, *J. Comput. Phys*. **199**, 355–375 (2004).

5. R. Li, T. Tang, and P. W. Zhang, A Moving Mesh Finite Element Algorithm for Singular Problems in Two and Three Space Dimensions, *J. Comput. Phys*. **177,** 365–393 (2002).

6. C. W. Hirt, A. A. Amsden, and J. L. Cook, An arbitrary Lagrangian-Eulerian

computing method for all flow speeds, *J. Comput. Phys.* **14**, 227-253 (1974).

7.   H. H. Hu, N. A. Patankar, and M.Y. Zhu, Direct numerical simulations of fluid-solid systems using arbitrary Lagrangian-Eulerian technique, *J. Comput. Phys*. 169, 427-462 (2001).

8.   J. Sarrate, A. Huerta, and J. Donea, Arbitrary Lagrangian-Eulerian formulation for fluid-rigid body interaction, *Comput. Methods Appl. Mech. Engrg*. **190**, 3171-3188, (2001).

9.   R. W. Anderson, N. S. Elliott and R. B. Pember, An Arbitrary Lagrangian-Eulerian Method with Adaptive Mesh Refinement for the Solution of the Euler Equations, *J. Comput. Phys*. **199**, 598–617 (2004).

10. J. Li, M. Hesse, J. Ziegler, and A. W. Woods, An arbitrary Lagrangian Eulerian method for moving-boundary problems and its application to jumping over water, *J. Comput. Phys*. **208**, 289–314 (2005).

11. C. S. Chew, K. S. Yeo, and C. Shu, A Generalized Finite Difference (GFD) ALE Scheme for Incompressible Flows around Moving Solid Bodies on Hybrid Mesh free-Cartesian Grids, *J. Comput. Phys*. **218**, 510-548 (2006).

12. D. Russell and Z. J. Wang, A Cartesian Grid Method for Modeling Multiple Moving Objects in 2D Incompressible Viscous Flow, *J. Comput. Phys*. **191**, 177–205 (2003).

13. J. J. Quirk, An alternative to unstructured grids for computing gas dynamic flows around arbitrary complex two-dimensional bodies, *Comput. Fluids*, **23**, 125-142 (1994).

14. H. S. Udaykumar, W. Shyy, and M. M. Rao, Elafint: a mixed Eulerian-Lagrangian method for fluid flows with complex and moving boundaries, *Int. J. Numer. Methods Fluids*, **22**, 691-705 (1996).

15. T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys*. **156**, 209-240 (1999).

16. M. P. Kirkpatrick, S. W. Armfield, and J. H. Kent, A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional Cartesian grid, *J. Comput. Phys*. **184**, 1-36 (2003).

17. C. H. Zhou, C. Shu, and Y. Z. Wu, Extension of Domain-Free Discretization Method to Simulate Compressible Flows over Fixed and Moving Bodies, *Int. J. Numer. Methods Fluids*, **53**, 175-199 (2007).

18. C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys*. *25*, 220-252 (1977).

19. D. Goldstein, R. Handler, and L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J. Comput. Phys*. **105**, 354-366 (1993).

20. D. Goldstein, R. Handler, and L. Sirovich, Direct numerical simulation of turbulent flow over a modeled riblet covered surface, *J. Fluid Mech*. **302**, 333-376 (1995).

21. J. Kim, D. Kim, and H. Choi, An immersed boundary finite volume method for simulations of flow in complex geometries, *J. Comput. Phys*. **171**, 131-150 (2001).

22. A. 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**, 351-370 (2003).

23. S. Chen and G. D. Doolen, Lattice Boltzmann method for fluid flows, *Ann. Rev. Fluid Mech*. **30**, 329-364 (1998).

24. Z. G. Feng and E. E. Michaelides, The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems, *J. Comput. Phys*. **195**, 602-628 (2004).

25. Z. G. Feng and E. E. Michaelides, Proteus: a direct forcing method in the simulations of particulate flows, *J. Comput. Phys*. **202**, 20-51 (2005).

26. Y. Peng, C. Shu, Y. T. Chew, X. D. Niu, and X. Y. Lu, Application of multi-block approach in the immersed boundary-Lattice Boltzmann method for viscous fluid flows, *J. Comput. Phys*. **218**, 460-478 (2006).

27. E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *J. Comput. Phys*. **161**, 35-60 (2000).

28. X. D. Niu, C. Shu, Y. T. Chew, and Y. Peng, A momentum exchanged-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows, *Phys. Lett. A*, **354**, 173-182 (2006).

29. 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**, 1607-1622 (2007).

30. Z. Guo, C. Zheng, and B. Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, *Phys. Rev. E*, **65**, 046308 (2002).

31. Y. H. Qian, D. d'Humieres, and P. Lallemand, Lattice BGK model for Navier-Stokes equation, *Europhys. Lett.* **17**, 479-484 (1992).

32. C. S. Peskin, The immersed boundary method, *Acta Number.* **11**, 479-517 (2002).

33. M. C. Lai and C. S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* **160**, 705-719 (2000).

34. C. Shu, X. D. Niu, and Y. T. Chew, Taylor-series expansion and least-squares-based lattice Boltzmann method: two-dimensional formulation and its applications, *Phys. Rev. E*, **65**, 036708 (2002).

35. S. C. R. Dennise and G. Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds number up to 100, *J. Fluid Mech.* **42**, 471-489 (1970).

36. F. Nieuwstadt and H. B. Keller, Viscous flow past circular cylinders, *Comput. Fluids.* **1**, 59-71 (1973).

37. R. K. Shukla, M. Tatineni, and X. Zhong, Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier-Stokes equations, *J. Comput. Phys.* **224**, 1064-1094 (2007).

38. S. Choi, H. Choi, and S. Kang, Characteristics of flow over a rotationally oscillating cylinder at low Reynolds number, *Phys. Fluids*, **14**, 2767-2777

(2002).

39. B. Protas and J. E. Wesfreid, Drag force in the open-loop control of the cylinder wake in the laminar regime, *Phys. Fluids*, **14**, 810-826 (2002).

40. Z. J. Wang, J. M. Birch, and M.H. Dickinson, Unsteady forces and flows in low Reynolds number hovering flight: two-dimensional computations *vs* robotic wing experiments, *J. Exp. Biol.* **207**, 449-460 (2004).

41. J. D. Eldredge, Numerical simulation of the fluid dynamics of 2D rigid body motion with the vortex particle method, *J. Comput. Phys.* **221**, 626-648 (2007).

42. J. M. Miao and M. H. Ho, Effect of flexure on aerodynamic propulsive efficiency of flapping flexible airfoil, *J. Fluids Struct.* **22**, 401-419 (2006).

Table 1 Comparison of drag coefficients and length of recirculation zone

Case	Authors	C_d	L
$Re = 20$	Dennis and Chang [35]	2.05	0.94
	Nieuwstadt and Keller [36]	2.053	0.893
	Shukla et al. [37]	2.07	0.92
	Niu et al. [28]	2.144	0.945
	Present	2.072	0.92
$Re = 40$	Dennis and Chang [35]	1.52	2.35
	Nieuwstadt and Keller [36]	1.54	2.18
	Shukla et al. [37]	1.55	2.34
	Niu et al. [28]	1.589	2.26
	Present	1.554	2.3
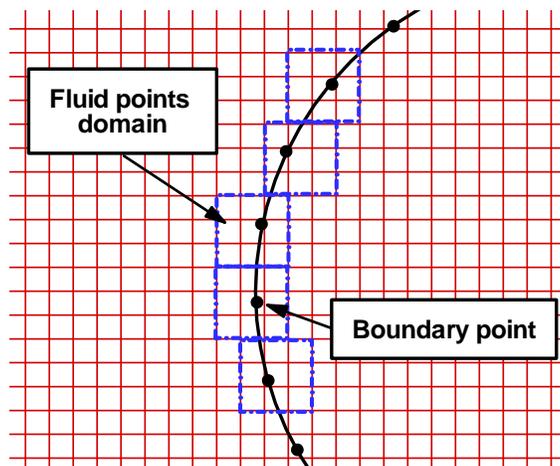
Fig. 1 Configuration of boundary points and their surrounding fluid points
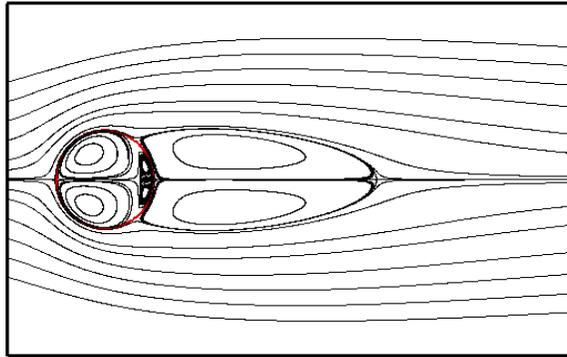
Fig. 2 Streamlines for flow over a stationary cylinder at *Re* = 40
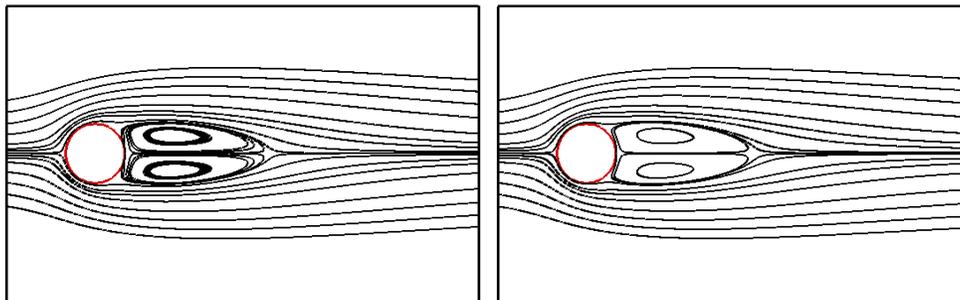
(a) moving cylinder case   (b) stationary cylinder case

Fig. 3 Streamlines for moving and stationary cylinder cases at *Re* = 40
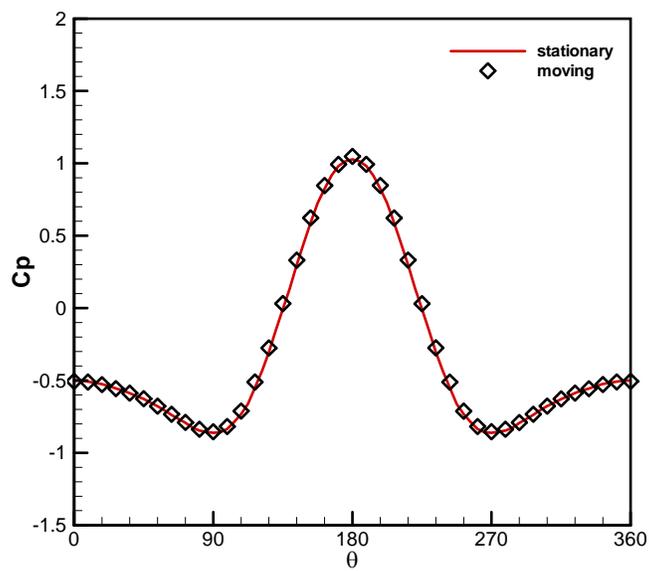
Fig. 4 Pressure distribution on the surface of cylinder at *Re* = 40
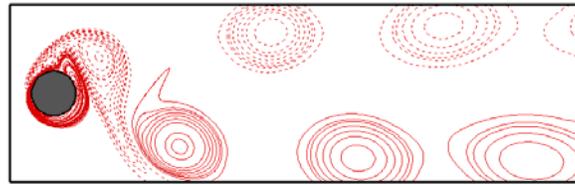
Result of present simulation
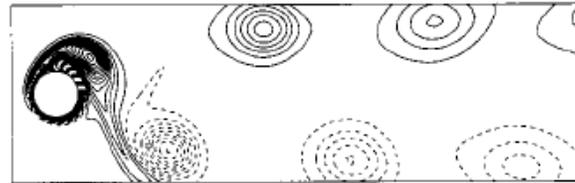
Result of Choi et al. [38]

Fig. 5 Comparison of vorticity contours when St$_f$ = St$_n$ at *Re* = 100
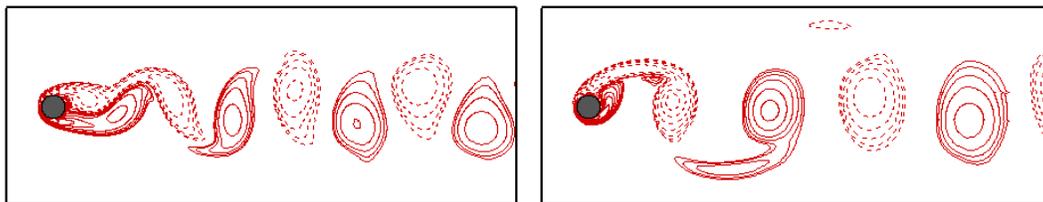
(a) stationary

(b) St$_f$ = 0.1
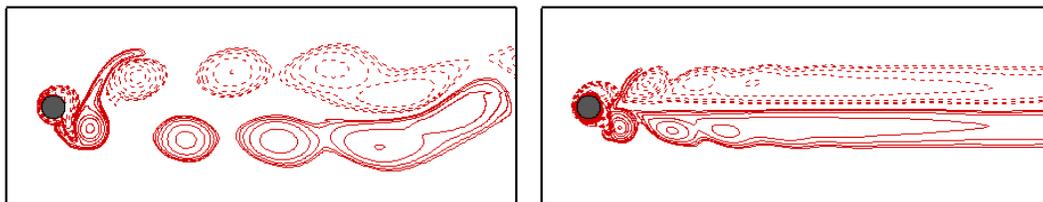
(c) St$_f$ = 0.16

(d) St$_f$ = 0.3
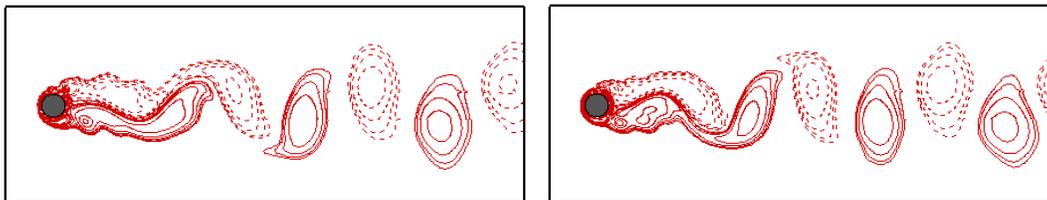
(e) St$_f$ = 0.7

(f) St$_f$ = 0.9

Fig. 6 Vorticity contours at *Re* = 100 for flow around a rotationally oscillating

cylinder (solid lines: positive; dashed lines: negative)
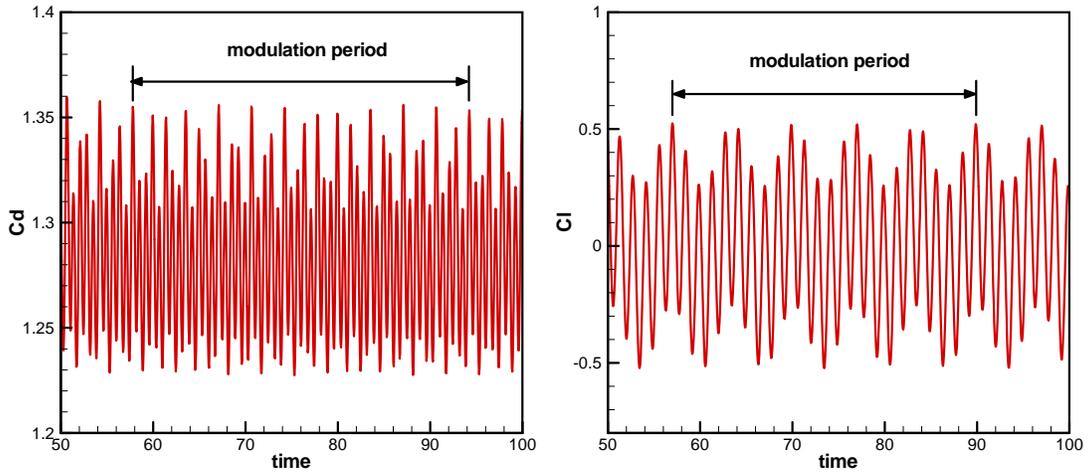
(a) $St_f = 0.7$
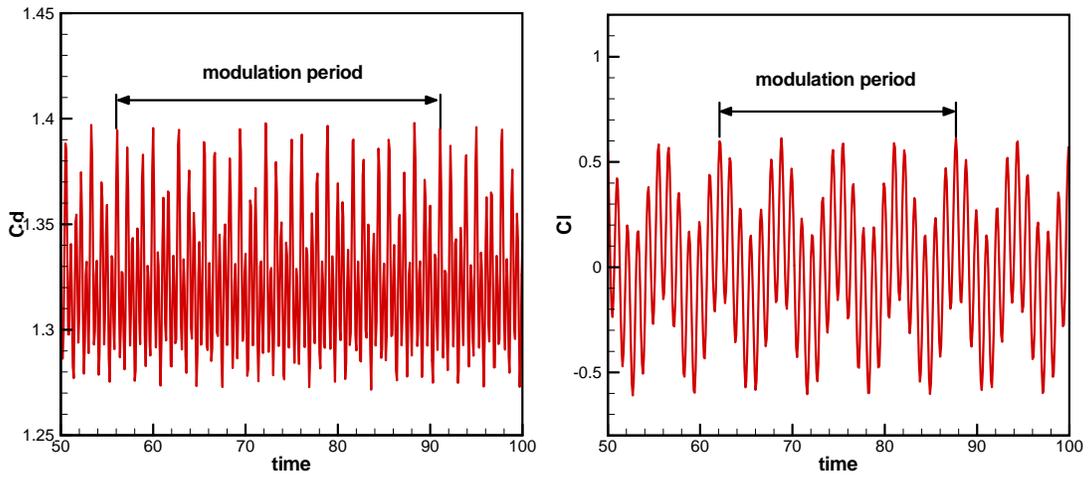
(b) $St_f = 0.9$

Fig. 7 Time histories of the lift and drag coefficients for two non lock-on cases
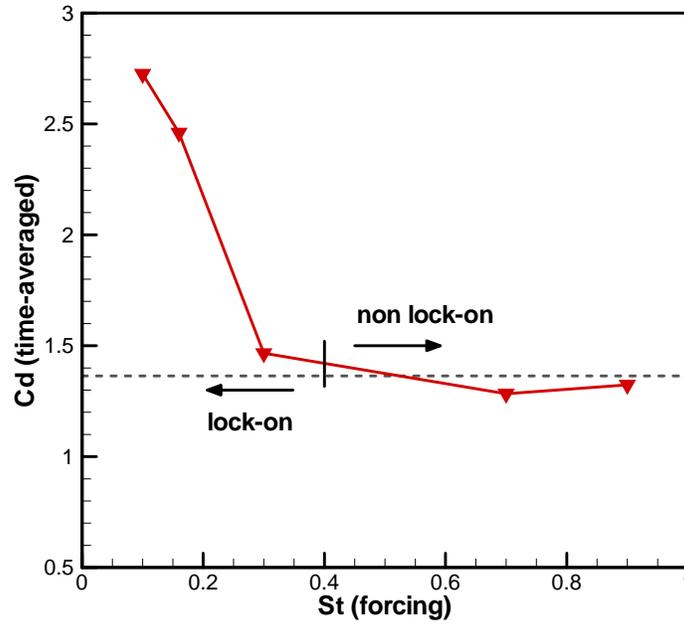
(a) time-averaged drag coefficients (dash line: the value for stationary cylinder case)
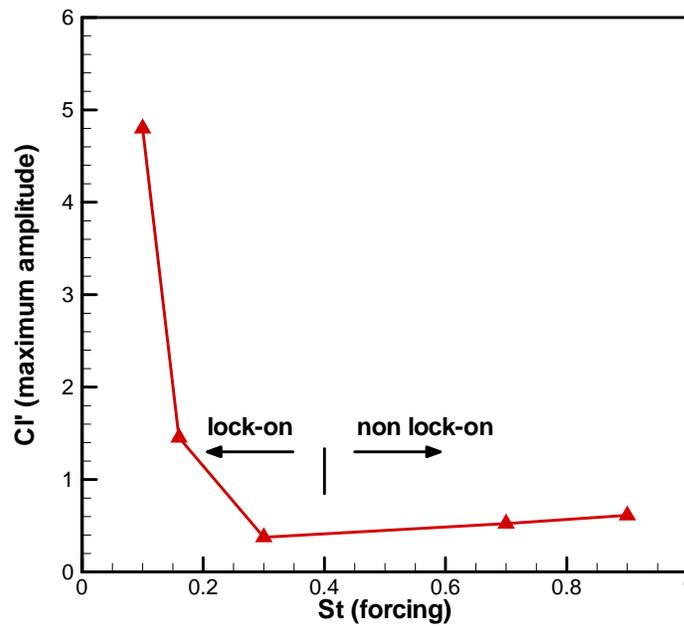
(b) maximum amplitude of the lift coefficient fluctuation

Fig. 8 Variations of drag and lift coefficients versus St_f
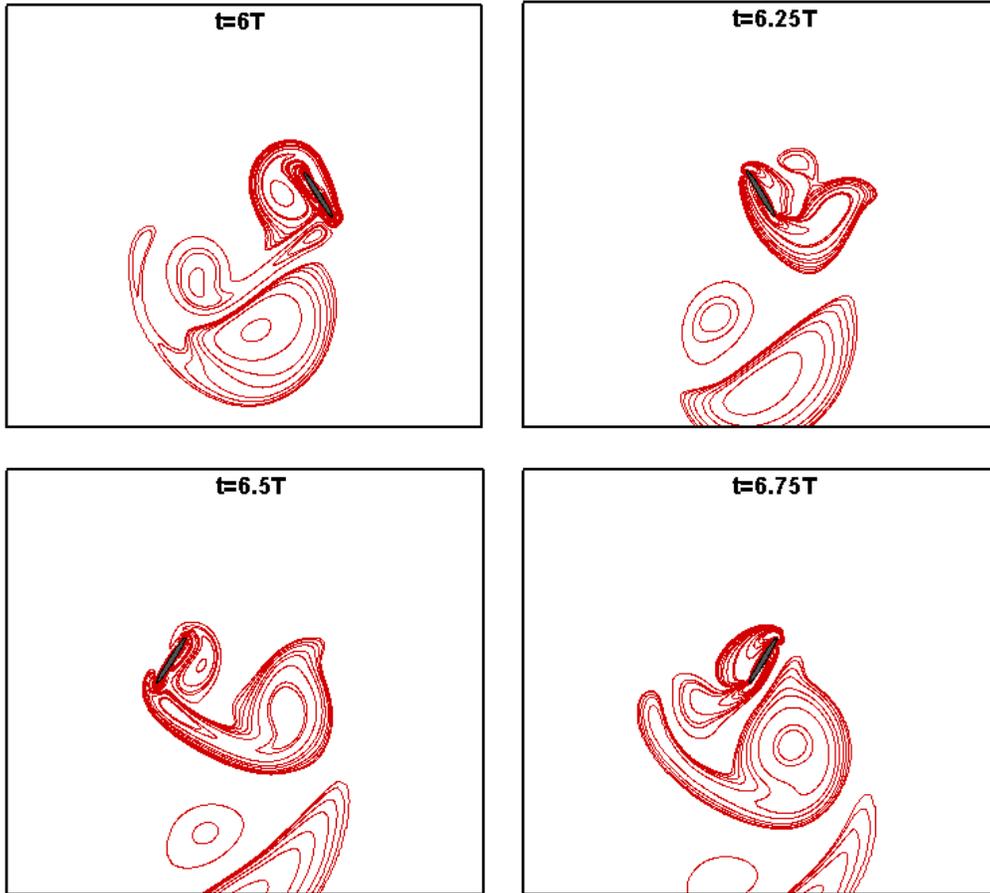
Fig. 9 Vorticity contours in one cycle for $\phi = \pi/4$
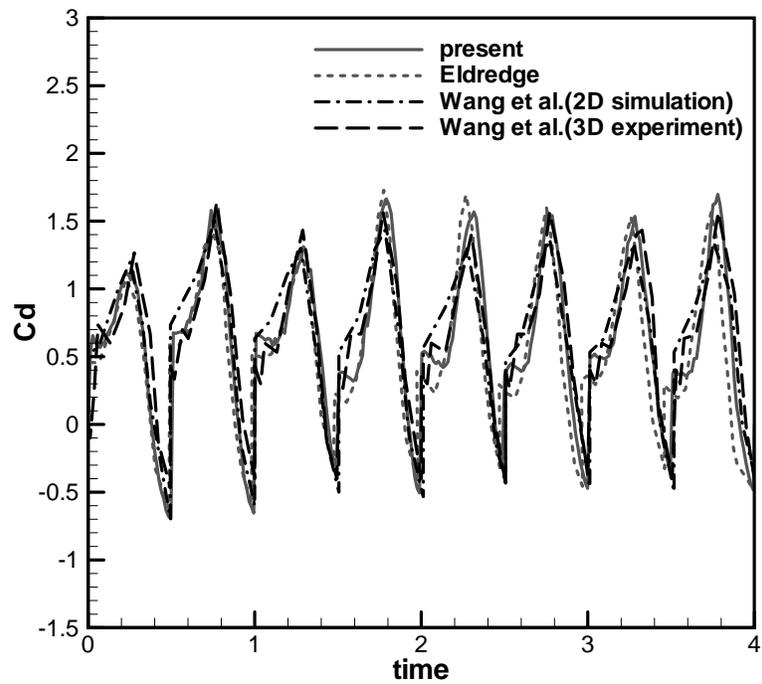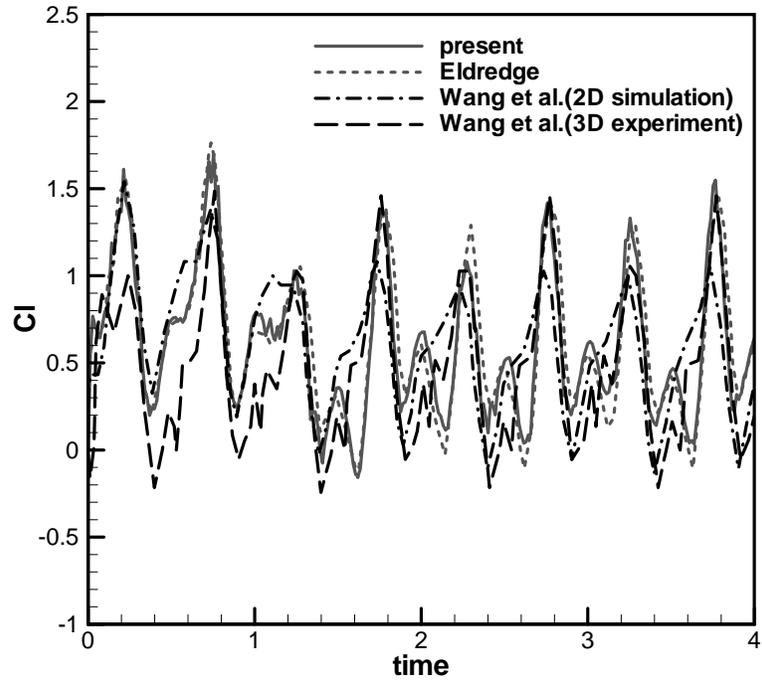
Fig. 10 Time histories of lift and drag coefficients for $\phi = \pi/4$
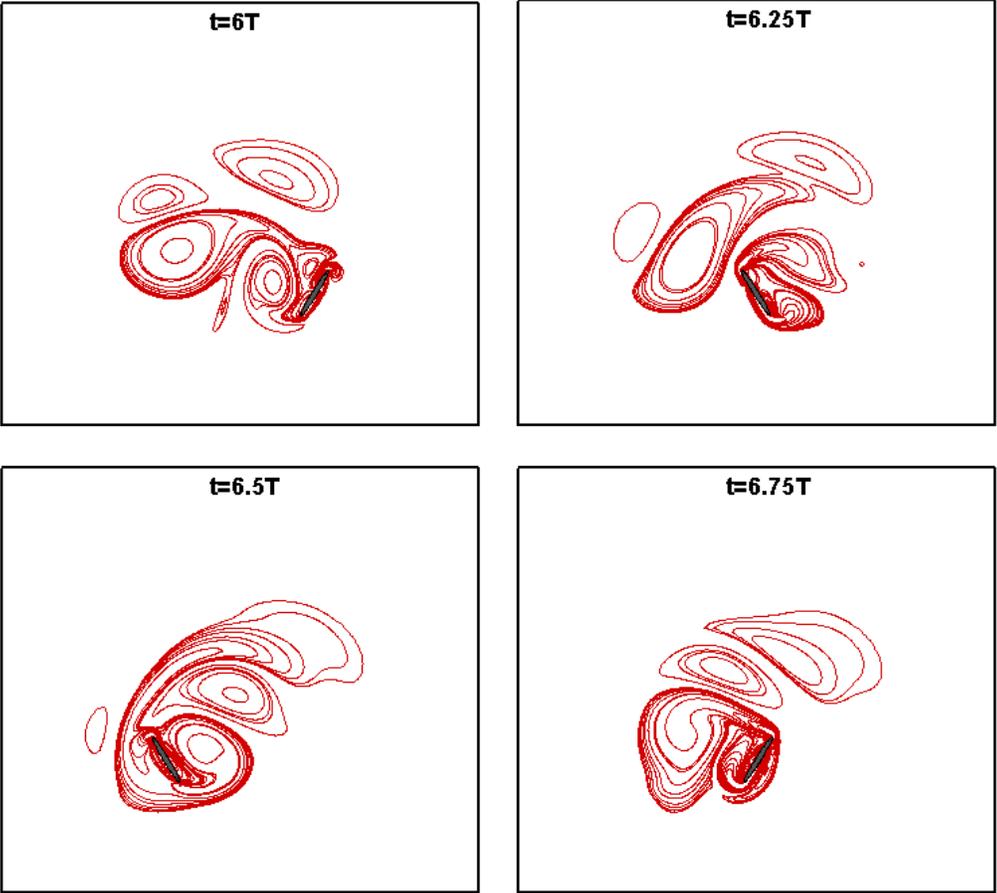
Fig. 11 Vorticity contours in one cycle for $\phi = -\pi/4$
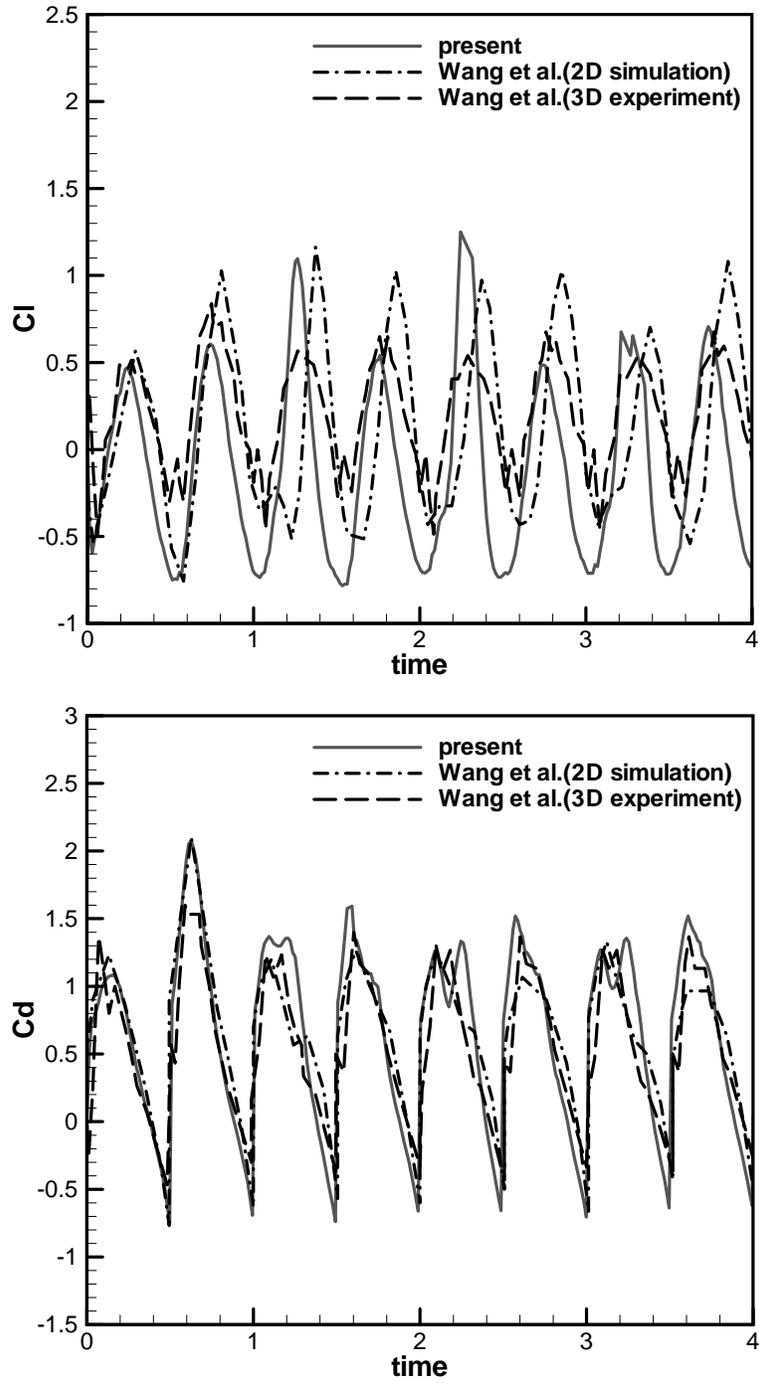
Fig. 12 Time histories of lift and drag coefficients for $\phi = -\pi/4$
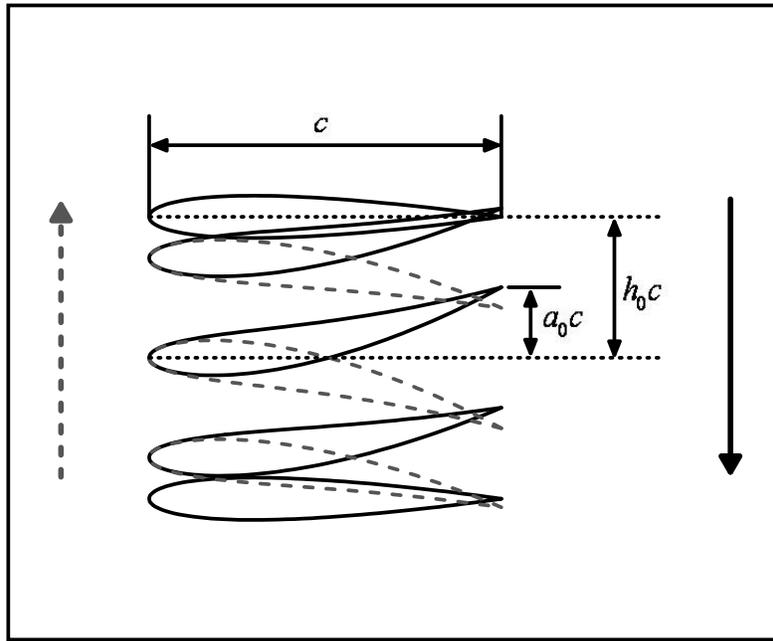
Fig. 13 Plunge and deflection motion of a single flexible airfoil

(solid line: downstroke motion; dashed line: upstroke motion)
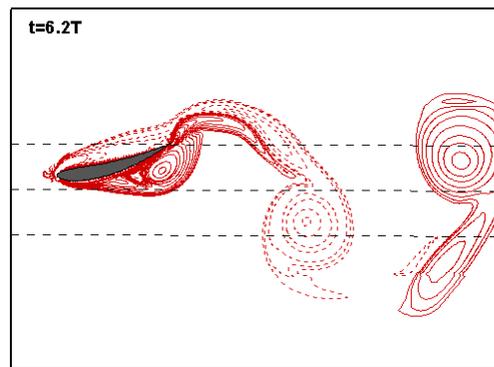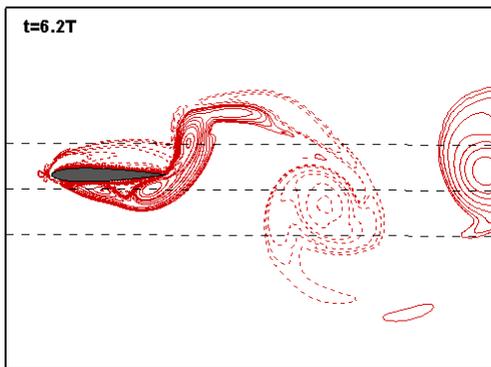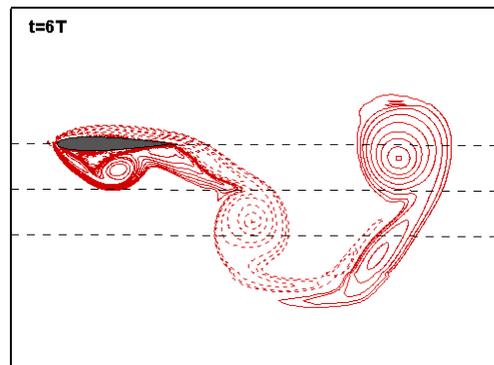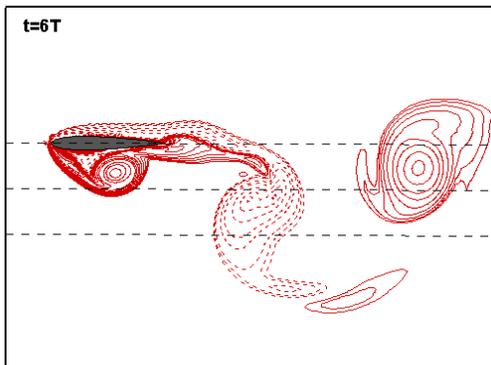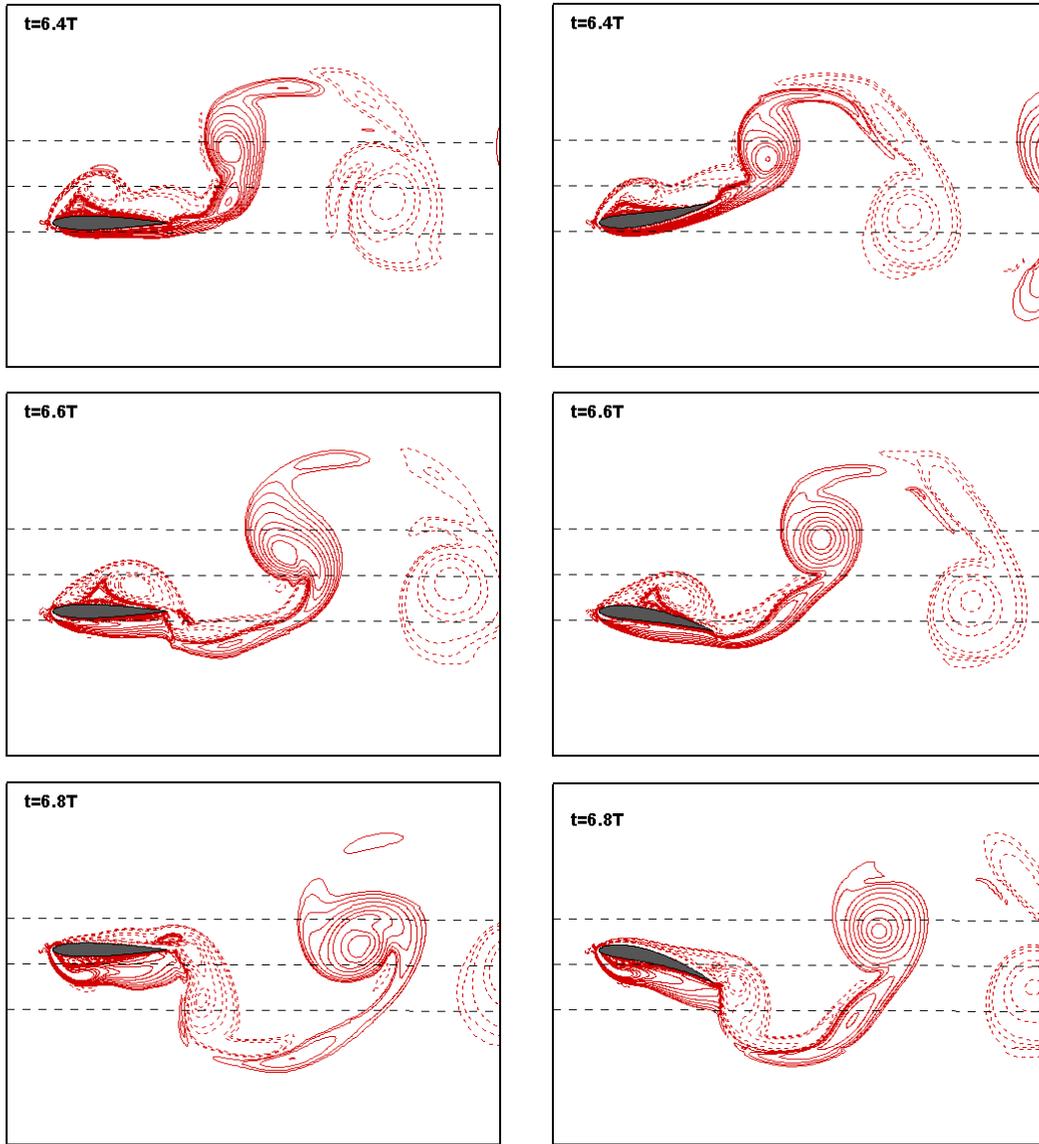
Fig. 14 Instantaneous vorticity contours for flapping airfoil in one cycle

(a) rigid airfoil; (b) flexible airfoil
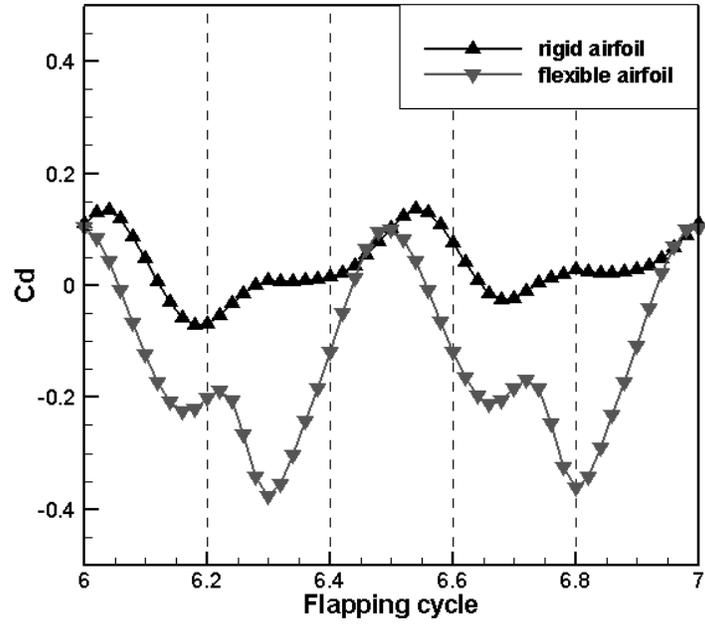
Fig. 15 The evolution of drag coefficients for rigid and flexible airfoils
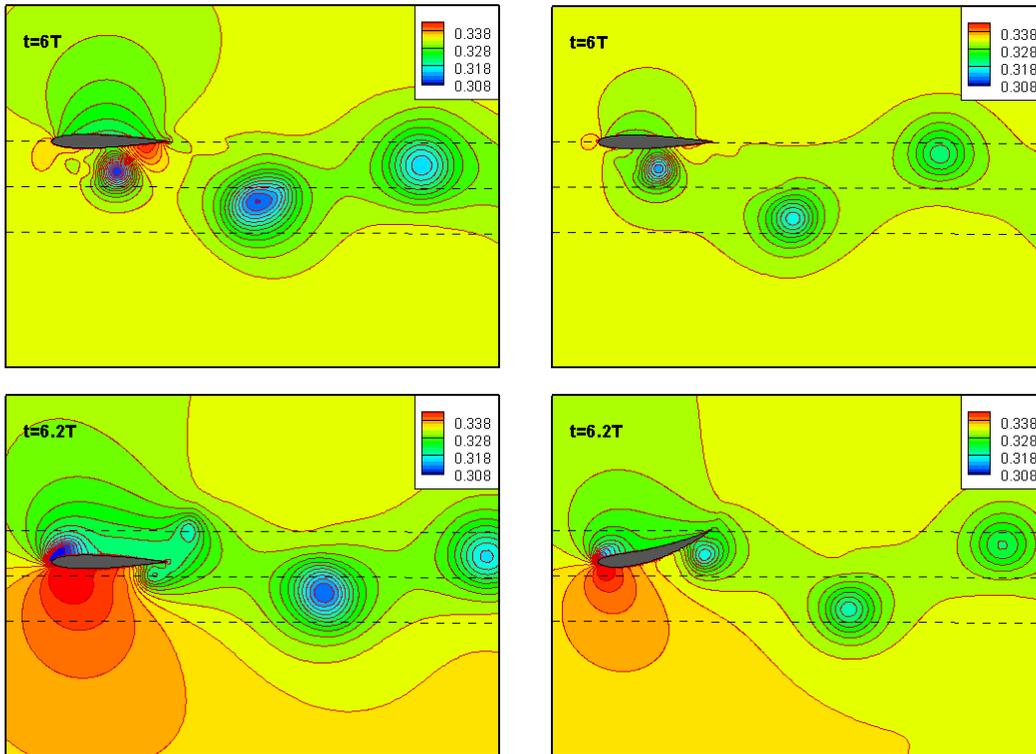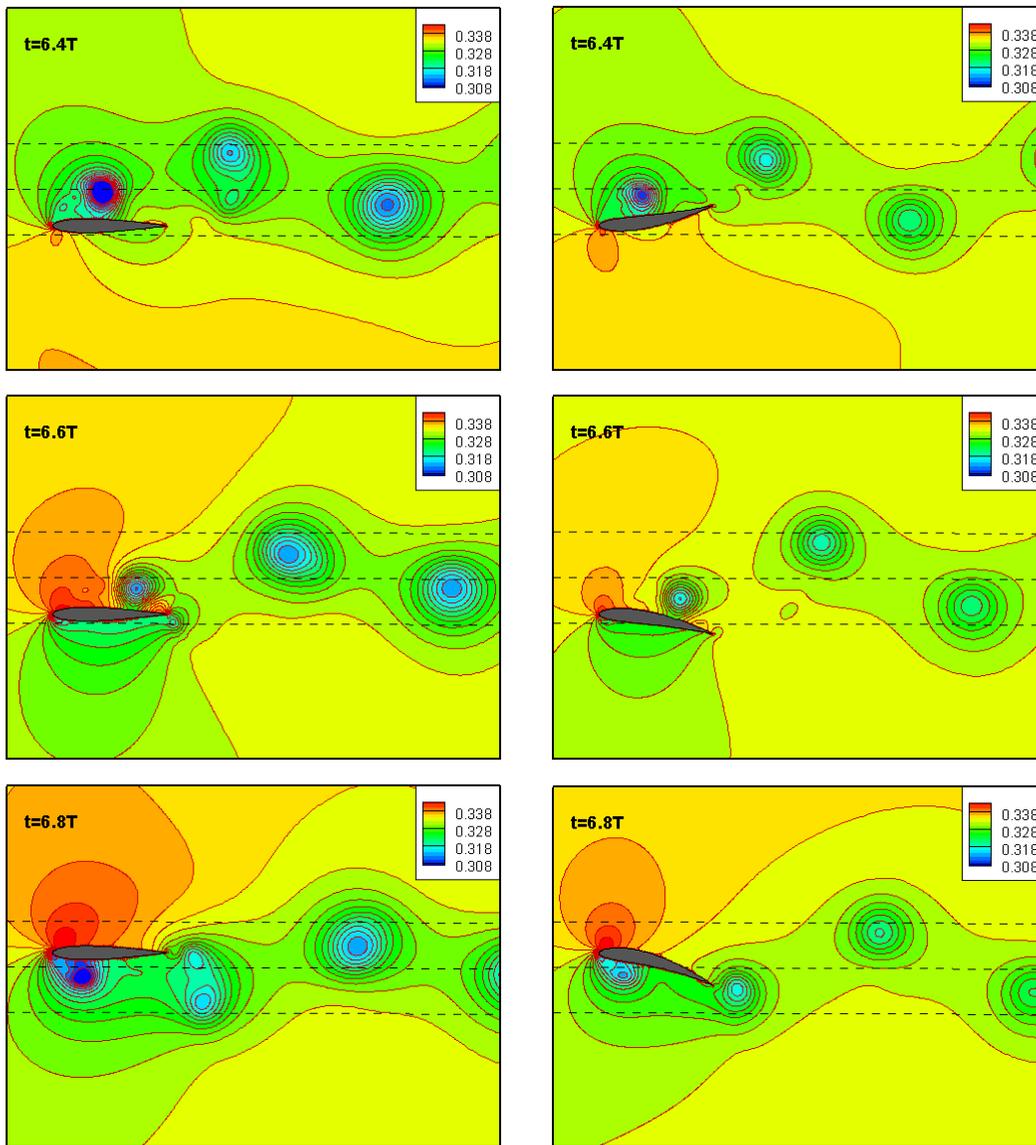
(a)   (b)

Fig. 16 The pressure contours for flapping airfoil in one cycle
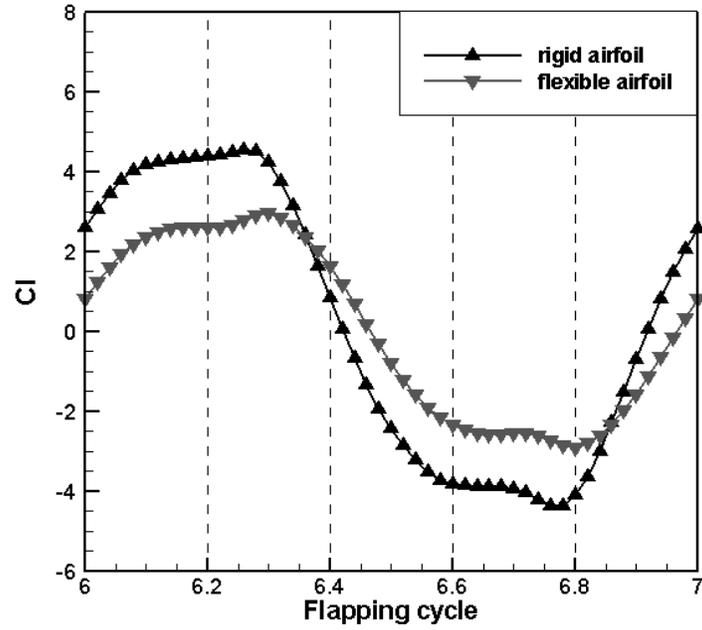
(a) rigid airfoil; (b) flexible airfoil

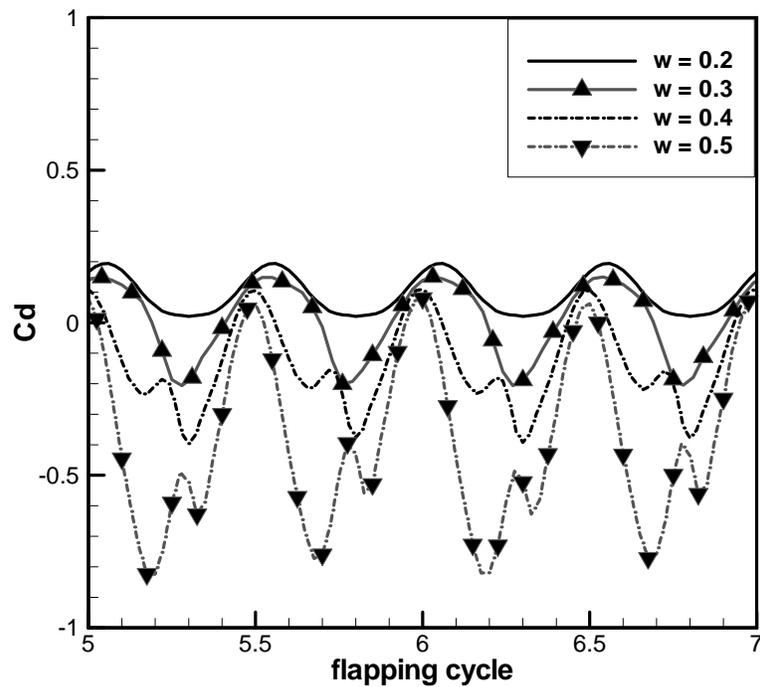**Fig. 17 The evolution of lift coefficients for rigid and flexible airfoils**

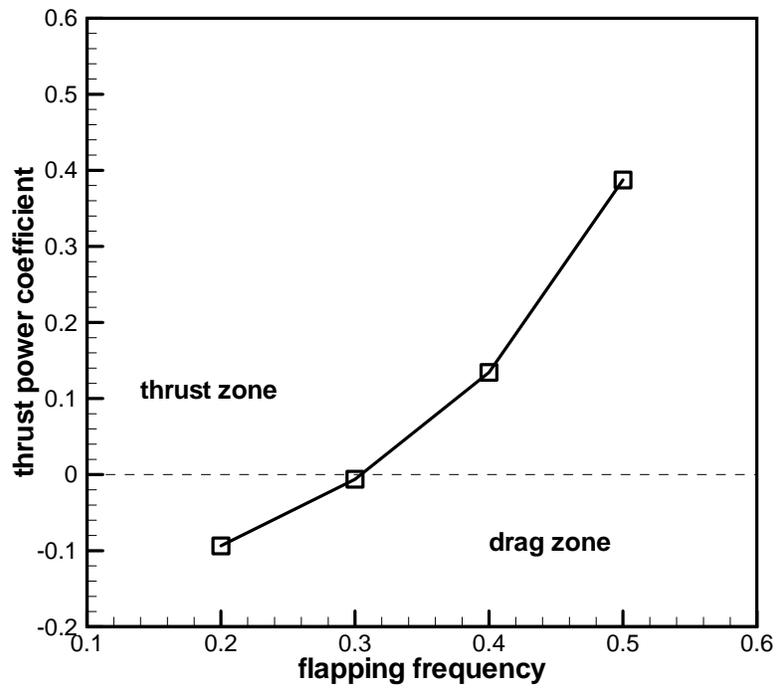**Fig. 18 The evolution of drag coefficients with different frequencies**

Fig. 19 Effect of flapping frequency on thrust power coefficient