Copyright © 2009 Tech Science Press CMC, vol.11, no.3, pp.209-228, 2009

A Direct Forcing Immersed Boundary Method Based Lattice Boltzmann Method to Simulate Flows with Complex Geometry

Cheng-Hsiu Yang[1], Cheng Chang[1] and Chao-An Lin[1,2]

Abstract: In the present study, a lattice Boltzmann method based new immersed boundary technique is proposed for simulating two-dimensional viscous incompressible flows interacting with stationary and moving solid boundaries. The lattice Boltzmann method with known force field is used to simulate the flow where the complex geometry is immersed inside the computational domain. This is achieved via direct-momentum forcing on a Cartesian grid by combining "solid-body forcing" at solid nodes and interpolation on neighboring fluid nodes. The proposed method is examined by simulating decaying vortex, 2D flow over an asymmetrically placed cylinder, and in-line oscillating cylinder in a fluid at rest. Numerical simulations indicate that this method is second order accurate, and all the numerical results are compatible with the benchmark solutions.

Keywords: Direct forcing, immersed boundary method, lattice Boltzmann method.

1 Introduction

Lattice Boltzmann method (LBM) [Qian, d'Humieres, and Lallemand (1992); Chen and Doolen (1998); Yu, Mei, Luo, and Shyy (2003); Shih-Kai Chien and Chen (2008)] has been successfully applied to various hydrodynamic problems and the major advantages of the LBM are explicit, easy to implement, and natural to parallelize. However, the capability of the lattice Boltzmann method to model complex geometry may not be trivial due to the Cartesian grid adopted and the complex boundary implementation along the curved boundary. The major difficulty encountered is the representation of the complex geometry, since the Cartesian grid does not conform with the curved boundary.

Various methodologies have been put forward to tackle the LBM simulation of

[1] Department of Power Mechanical Engineering,National Tsing Hua University, Hsinchu 30013, Taiwan.

[2] Corresponding author, calin@pme.nthu.edu.tw

complex geometry flows. The simplest approach is to employ bounce back scheme. Because the bounce-back on the node (BBN) has been shown to be first order accurate, the bounce-back on the link (BBL) is proposed by Ladd (1994). However, in order to preserve the geometric integrity, it is necessary to use a large number of lattice grids. On the other hand, Chen, Martinez, and Mei (1996) extended the "extrapolation scheme" to curved boundary, where the boundary is represented by the lattice nodes closest to the curved surface using castellated approach. Filippova and Hänel (1998) proposed a method using a simple linear interpolation between a fictitious equilibrium distribution function and a well-chosen near-boundary distribution function. The weighting factor of the interpolation is determined by the distance between the boundary and the near-boundary lattice. Mei, Luo, and Shyy (1999) further improved its numerical stability. Lallemand and Luo (2003) combined the bounce-back scheme and interpolation scheme to treat a moving curved boundary by the lattice Boltzmann method. The bounce-back scheme simulates a stationary boundary, and an additional term is added to implement a moving boundary. This treatment is an extension of that proposed by Bouzidi, Firdaouss, and Lallemand (2001). An alternative proposed by Chang, Liu, and Lin (2009) is to impose the boundary condition along the complex geometry based on the linear interpolation of the primitive variables.

The immersed boundary method (IBM) is another convenient approach to treat fluid flows involving complex boundary by generating external force field to mimic the immersed boundary. IBM can be categorized as feedback forcing [Peskin (1972); Saiki and Biringen (1996); Mittal and Iaccarino (2005)] and direct forcing [Mohd-Yusof (1997); Fadlun, Verzicco, Orlandi, and Mohd-Yusof (2000); Mittal and Iaccarino (2005)] approaches. However, the major drawback of the existing feedback forcing is the restriction of small CFL number. Aiming at improving the CFL number restriction, Su, Lai, and Lin (2007) proposed a new immersed boundary technique for the simulation of flows interacting with solid boundary within the Navier-Stokes framework. In Chen, Lin, and Lin (2007), the lattice Boltzmann method is combined with the immersed boundary technique of Su, Lai, and Lin (2007) to simulate flows with complex boundary. In Su, Lai, and Lin (2007) and Chen, Lin, and Lin (2007), dirac delta function was employed to link the force between the Lagrangian marker and the computational grid, and this may potentially smear the solution across the interface.

In contrast to the feedback forcing approach, the direct forcing approach seems to be more efficient in simulating flows with rigid immersed boundary. This scheme was originally proposed by Mohd-Yusof (1997), and the direct momentum forcing is applied to a set of points adjacent to the immersed surface. It is equivalent to the direct imposition of the velocity boundary conditions on the Eulerian grids. Therefore, information regarding these grids (forcing nodes) either external or internal to the immersed boundary must be determined [Kim, Kim, and Choi (2001); Balaras (2004); Liao, Chang, Lin, and McDonough (2010)].

Although the direct forcing concept has been applied within the immersed boundary method, it is mostly implemented within the Navier-stokes equation[Kim, Kim, and Choi (2001); Liao, Chang, Lin, and McDonough (2010)]. In this paper, a combination of the lattice Boltzmann method and the direct forcing based immersed boundary method is proposed to simulate flows with complex geometry. Here, the curved boundary is represented by a series of Lagrangian markers. The fluid velocity of node adjacent to the solid boundary is obtained by linear interpolation between the Lagrangian marker and the second fluid node further away. To improve the capability to model flow with moving object, a solid-body-forcing procedure is also employed within the solid region. The validity and accuracy of the new method are scrutinized by simulating decaying vortex, 2D flow over an asymmetrically placed cylinder, and in-line oscillating cylinder in a fluid at rest. All the numerical results are compatible with the benchmark solutions.

2 The lattice Boltzmann method

2.1 The lattice Boltzmann equation

The lattice Boltzmann equation adopting a uniform lattice with Bhatnagar-Gross-Krook collision model [Chen, Chen, Martinez, and Matthaeus (1991); Qian, d'Humieres, and Lallemand (1992); Chen and Doolen (1998)] can be expressed as,

$$f_i(\vec{x}+\vec{e}_i dt, t+dt) - f_i(\vec{x},t) = -\frac{1}{\tau}[f_i(\vec{x},t) - f_i^{eq}(\vec{x},t)] \tag{1}$$

where f_i is the particle distribution function along the particle speed direction $\vec{e}_i$ at position $\vec{x}$ and time t. f_i^{eq} is the equilibrium distribution function and τ is the single relaxation time that controls the rate approaching equilibrium. Based on the particle distribution function, the macroscopic density ρ and velocity $\vec{u}$ are defined as,

$$\sum_i f_i = \rho, \quad \sum_i f_i \vec{e}_i = \rho \vec{u} \tag{2}$$

The equilibrium distribution functions, which depend on the local density and velocity, are given by the form [Qian, d'Humieres, and Lallemand (1992)],

$$f_i^{eq} = \omega_i \rho [1 + \frac{3}{c^2}\vec{e}_i \cdot \vec{u} + \frac{9}{2c^4}(\vec{e}_i \cdot \vec{u})^2 - \frac{3}{2c^2}\vec{u} \cdot \vec{u}] \tag{3}$$

where $c = dx/dt$ is the lattice speed, and dx and dt are the lattice width and time step, respectively. ω_i is a weighting factor. For the present 2D applications, D2Q9 model are adopted (see Fig. 1). The particle speed $\vec{e}_i$ adopting D2Q9 model are defined as,

$$\vec{e}_0 = 0 \tag{4}$$

$$\vec{e}_i = (\cos[\pi(i-1)/2], \sin[\pi(i-1)/2])c \quad i = 1,2,3,4 \tag{5}$$

$$\vec{e}_i = (\cos[\pi(i-4-1/2)/2], \sin[\pi(i-4-1/2)/2])\sqrt{2}c \quad i = 5,6,7,8 \tag{6}$$

and the weighting factors are $\omega_0 = 4/9$, $\omega_{i=1,2,3,4} = 1/9$, and $\omega_{i=5,6,7,8} = 1/36$. Moreover, the speed of sound is $C_s = c/\sqrt{3}$ and the corresponding kinematic viscosity is $\nu = (\tau - 0.5)C_s^2 dt$ in the simulation.

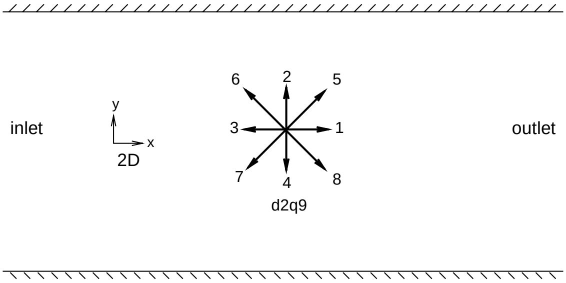

Figure 1: 2D (d2q9) model.

2.2 The forcing strategies

In the proposed simulation, the lattice Boltzmann method is combined with the direct forcing based immersed boundary method. Therefore, a force term should be added to the lattice Boltzmann equation. Then Eq. (1) becomes,

$$f_i(\vec{x}+\vec{e}_i dt, t+dt) - f_i(\vec{x},t) = -\frac{1}{\tau}[f_i(\vec{x},t) - f_i^{eq}(\vec{x},t)] + \Delta t F_i \tag{7}$$

The issue of computing the forcing function F_i of Eq. 7, when the interface does not coincide with grid nodes is discussed here. A combination of two strategies is adopted for calculations to be reported herein: viz., interpolation (at forcing points in the fluid domain) and a solid-body-forcing procedure within the solid region, the latter of which leads to improved accuracy (in particular, significant reduction of non-physical temporal oscillations of dependent variables) when treating moving boundaries. This has been investigated by Liao, Chang, Lin, and McDonough (2010) using the Navier-Stokes equation approach.

Thus, two issues should be addressed here. First is the forcing location in the fluid region, and the second is specific details of solid-body-forcing. A typical solid-fluid boundary within the computational domain is shown in Fig. 2, where the closest nodes adjacent to the boundary in the fluid domain are termed the forcing nodes (C) as represented by the filled triangles. The open squares are the Lagrangian markers (B) used to mimic the solid-fluid boundary, and the filled circles represent the second fluid nodes (A) beyond the forcing nodes. In the present curved boundary approach, the forcing nodes are used to impose the momentum condition due to the presence of the solid-fluid boundary.

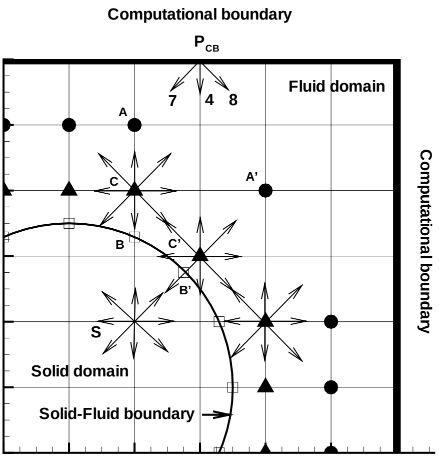

Figure 2: Distribution functions along the boundary nodes. ▲: forcing node, □: Lagrangian marker, •: second fluid node.

The correct macroscopic velocity of the forcing nodes are obtained by linear interpolation between the velocities of the Lagrangian marker (B) and the second fluid node (A), as shown in Fig 3.

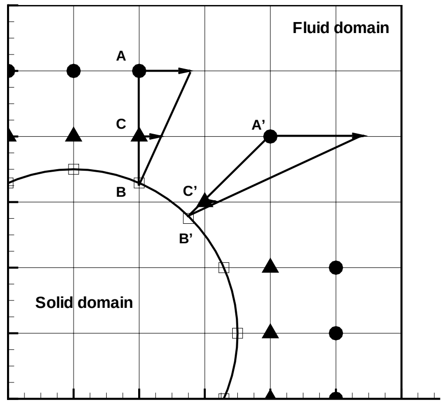

Figure 3: Geometry of flows with an immersed boundary. ▲: forcing node, □: Lagrangian marker, •: second fluid node.

$$\vec{V}_C = \vec{V}_A + (\vec{V}_A - \vec{V}_B)\frac{\overline{AC}}{\overline{AB}} \tag{8}$$

The determination of the location of the Lagrangian marker is addressed here. If the forcing node one vertical or horizontal edge connected with the solid-fluid boundary, for example at node C, the Lagrangian marker is determined as the intersection of the line $\overrightarrow{AC}$ with the solid-fluid boundary, i.e. marker point B. However, for node C' there are one vertical and one horizontal edges connected to the solid-fluid boundary. Therefore, the interpolation is not unique. In order to avoid this ambiguity, the method proposed by Liao, Chang, Lin, and McDonough (2010) (using the Navier-Stokes solver) is adopted, that the Lagrangian marker is determined as the intersection of the line $\overrightarrow{A'C'}$ with the solid-fluid boundary, i.e. marker point B'.

After the flow variables at each forcing node is obtained, the force required to modify the momentum is expressed as,

$$\rho V_C = \sum_i f_i \vec{e}_i + \frac{\Delta t}{2}\vec{F} \tag{9}$$

where V_C is the correct velocity at forcing nodes, and the first term in the right represents the macroscopic velocity computed by LBM. By Eq. (9), $\vec{F}$ can be obtained for each time step and substituted into the equation [Guo, Zheng, and Shi (2002)],

$$F_i = (1 - \frac{1}{2\tau})\omega_i [\frac{e_i - V_C}{C_s^2} + \frac{(e_i \cdot V_C)}{C_s^4} e_i] \cdot \vec{F} \tag{10}$$

to determine the force term F_i in the lattice Boltzmann equation with a force term, i.e. Eq. (7). It should be noted that in Eq. (7), the velocity in the equilibrium distribution function should be updated by V_C in Eq. (9).

As indicated earlier, complications for moving-boundary problems arise from the fact that positions of the forcing points near the interface between fluid and solid change from time step to time step. This is alleviated by adopting a solid-body-forcing strategy within the solid domain, wherein the velocity of the point S within the solid is forced to equal the velocity of the moving object. In particular, the solid-body forcing strategy is imposed within the solid region through the solid-body force calculated as

$$\rho \vec{V}_{move} = \sum_i f_i \vec{e}_i + \frac{\Delta t}{2} \vec{F} \tag{11}$$

where $\vec{V}_{move}$ represents the velocity of the moving object. Note that the solid-body-forcing procedure must be carried out at every time step for moving-boundary problems.

2.3 Boundary condition

Along the computational boundary, $f_i(\vec{x},t)$ due to the inward streaming operations may originate from the undefined nodes external to the computational domain, therefore measures have to be taken to prescribe these unknown particle distribution functions.

The unknown particle distribution functions at the plane boundary are expressed as a combination of the local known value and a corrector [Chang, Liu, and Lin (2009); Ho, Chang, Lin, and Lin (2009)],

$$f_p(\vec{x},t) = f_p^*(\vec{x},t) + \frac{\alpha_p}{c} \vec{e}_p \cdot \vec{Q} \tag{12}$$

where $\vec{Q}$ is the corrector to enforce the required momentum.

Consider a typical point P_{CB} along the computational boundary shown in Fig. 2, where the unknown distribution functions are f_4, f_7 and f_8, i.e. $f_4 = f_4^* - \alpha_4 Q_y$,

$f_7 = f_7^* - \alpha_7(Q_x + Q_y)$ and $f_8 = f_8^* + \alpha_8(Q_x - Q_y)$. Therefore, the macroscopic velocity and density at the point P_{CB} using equation 2, in conjunction with equation 12, can be expressed as,

$$\begin{aligned}
\rho &= f_0 + f_1 + f_2 + f_3 + (f_4^* - \alpha_4 Q_y) + f_5 + f_6 \\
&\quad + (f_7^* - \alpha_7(Q_x + Q_y)) + (f_8^* + \alpha_8(Q_x - Q_y)) \\
\rho u &= f_1 + f_5 + (f_8^* + \alpha_8(Q_x - Q_y)) - f_3 - f_6 - (f_7^* - \alpha_7(Q_x + Q_y)) \\
\rho v &= f_2 + f_5 + f_6 - (f_4^* - \alpha_4 Q_y) \\
&\quad - (f_7^* - \alpha_7(Q_x + Q_y)) - (f_8^* + \alpha_8(Q_x - Q_y)) \qquad (13)
\end{aligned}$$

These equations can be used to solve for ρ, Q_x and Q_y, and hence f_4, f_7 and f_8. For simplicity, $\alpha_p = \omega_p$. This coefficient produces a more compact form of the distribution functions and the boundary condition of Zou and He (1997) can be recovered. The explicit forms of the unknown particle distribution functions as shown below.

$$\rho = \frac{f_0 + f_1 + f_3 + 2(f_2 + f_5 + f_6)}{1 + v} \qquad (14)$$

$$f_4 = f_4^* - \frac{2}{3}\rho v + \frac{2}{3}(f_2 - f_4^* + f_5 - f_7^* + f_6 - f_8^*) \qquad (15)$$

$$\begin{aligned}
f_7 &= f_7^* - \frac{1}{2}\rho u - \frac{1}{6}\rho v + \frac{1}{2}(f_1 - f_3) \\
&+ \frac{1}{6}(f_2 - f_4^*) + \frac{2}{3}(f_5 - f_7^*) - \frac{1}{3}(f_6 - f_8^*) \qquad (16)
\end{aligned}$$

$$\begin{aligned}
f_8 &= f_8^* + \frac{1}{2}\rho u - \frac{1}{6}\rho v - \frac{1}{2}(f_1 - f_3) \\
&+ \frac{1}{6}(f_2 - f_4^*) - \frac{1}{3}(f_5 - f_7^*) + \frac{2}{3}(f_6 - f_8^*) \qquad (17)
\end{aligned}$$

For simplicity, formulation $f_p^*(\vec{x},t) = f(\vec{x},\vec{e}_p,t-dt)$ is adopted. However, if $f_p^*(\vec{x},t) = f(\vec{x},-\vec{e}_p,t)$, the present form recovers the form by Zou and He (1997).

Table 1: The maximum relative errors of the decaying vortex simulated with IBM-LBM method.

Lattice size	Maximum relative error	Order
41×41	2.0016847688226687E-003	-
81×81	6.0194496776706092E-004	1.824
161×161	1.7075582952682731E-004	1.878
321×321	4.6410782536254747E-005	1.918

3 Numerical Results

3.1 Decaying vortex

The decaying vortex problem is frequently used to examine the accuracy of the simulation since the analytic solution is available, which is shown below,

$$u(x,y,t) = -Ucos(\pi x/L)sin(\pi y/L)e^{-2\pi^2 Ut/(ReL)}, \tag{18}$$

$$v(x,y,t) = Usin(\pi x/L)cos(\pi y/L)e^{-2\pi^2 Ut/(ReL)}, \tag{19}$$

$$\rho(x,y,t) = \rho_o - \frac{\rho_o U^2}{4C_s^2}[cos(2\pi x/L) + sin(2\pi y/L)]e^{-4\pi^2 Ut/(ReL)}. \tag{20}$$

The computational domain is set up as $[-L,L] \times [-L,L]$ where $L = 1$. The dimensionless relaxation time is $\tau = 0.65$, the Reynolds number is $Re = UL/\nu = 10$, and the average density is $\rho_0 = \int_{-L}^{L}\int_{-L}^{L}\rho_0 dxdy/2L/2L$. Since the simulation is time varying, so the computations are all up to dimensionless time $Ut/L = 1$. Four different uniform grids $(41, 81, 161, 321)$ are used in the simulations.

The immersed boundary technique is applied on a circle inside the computational domain, where the radius is 0.5. The exact time varying velocity conditions are imposed along the circular immersed boundary. The definition of maximum relative error is denoted as,

$$Err_{max} = \frac{|u - u_{exact}|_{max}}{U} \tag{21}$$

The rate of convergence is computed by taking logarithmic for the ratio of the two successive errors as $Rate = \log_2 \frac{Err_N}{Err_{N/2}}$, where Err_N denotes the error of the grid resolution N. As shown in Table 1 and Fig. 4, the present method is approximately second-order accurate.

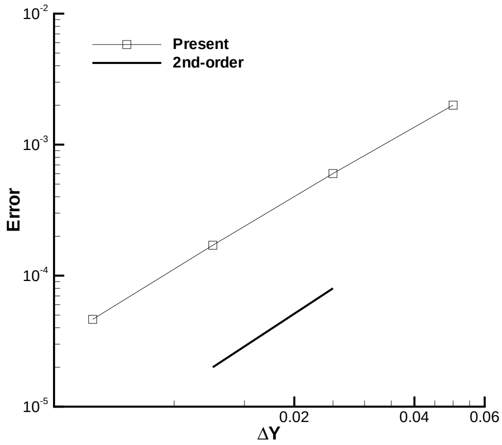

Figure 4: Maximum relative error of decaying vortex.

3.2 Flow over an asymmetrically placed cylinder in a channel

The flow past a stationary circular cylinder is a typical problem and has been widely investigated [Schafer and Turek (1996); Mei, Yu, Shyy, and Luo (2002); Chen, Lin, and Lin (2007)]. For Reynolds number below 47, the flow structure remains steady with stationary recirculating vortices behind the cylinder. As the Reynolds number is elevated, the steadiness breaks down and the vortex starts to shed up and down alternatively. This shedding frequency and the intensity of the vortex also increase in tandem with the elevated level of the Reynolds number.

Schafer and Turek (1996) reported a set of 2D and 3D benchmark results for laminar flows over a circular cylinder of radius r that is asymmetrically placed inside a channel. The distances from the center of the cylinder to the upper wall and lower wall are $4.2r$ and $4.0r$, respectively. The 2D geometric layout is shown in Fig. 5. The inlet boundary is placed at 4 radii upstream of the cylinder center, $l^{+} = 4.0r$, and the exit boundary is located 40 radii downstream of the cylinder center. In the present study, $r = 30\Delta x$ is used. The grid resolutions in the x and y directions are 1321 and 247, respectively. The Eulerian grid spacing is $\Delta x = \Delta y = 1/40$, and the time step size is $\Delta t = \Delta x$. The Reynolds numbers are $Re = 2rU_{ave}/\nu = 20$ and 100. At $Re = 100$, the flow becomes unsteady and periodic vortex shedding is observed.

A parabolic velocity profile of maximum speed U_{max} is applied at the inlet boundary, and the maximum Mach number is $Mc = U_{max}/C_s = 0.1$. No-slip boundary

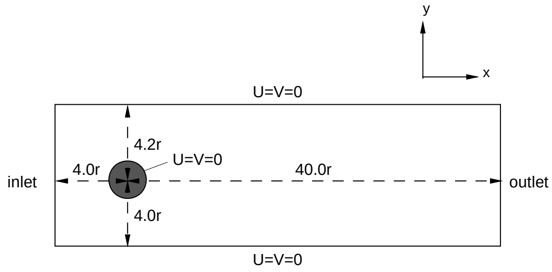

Figure 5: Configuration of flow over an asymmetrically placed cylinder in a channel.

condition is applied along the upper and lower wall. At the exit, linear extrapolation is applied for the unknown distribution functions.

The Reynolds number based on the average inlet velocity $U_{ave} = 2U_{max}/3$ is $Re = 2rU_{ave}/\nu$. The drag and lift coefficient over the cylinder are defined as,

$$C_D = \frac{F_D}{\rho_{in} U_{ave}^2 r}, \qquad C_L = \frac{F_L}{\rho_{in} U_{ave}^2 r} \tag{22}$$

where the drag force F_D and the lift force F_L are obtained from the following equations [Chen, Lin, and Lin (2007); Su, Lai, and Lin (2007)],

$$F_D = -\sum_m F_{x,m} \Delta x^2, \qquad F_L = -\sum_m F_{y,m} \Delta x^2 \tag{23}$$

Flows at $Re = 20$ and $Re = 100$ are simulated. The flow is steady when the Reynolds number is equal to 20, and the drag and lift coefficient remain constant after sufficient time steps, as shown in Table 2. At $Re = 100$, periodic vortex shedding is observed. The drag and lift coefficient have a periodic fluctuation, as shown in Fig. 6 and 7. Two peaks in the drag coefficient correspond to the existence of a weaker vortex and a stronger vortex alternately shed behind the cylinder. This phenomenon is due to that the cylinder is asymmetry placed in the channel. Instantaneous vorticity contours at $Re = 100$ for a single period are shown in Fig. 8.

Table 2: C_D and C_L for the flow over a cylinder asymmetrically placed in a channel at Re=20.

Re=20	Present method	Chen et al. (2007)	Schäfer and Turek (1996)
C_D	5.633	5.679	5.57-5.59
C_L	0.0109	0.0114	0.0104-0.0110

The Strouhal number, $St = 2r/U_{ave}T$, can also be evaluated, where T is the period of the flow. Table 3 lists the maximum of C_D, C_L and the Strouhal number of the simulation. The present results compare favorably with previous findings [Schafer and Turek (1996); Mei, Yu, Shyy, and Luo (2002); Chen, Lin, and Lin (2007)].

3.3 In-line oscillating cylinder in a fluid at rest

To extend the present curved boundary technique to moving boundary problem, an in-line oscillating cylinder within a fluid at rest is simulated. The two key parameters in this flow are the Reynolds number, $Re = U_{max}D/\nu$, and the Keulegan-Carpenter number, KC= U_{max}/fD, where U_{max} is the maximum velocity of the cylinder, D is the diameter of the cylinder, ν is the kinematic viscosity of the fluid, and f is the characteristic frequency of the oscillation.

Based on the work of Dütsch, Durst, Becker, and Lienhart (1998), the Reynolds and Keulegan-Carpenter numbers in the present simulation are set to be 100 and 5, respectively. The cylinder's translational motion is described by a simple harmonic oscillation, $x(t) = -A\sin(2\pi ft)$, where A is the amplitude of the oscillation. The cylinder is initially located at the center of the computational domain, which is $15D\times 8D$ in the x and y directions, as shown in Fig. 9. The grid resolution of the domain is 901×481, and 60 grid spacings correspond to the diameter of the cylinder. All computations are started from a quiescent field and the integration in time is performed until periodic vortex shedding is established.

Fig. 10 shows vorticity contours at four different phase-angles, which can be compared directly with the work of Dütsch, Durst, Becker, and Lienhart (1998). When the cylinder moves to the left, upper and lower boundary layers develop on the leading face the cylinder and separate at symmetric (top and bottom) positions downstream. Two counter-rotating vortices are produced by the separating flow. These vortices persist until the cylinder reaches the left-maximum position and starts its rightward motion; then the same vortex structure is created on the opposite side of the cylinder. It is interesting to note that this reversed motion carries the cylinder back through already-stirred fluid, resulting in splitting of vortex pairs produced during the preceding leftward motion. These results are consistent with the corre-

Table 3: C_D, C_L, and *St* for the flow over a cylinder asymmetrically placed in a channel at Re=100.

Re=100	Present method	Chen et al. (2007)	Mei et al. (2002)	Schäfer and Turek (1996)
C_D	3.324	3.333	3.2275	3.22-3.24
C_L	1.011	1.0511	1.0040	0.99-1.01
St	0.302	0.3003	0.3033	0.295-0.305

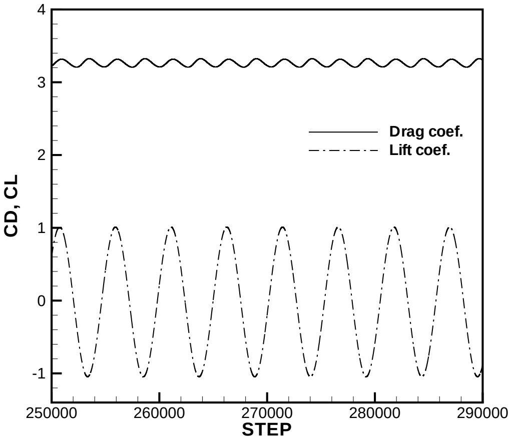

Figure 6: The time evolution of drag and lift coefficients at Re=100.

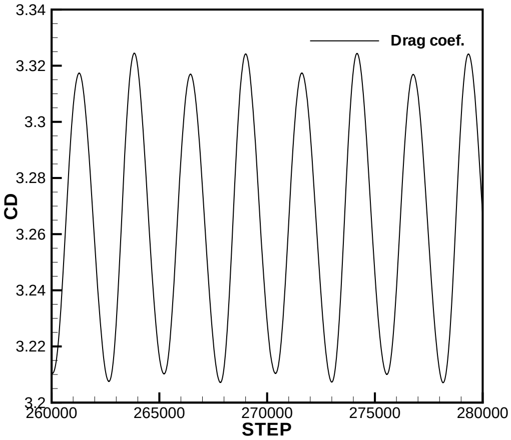

Figure 7: The time evolution of drag coefficients at Re=100.

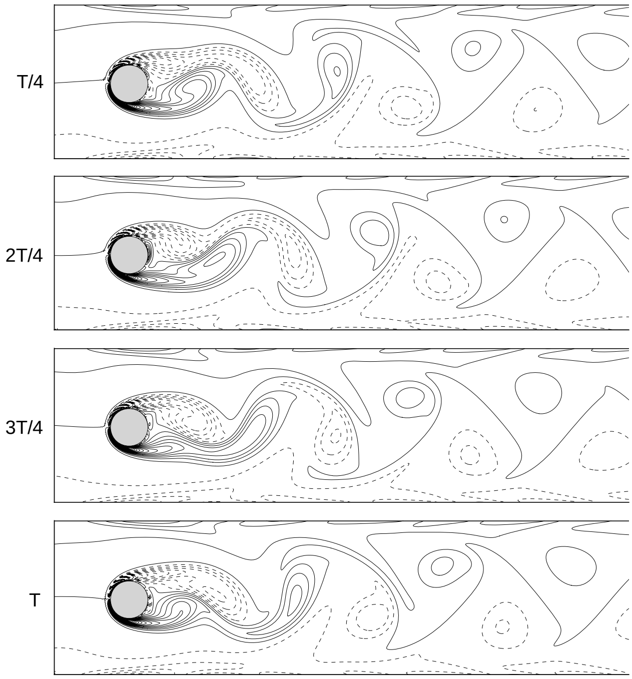

Figure 8: The instantaneous vorticity contours for a single period at Re=100, dotted and solid lines denote negative and positive contours.

sponding laboratory experiments reported in Dütsch, Durst, Becker, and Lienhart (1998), implying that qualitative features of the vorticity field dynamics can be properly captured by the present method.

To further compare the present numerical results with experimental data, velocity profiles at four x locations at 180^o phase angle are plotted in Fig. 11 along with corresponding experimental results from Dütsch, Durst, Becker, and Lienhart (1998). It is expected that the velocity distributions should be symmetric and antisymmetric for the u and v velocities, respectively, along the y direction. There is generally good agreement, although measurements show slightly nonsymmetric velocity distribution in the y direction, indicating the level of measurement uncertainty.

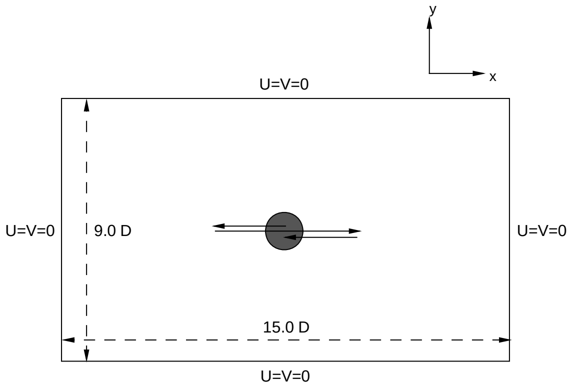

Figure 9: Configuration of In-line oscillating cylinder in a fluid at rest.

4 Conclusion

In this paper, a combination of the lattice Boltzmann method and the direct forcing based immersed boundary method is proposed to simulate flows with complex geometry. Here, the curved boundary is represented by a series of Lagrangian markers. The fluid velocity of node adjacent to the solid boundary is obtained by linear interpolation between the Lagrangian marker and the second fluid node further away. To improve the capability to model flow with moving object, a solid-body-forcing procedure is also employed within the solid region. The validity and accuracy of the new method are scrutinized by simulating decaying vortex, 2D flow over an asymmetrically placed cylinder, and in-line oscillating cylinder in a fluid at rest. The decaying vortex simulation indicates that the present method is second order accurate and all the results are compatible with the benchmark solutions.

Acknowledgement: The authors gratefully acknowledge the support by the Taiwan National Science Council (grant 94-2212-E-007-059) and the computational facilities provided by the Taiwan National Center for High-Performance Computing.

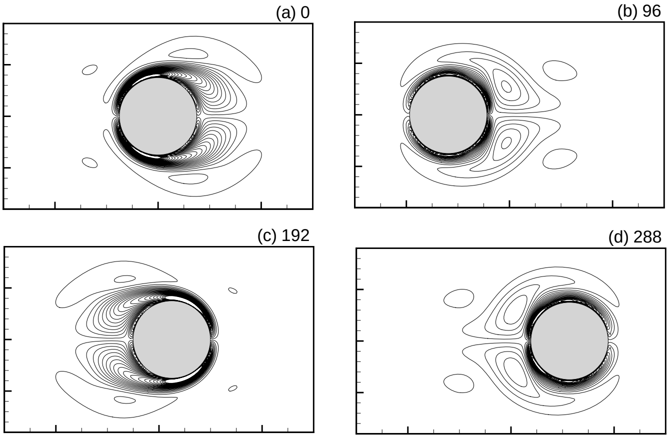

Figure 10: Vorticity contours at four different phase-angles. (a) 0°, (b) 96°, (c) 192°, and (d) 288°.

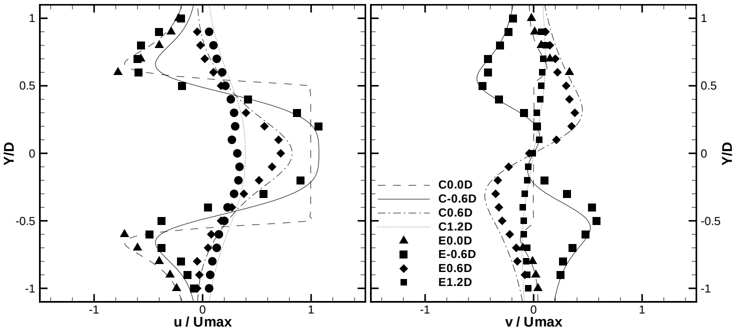

Figure 11: Velocity profiles at 180°. Lines are the present computational results, symbols are the experimental data in Dütsch, Durst, Becker, and Lienhart (1998).

References

Balaras, E. (2004): Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. *Comput. Fluids*, vol. 33, pp.

375–404.

Bouzidi, M.; Firdaouss, M.; Lallemand, P. (2001): Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Phys. Fluids*, vol. 13, pp. 3452–3459.

Chang, C.; Liu, C. H.; Lin, C. A. (2009): Boundary conditions for lattice Boltzmann simulations with complex geometry flows. *Computers & Mathematics with Applications*, vol. 58, no. 5, pp. 940–949.

Chen, D. J.; Lin, K. H.; Lin, C. A. (2007): Immersed boundary method based lattice Boltzmann method to simulate 2D and 3D complex geometry flows. *Int. J. Mod. Phys. C*, vol. 18, pp. 585–594.

Chen, S.; Doolen, G. D. (1998): Lattice Boltzmann method for fluid flow. *Annu. Rev. Fluid Mech.*, vol. 30, pp. 329–364.

Chen, S. Y.; Chen, H. D.; Martinez, D.; Matthaeus, W. (1991): Lattice Boltzmann model for simulation of magnetohydrodynamics. *Phys. Rev. Lett.*, vol. 67, pp. 3776–3779.

Chen, S. Y.; Martinez, D.; Mei, R. W. (1996): On boundary conditions in lattice Boltzmann methods. *Phys. Fluids*, vol. 8, pp. 2527–2536.

Dütsch, H.; Durst, F.; Becker, S.; Lienhart, H. (1998): Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan-Carpenter numbers. *J. Fluid Mech.*, vol. 360, pp. 249–271.

Fadlun, E. A.; Verzicco, R.; Orlandi, P.; Mohd-Yusof, J. (2000): Combined immersed-boundary finite-difference methods for tree-dimensional complex flow simulations. *J. Comput. Phys.*, vol. 161, pp. 35–60.

Filippova, O.; Hänel, D. (1998): Grid refinement for lattice-BGK models. *J. Comput. Phys*, vol. 147, pp. 219–228.

Guo, Z. L.; Zheng, C. G.; Shi, B. C. (2002): Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Phys. Rev. E*, vol. 65, no. 046308.

Ho, C. F.; Chang, C.; Lin, K. H.; Lin, C. A. (2009): Consistent boundary conditions for 2D and 3D lattice Boltzmann simulations. *CMES: Computer Modeling in Engineering & Sciences*, vol. 44, no. 2, pp. 137–155.

Kim, J.; Kim, D.; Choi, H. (2001): An immersed boundary finite-volume method for simulations of flow in complex geometries. *J. Comput. Phys*, vol. 171, pp. 132–150.

Ladd, A. J. C. (1994): Numerical simulation of particular suspensions via a discretized Boltzmann equation. Part II. Numerical results. *J. Fluid Mech.*, vol. 271, pp. 311–339.

Lallemand, P.; Luo, L. S. (2003): Lattice Boltzmann method for moving boundaries. *J. Comput. Phys.*, vol. 184, pp. 406–421.

Liao, C. C.; Chang, Y. W.; Lin, C. A.; McDonough, J. M. (2010): Simulating flows with moving rigid boundary using immersed boundary method. *Computers & Fluids*, vol. 39, no. 1, pp. 152–167.

Mei, R. W.; Luo, L. S.; Shyy, W. (1999): An accurate curved boundary treatment in the lattice Boltzmann method. *J. Comput. Phys*, vol. 155, pp. 307–330.

Mei, R. W.; Yu, D. Z.; Shyy, W.; Luo, L. S. (2002): Force evaluation in the lattice Boltzmann method involving curved geometry. *Phys. Rev. E*, vol. 65, no. 041203.

Mittal, R.; Iaccarino, G. (2005): Immersed boundary methods. *Annu. Rev. Fluid Mech.*, vol. 37, pp. 239–261.

Mohd-Yusof, J. (1997): *Combined immersed boundary/B-Spline method for simulations of flows in complex geometries*. CTR Annual Research Briefs, NASA Ames/Stanford University.

Peskin, C. S. (1972): Flow patterns around heart valves: a numerical method. *J. Comput. Phys.*, vol. 10, pp. 252–271.

Qian, Y. H.; d'Humieres, D.; Lallemand, P. (1992): Lattice BGK model for Navier-Stokes equation. *EPL*, vol. 17, pp. 479–484.

Saiki, E. M.; Biringen, S. (1996): Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. *J. Comput. Phys.*, vol. 123, pp. 450–465.

Schafer, M.; Turek, S. (1996): *Flow simulation with high-performance computer II," edited by E. H. Hirschel*, volume 52. Notes in Numerical Fluid Mechanics (Vieweg, Braunschweig).

Shih-Kai Chien, Tzu-Hsiang Yen, Y.-T. Y.; Chen, C.-K. (2008): Lattice Boltzmann Method Simulation of 3D Fluid Flow in Serpentine Channel. *CMES: Computer Modeling in Engineering & Sciences*, vol. 29, no. 3, pp. 163–174.

Su, S. W.; Lai, M. C.; Lin, C. A. (2007): An immersed boundary technique for simulating complex flows with rigid boundary. *Comput. Fluids*, vol. 36, pp. 313–324.

Yu, D. Z.; Mei, R. W.; Luo, L. S.; Shyy, W. (2003): Viscous flow computations with the method of lattice Boltzmann equation. *Progress in Aerospace Sciences*, vol. 39, no. 5, pp. 329–367.

Zou, Q. S.; He, X. Y. (1997): On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Phys. Fluids*, vol. 9, pp. 1591–1598.