

Interpolated Velocity Correction Immersed Boundary-Lattice Boltzmann Method for Fluid Flows with Flexible Boundary

Y. G. Chen and L. Wan

Abstract—An interpolated velocity correction scheme for the simulation of the interaction between fluid and flexible boundary using an immersed boundary-lattice Boltzmann method (IB-LBM) is proposed. In the conventional IB-LBM, the velocity field on the immersed boundary is determined by interpolating from an Eulerian grid to a Lagrangian grid using a discrete Dirac delta function, which is not divergence-free. As a result, this method can generally suffer from poor volume conservation for the closed immersed boundary. The key idea of the proposed interpolated velocity correction scheme is correcting the interpolated velocity field to satisfy a discrete divergence-free constraint defined on the Lagrangian boundary in the fluid. The proposed scheme makes no modifications to solve Navier-Stokes (N-S) equations using the lattice Boltzmann method (LBM) on the Eulerian grid and also improves volume conservation for the closed immersed boundary. Two examples are presented to verify the efficiency and accuracy of the proposed scheme.

Index Terms—Interpolated velocity, immersed boundary method, lattice Boltzmann method, fluid-structure interaction, Navier-Stokes equations.

I. INTRODUCTION

Problems involving the interaction of fluid with flexible structures are ubiquitous in nature. An IB-LBM was firstly presented by Feng *et al.* [1] to solve fluid-particles interaction problems, which is a combination and utilizes the most desirable features of the immersed boundary method (IBM) and the LBM. After that, this method has been developed and applied for the simulation of the flow involving rigid bodies [2]-[4] and elastic membranes [5]-[8], which might be an alternative and promising numerical scheme for simulating fluid-structure interaction (FSI) problems. The LBM originated from lattice gas automata is based on microscopic models and mesoscopic kinetic equations [9]. It is a regular lattice-based scheme for fluid flow simulation, which has been proved to be accurate and applicable to various hydrodynamic problems for its simplicity, efficiency, and parallelism characteristics [10]-[15]. The IBM was initially

proposed by Peskin [16] in 1970s for simulating cardiac mechanics and associated blood flow, which employs a fixed Eulerian grid for the flow field and a Lagrangian grid for an immersed boundary in the fluid, which is modeled by a singular force which is incorporated into the forcing term in the N-S equations. The interaction between the fluid and the immersed boundary is tackled using the IBM. A discrete Dirac delta function is used to spread the singular force from the Lagrangian grid to the Eulerian grid, and to interpolate the velocity from the Eulerian grid to the Lagrangian grid.

It is well known that the IBM can suffer from poor volume conservation for the closed immersed boundary in the fluid [17]-[21]. One cause of this lack of volume conservation is that the interpolated velocity field that determines the motion of the Lagrangian structure is not generally divergence-free, even if the Eulerian velocity is divergence-free with respect to the discrete divergence operator used in the numerical solution of the incompressible N-S equations. This problem was solved by modifying their divergence operator relative to their chosen interpolation operator, which would remain discretely divergence free on the Lagrangian grid when interpolated with their specific interpolation operator [18]. The immersed interface method [19] modifies the finite difference stencils of the fluid solver near the immersed boundary instead of utilizing discrete delta functions to spread the force from the Lagrangian to Eulerian grid. The Blob projection method [20] finds an analytic expression that represents the projection of a regularized form of the forces along the boundary onto the space of divergence-free vector fields. However, all these methods are difficult to implement and are not readily extendable to three dimensions for general problems, they also need more computationally expensive interpolation and spreading. In our work, when the LBM is used for solving the N-S equations, the velocity field on the Eulerian grid will be approximately divergence-free, but this does not guarantee that the interpolated velocity field where the immersed boundary moves is continuously divergence-free with the IBM. In this paper, we present a simple interpolated velocity correction scheme by correcting the interpolated velocity field to satisfy a discrete divergence-free constraint defined on the Lagrangian immersed boundary. This scheme can substantially improve volume conservation, which has no effect on solving the flow field on the Eulerian grid using the LBM. Our method only requires minor modification of existing codes and that has a negligible computational cost, and enforces interpolated velocity to maintain the incompressibility constraint.

Manuscript received January 12, 2015; revised April 24, 2015. This work was supported by the National Natural Science Foundation of China (Grant Nos. 11271295 and 10801109) and Science and Technology Research Projects of Hubei Provincial Department of Education (Grant No D20131602).

Y. G. Chen is with the School of Water Resources and Hydropower Engineering, Wuhan University, Wuhan 430072, P. R. China (e-mail: chenyingguang@wtu.edu.cn).

L. Wan and Y. G. Chen are with College of Mathematics and Computer Science, Wuhan Textile University, Wuhan 430073, P. R. China (e-mail: wanli@wtu.edu.cn).

II. NUMERICAL METHODS

A. Immersed Boundary Method

The IBM is defined by a set of differential equations involving a mixture of Eulerian and Lagrangian variables. For the viscous incompressible flows of Newtonian fluids in a two-dimensional domain Ω containing an immersed boundary Γ , the governing equation can be written as follows,

$$\rho (u_t + (u \cdot \nabla) u) = -\nabla p + \mu \Delta u + f \quad (1)$$

$$\nabla \cdot u = 0 \quad (2)$$

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

$$F(s, t) = S[X(s, t)] \quad (4)$$

$$\begin{aligned} \frac{\partial X(s, t)}{\partial t} &= u(X(s, t), t) \\ &= \int_{\Omega} u(s, t) \delta(x - X(s, t)) dx \end{aligned} \quad (5)$$

where x , u , ρ , p , μ and f are the Eulerian coordinates, fluid velocity vector, fluid density, fluid pressure, fluid viscosity and the force density term. $X(s, t) = (X(s, t), Y(s, t))$ is a vector function giving the location of points on Γ as function of arc-length (in some reference configuration) s , and time t , X and F stand for the Lagrangian coordinates and boundary force density. $\delta(x - X(s, t))$ is the Dirac delta function, more details are given in [17]. S is the force generation operator. Eq. (1) and (2) are the incompressible N-S equations. The Lagrangian and Eulerian force and velocity are related through Eq. (3) and (5). Eq. (4) is the constitutive law modeling the force generated from the configuration of the immersed boundary.

The constitutive law used for the force generation operator is to assume that the material behaves like an object under elastic tension. The force on the boundary is given by

$$F(s, t) = \frac{\partial}{\partial s} (T(s, t) \tau(s, t)) \quad (6)$$

$$T(s, t) = T_0 \left(\left\| \frac{\partial X(s, t)}{\partial s} \right\| - 1 \right) \quad (7)$$

$$\tau(s, t) = \frac{\partial X}{\partial s} / \left\| \frac{\partial X}{\partial s} \right\| \quad (8)$$

where T is the tension and τ is the unit tangent vector, s and is the arc-length measured along the current configuration of the boundary. The scalar T_0 is the stiffness constant which describes the elastic property of the boundary.

The Dirac delta function $\delta(r)$ appearing in Eq. (3) and (5) is used the common decomposition as follows,

$$\delta(r) = \phi(x) \phi(y) \quad (9)$$

$$\phi(r) = \begin{cases} \frac{1}{8} (3 - 2|r| - \sqrt{1 + 4|r| - 4r^2}), & 0 \leq |r| \leq 1, \\ \frac{1}{8} (5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}), & 1 \leq |r| \leq 2, \\ 0, & |r| \geq 2. \end{cases} \quad (10)$$

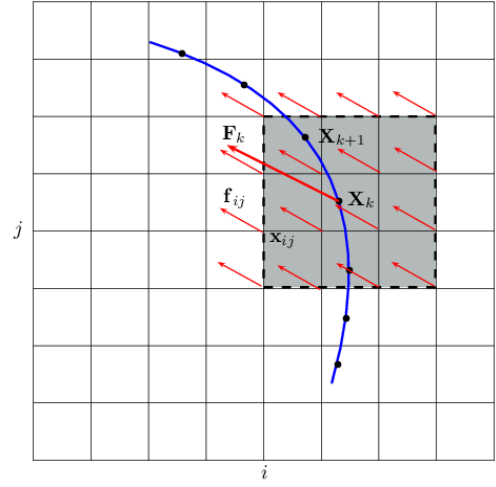


Fig. 1. The classification of grid nodes illustration. A cartesian grid for eulerian variables and a discrete set of points for the lagrangian variables

The above equations are solved on a pair of computational grids: a cell centered Cartesian grid for Eulerian variables, and a discrete set of points for the Lagrangian variables. As a 2D example is shown in Fig. 1. Assuming that the lower left corner of the domain is at the origin, the coordinates of the ij -th Eulerian grid points are $x_{ij} = (i\Delta x, j\Delta y)$ ($i = 0, 1, \dots, Nx$; $j = 0, 1, \dots, Ny$). Where $h = \Delta x = \Delta y$ is the fluid node spacing. Thus u_{ij} denotes the value of the variable u at x_{ij} , the discrete delta function $\delta_h(x, y) = (1/h^2)\phi(x/h)\phi(y/h)$. We use index n and k ($k = 0, 1, \dots, n_b$) to denote the value of a variable at a given time step and the position of Lagrangian grid points, thus $F_k^n = F_k(s, n\Delta t)$, $X_k^n = X_k(s, n\Delta t)$ and Δs_k is the boundary segment length. T_0 is a spring constant and L_0 is the resting length of the springs connecting immersed boundary points. Using this notation, temporally discretized forms of Eq. (3) (5) and (6) as follows:

$$f_{ij}^n = \sum_k F_k^n \delta_h(x_{ij} - X_k^n) \Delta s_k \quad (11)$$

$$\frac{X_k^{n+1} - X_k^n}{\Delta t} = u_k^{n+1} = \sum_{ij} u_{ij}^{n+1} \delta_h(x_{ij} - X_k^n) \Delta x \Delta y \quad (12)$$

$$F_k^n = \sum_{i=k-1, k+1} \frac{T_0}{\Delta s_{ik}} (\|X_i^n - X_k^n\| - L_0) \frac{1}{\Delta s_k} \frac{X_i^n - X_k^n}{\|X_i^n - X_k^n\|} \quad (13)$$

B. Lattice Boltzmann method

The LBM is a kinetic-based approach for simulating fluid flows. In this work, the lattice Boltzmann equation must be modified for solving the flow field with a force density. The form of LBE proposed by Guo et al [22] is adopted, which can be written as

$$f_i(x + e_i \Delta t, t + \Delta t) - f_i(x, t) = -\frac{1}{\tau} [f_i(x, t) - f_i^{eq}(x, t)] + F_i \Delta t \quad (14)$$

$$F_i = \left(1 - \frac{1}{2\tau}\right) \omega_i \left(\frac{e_i \cdot u}{c_s^2} + \frac{e_i \cdot u}{c_s^4} e_i \right) f \quad (15)$$

$$\rho = \sum_i f_i, u = \frac{1}{\rho} \left(\sum_i e_i f_i + \frac{1}{2} f \Delta t \right), p = \rho c_s^2 \quad (16)$$

where f_i is the distribution function for particles with velocity e_i at position x and time t , f_i^{eq} is the equilibrium distribution function, τ is the dimensionless relaxation parameter, F_i is the forcing term, c_s is the sound speed, and equals $\sqrt{1/3} \Delta x / \Delta t$.

ω_i is the lattice weights, depending on the lattice structure. The lattice structure, the corresponding velocities and the lattice weights are described in [23]. The relaxation parameter is connected to the speed of sound and the kinematic viscosity in the N-S equation in the form of $\nu = (\tau - 1/2) c_s^2 \Delta t$. When the particle density distribution is known, the fluid density, velocity and pressure can be calculated by formula (16).

C. The Interpolated Velocity Correction Algorithms

In order to derive the N-S equations from lattice Boltzmann equation (LBE), the Chapman-Enskog expansion [24] is used. If ρ is a constant, macroscopic physics equation derived from lattice Boltzmann equation is the standard incompressible N-S equations. When the LBM is used for solving the N-S equations on the Eulerian grid, the velocity field will be approximately divergence-free. The computed velocity field on the Eulerian grid is divergence-free, but this velocity is then interpolated to the Lagrangian grid with interpolation operators that ignore the divergence-free constraint. This can result in the volume of a closed pressurized object shrinking. The idea of our scheme is to alter the interpolation operator to maintain the incompressibility constraint. We weaken Eq. (2) to integral form:

$$\int_{\Omega} \nabla \cdot u dV = 0 \quad (17)$$

However, what we really care about is the region defining Ω_{in} interior to the immersed boundary, applying the divergence theorem we obtain

$$\int_{\Omega_{in}} \nabla \cdot u dV = \int_{\Gamma} u \cdot n ds = 0 \quad (18)$$

where n is the outward unit normal to the boundary Γ . Above equation is discretized only on the Lagrangian boundary, so we obtain constraint

$$\sum_k u_k \cdot n_k \Delta S_k = 0 \quad (19)$$

where ΔS_k is some discretization of arc-length in the current configuration. Interpolating velocities from the Eulerian to the Lagrangian grid by using formula (12), we can obtain velocity u_k^{n+1} of the Lagrangian grid point on the immersed boundary. u_k^{n+1} is not meet the constraint (19) this moment, so we enforce correction velocity \bar{u}_k^{n+1} to satisfy the constraint and then assume the equation

$$\bar{u}_k^{n+1} = u_k^{n+1} - \ell n_k \quad (20)$$

where ℓ is an undetermined parameter. It is easy to derive ℓ from the condition that \bar{u}_k^{n+1} satisfy the constraint (19). Finally, we can obtain

$$\ell = \frac{1}{\sum_k \Delta S_k} \sum_k u_k^{n+1} \cdot n_k \Delta S_k \quad (21)$$

When update interpolated velocity \bar{u}_k^{n+1} using the above correction scheme, we need not to modify LBE. This scheme has no affect on solving N-S equations on Eulerian grid using LBM. Furthermore, these attempts only require minor modification of existing codes and that have a negligible computational cost, and enforce interpolated velocity to maintain the incompressibility constraint.

III. NUMERICAL RESULTS

A. An elastic Pressurized Membrane

To verify the effectiveness and accuracy of the proposed scheme, an elastic pressurized membrane immersed in a viscous fluid is simulated [19], [25] and [26]. We use a dimensionless system in the simulation. The initial immersed boundary is an ellipse with major and minor axes $r_a = 0.75$, $r_b = 0.5$, respectively. The unstretched immersed boundary is a circle with radius $r_0 = 0.5$. Due to the restoring force, the ellipse will converge to an equilibrium circle with radius $r_e \approx 0.61237$; this is larger than the unstretched interface because of the incompressibility of the closed fluid. So the membrane is still stretched at the equilibrium state, and the nonzero boundary force is balanced by a nonzero jump in the pressure. The computational domain is $[-1, 1] \times [-1, 1]$, and we set $\rho_0 = 1.0$ and $u|_{\partial\Omega} = 0$ throughout the domain. In order to record the process of flow variables, points $a(-0.5, 0.0)$, $b(0.0, -0.5)$ and $c(0.0, 0.0)$ are set in the flow field. In addition, $N_x \times N_y = 200 \times 200$ and $n_b = 1199$ are adopted to conduct the simulation. For the solution of this problem, we chose each lattice simulation time step, say 2.0×10^{-4} . We will compare the corrected and uncorrected interpolated velocity field results using IB-LBM. In order to compare results over longer times, let r_x be the radius of the boundary measured along the x-axis and r_y the radius of the boundary measured along the y-axis. These should both oscillate and converge towards the common value r_e as the oscillation is

damped by viscosity and the membrane becomes more circular. Similarly, let A be the closed area of the boundary, which should converge to initial ellipse area $A_0 = \pi r_a r_b \approx 1.17810$ at the equilibrium time.

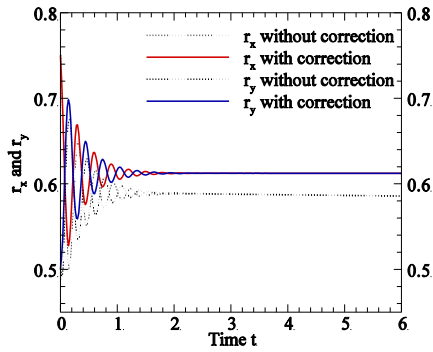


Fig. 2. The evolution of r_x and r_y with and without interpolated velocity correction algorithm

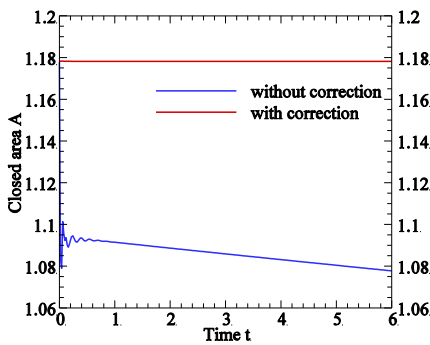


Fig. 3. The evolution of the closed area A with and without interpolated velocity correction algorithm.

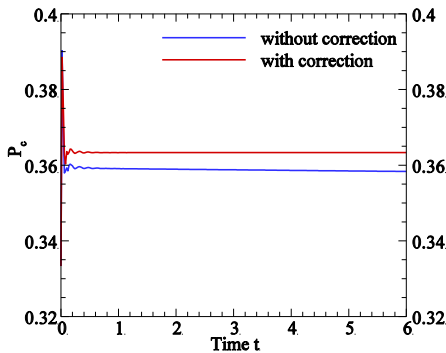


Fig. 4. The evolution of the pressure P_c at point c without and with interpolated velocity correction algorithm.

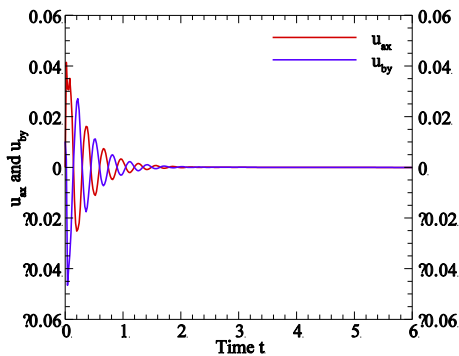


Fig. 5. The evolution of flow velocities component u_{ax} at point a and u_{by} at point b with interpolated velocity correction algorithm.

In our test, we take $v = 0.1$ and $T_0 = 10.0$. Fig. 3 shows the evolution of r_x and r_y computed by both with and without

interpolated velocity correction algorithm respectively. We can know that the membrane cannot reach the equilibrium, r_x and r_y have been shrinking as the evolution of time without correction algorithm. Fig. 2 also clearly shows that both r_x and r_y converge to r_e as the evolution of time with presented correction algorithm. Furthermore, the closed area A has only a very small decrease at the beginning, it converge to a constant closed to the initial area A_0 as the evolution of time in Fig. 3, but A has been reducing as the evolution of time without correction algorithm. Fig. 4 shows inside pressure P_c at point c with and without interpolated velocity correction algorithm respectively. we can find that the pressure P_c has been declining as the evolution of time from Fig. 4 without correction, but it shows the pressure P_c maintain unchanged with correction scheme. Fig. 5 shows flow velocities component u_{ax} at point a and u_{by} at point b as the evolution of time with presented correction algorithm. We can find that u_{ax} and u_{by} are both convergence to 0 as the evolution of time.

In the equilibrium state, the boundary is a circle, the pressure has two different constant values inside and outside the boundary, as relative pressure $\Delta p = p - 1/3$ is shown with correction algorithm in Fig. 6. The pressure distribution in the equilibrium state shows the sharp pressure jump across the boundary. At the same time, Fig. 7 shows slice of the pressure at time $t = 4$, which is the pressure distribution along the x axis at $y = 0$. At equilibrium, we can know that the velocity is zero and the pressure is piecewise constant inside and outside the boundary with a jump across the immersed boundary.

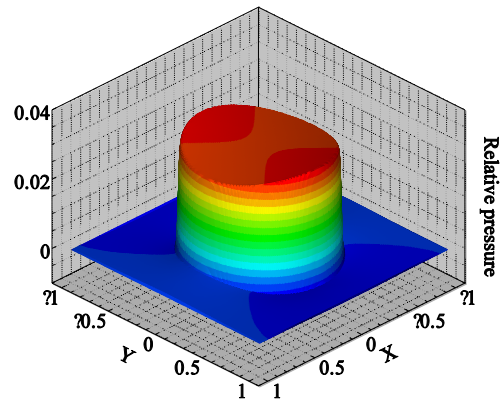


Fig. 6. The relative pressure distribution in the equilibrium state with interpolated velocity correction algorithm.

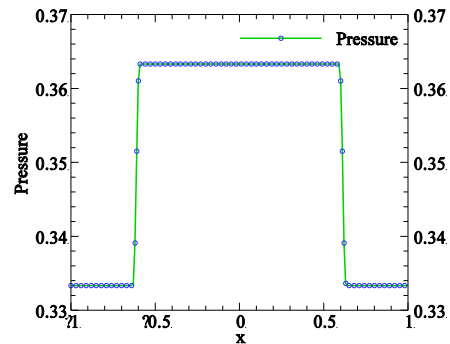


Fig. 7. The pressure distribution along the x -axis at $y = 0$ at time $t = 4$.

Furthermore, we also tested the above example with interpolated velocity correction scheme when parameters

change. We simulated the problem for $\nu = 1.0$, $T_0 = 1.0$ and $\nu = 0.002$, $T_0 = 0.02$, which still adopted the same Eulerian and Lagrangian grids as the former. Table 1 lists some numerical results of A , r_x , r_y and P_c at the equilibrium state. From Table 1 we know that the numerical results agree very well with the analytical solution. The maximum relative volume error less than 0.00509%, the maximum relative radius errors less than 0.00327%. Table I indicates a quite good volume conservation characteristic when the Reynolds number is smaller or bigger. All of the results show that the interpolated velocity correction algorithm has a good adaptability and accuracy.

TABLE I: NUMERICAL RESULTS OF A , R_x , R_y AND P_c AT THE EQUILIBRIUM TIME T WITH INTERPOLATED VELOCITY CORRECTION SCHEME WHEN PARAMETERS N AND T_0 CHANGE

ν	T_0	t	A	r_x	r_y	P_c
1.0	1.0	6.22	1.17816	0.61239	0.61239	0.336330
0.1	10.0	4	1.17813	0.61238	0.61238	0.363302
0.002	0.02	188	1.17810	0.61237	0.61237	0.333393
Analytical	-	-	1.17810	0.61237	0.61237	-
Maximum difference	-	-	0.00509%	0.00327%	0.00327%	-

TABLE II: GRID REFINEMENT ANALYSIS OF INTERPOLATED VELOCITY CORRECTION SCHEME WITH $N=0.1$, $T_0=10.0$ AT TIME $T=1.6$

$N_x \times N_y, n_b$	$ A - A_0 $	Order
$32 \times 32, 191$	0.00128	-
$64 \times 64, 381$	0.00031	2.04580
$128 \times 128, 767$	0.00007	2.14684
$256 \times 256, 1536$	0.00002	1.80735

Finally, we perform a grid refinement analysis at time $t = 1.6$ to study the conservation of the area closed by immersed boundary. Table II shows the results of a grid refinement study using the correction algorithm where the values on four different $N_x \times N_y$ grids with 32×32 , 64×64 , 128×128 and 256×256 are compared with the analytical solution. We can know that the area is conserved with second-order accuracy using our correction scheme.

B. A Moving Membrane in a Lid-Driven Cavity Flow

As another example, we put the same pressurized circular membrane as the first example in the center of a lid-driven cavity flow, with zero initial fluid velocity. The lid is moving from left to right with speed 0.1 and the other three walls are fixed with no-slip boundary conditions. This problem is computed with $\nu = 0.1$ and stiffness constant $T_0 = 1.0$. When the lid starts moving, the viscosity will drive the fluid motion.

The membrane will also be driven and distorted by the flow, although the membrane tension tends to keep the membrane circular. Fig. 8 shows the solution at several times, as computed with the interpolated velocity correction algorithm on a fine grid $N_x \times N_y = 200 \times 200$ and $n_b = 1199$. For the solution of this problem, we chose each lattice simulation time step, say 2.0×10^{-4} . We also computed the relative volume error at different time. The maximum relative volume error less than 0.727442% for the moving membrane in the lid-driven cavity flow. The result indicates volume conservation for the closed moving membrane immersed in fluid with our correction scheme.

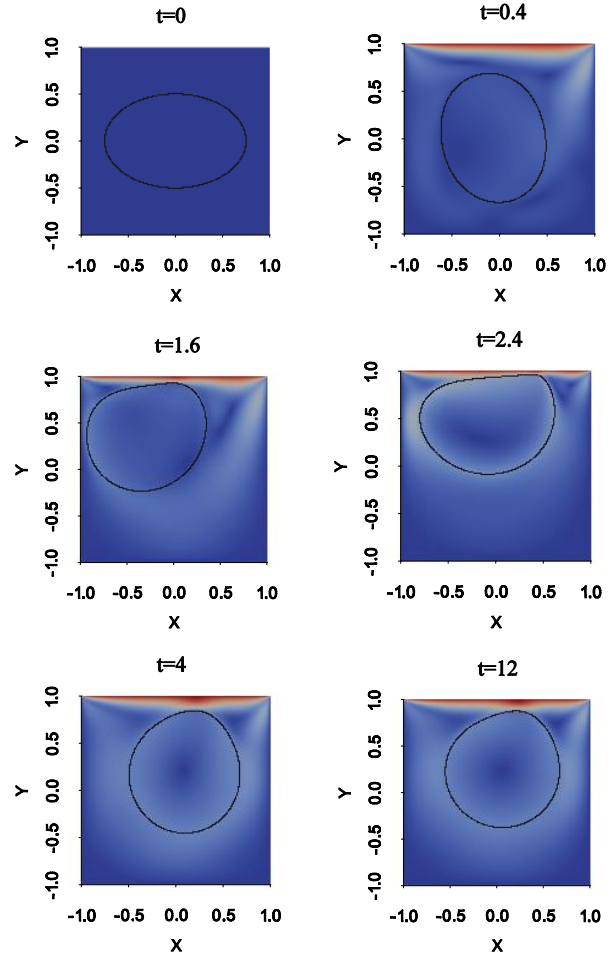


Fig. 8. Membrane location at different times.

IV. CONCLUSION

We have simulated the interaction between fluid and flexible boundary using IB-LBM with the presented interpolated velocity correction scheme. The implementation has been tested with two examples that are an elastic pressurized membrane immersed in a viscous fluid and a moving membrane in a lid-driven cavity. The described method can capture the process of fluid flow with flexible boundaries. Some results verify the effectiveness and accuracy of the proposed scheme. The work is to further extend to three dimensions, which will give some help for large scale simulation involving flexible boundaries in the future.

REFERENCES

- [1] Z. G. Feng and E. E. Michaelides, "The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems," *J. Comput. Phys.*, vol. 195, pp. 602-628, 2004.
- [2] C. Shu, N. 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.*, vol. 226, pp. 1607-1622, 2007.
- [3] J. Wu and C. Shu, "Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications," *J. Comput. Phys.*, vol. 228, pp. 1963-1979, 2009.
- [4] J. Wu and C. Shu, "An improved immersed boundary-lattice Boltzmann method for simulating three-dimensional incompressible flows," *J. Comput. Phys.*, vol. 229, pp. 5022-5042, 2010.
- [5] D. V. Le, J. White, J. Peraire, K. M. Lim, and B. C. Khoo, "An implicit immersed boundary method for three-dimensional fluid-membrane interactions," *J. Comput. Phys.*, vol. 228, pp. 8427-8445, 2009.

- [6] T. Kruger and F. Varnik, and D. Raabe, "Efficient and accurate simulations of deformable particles immersed in a fluid using a combined immersed boundary lattice Boltzmann finite element method," *Comput. Math. Appl.*, vol. 61, pp. 3485-3505, 2011.
- [7] J. Hao and L. Zhu, "A lattice Boltzmann based implicit immersed boundary method for fluid-structure interaction," *Comput. Math. Appl.*, vol. 59, pp. 185-193, 2010.
- [8] L. Zhu, G. He, S. Wang, L. Miller, X. Zhang, Q. You, and S. Fang, "An immersed boundary method by the lattice Boltzmann approach in three dimensions with application," *Comput. Math. Appl.*, vol. 61, pp. 3506-3518, 2010.
- [9] S. Chen and G. D. Doolen, "Lattice Boltzmann method for fluid flows," *Ann. Rev. Fluid Mech.*, vol. 30, pp. 329-364, 1998.
- [10] H. W. Zheng, C. Shu, and Y. T. Chew, "A lattice Boltzmann model for multiphase flows with large density ratio," *J. Comput. Phys.*, vol. 218, pp. 353-371, 2006.
- [11] C. X. Pan, L. S. Luo, and C. T. Miller, "An evaluation of lattice Boltzmann schemes for porous medium flow simulation," *Comput. Fluids.*, vol. 35, pp.898-909, 2006.
- [12] S. Chen, H. D. Chen, D. Martinez, and W. Matthaeus, "Lattice Boltzmann model for simulation of magneto hydrodynamics," *Phys. Rev. Lett.*, vol. 67, pp. 3776-3779, 1991.
- [13] C. K. Aidun and Y. N. Lu, "Lattice Boltzmann simulation of solid particles suspended in fluid," *J. Stat. Phys.*, vol. 81, pp. 49-61, 1995.
- [14] R. Zhang, X. Shan, and H. Chen, "Efficient kinetic method for fluid simulation beyond the Navier-Stokes equation," *Phys. Rev. E.*, vol. 74, p. 46703, 2006.
- [15] Y. B. Gan, A. G. Xu, G. C. Zhang, X. J. Yu, and Y. Li, "Two-dimensional lattice Boltzmann model for compressible flows with high Mach number," *Phys. A.*, vol. 387, pp. 1721-1732, 2008.
- [16] C. S. Peskin, "Numerical analysis of blood flow in the heart," *J. Comp. Phys.*, vol. 25, pp. 220-252, 1977.
- [17] C. S. Peskin, "The immersed boundary method," *Acta Numer.*, vol. 11, no. 2, pp. 479-517, 2002.
- [18] C. Peskin and B. Printz "Improved volume conservation in the computation of flows with immersed elastic boundaries," *J. Comp. Phys.*, vol. 105, pp. 33-46, 1993.
- [19] R. J. L. Veque and Z. Li, "Immersed interface method for Stokes flow with elastic boundaries or surface tension," *SIAMJ. Sci. Comput.*, vol. 18, p. 709, 1997.
- [20] R. Cortez and M. Minion, "The Blob Projection Method for Immersed Boundary Problems," *J. Comput. Phys.*, vol. 161, pp. 428-453, 2000
- [21] E. P. Newren, "Enhancing the immersed boundary method: stability, volume conservation, and implicit solvers," PhD thesis, University of Utah, 2007.
- [22] Z. Guo, C. Zheng, and B. Shi, "Discrete lattice effects on the forcing term in the lattice Boltzmann method," *Phys. Rev. E.*, vol. 65, p. 46308, 2002.
- [23] Y. H. Qian, D. Humieres and P. Lallemand, "Lattice BGK model for Navier-Stokes equation," *Europhys. Lett.*, vol. 17, pp. 479-484, 1992.
- [24] S. Chapman and T. G. Cowling, *The Mathematical Theory of Nonuniform Gases*, Cambridge: Cambridge University Press, 1970.
- [25] L. Lee and R. J. L. Veque, "An immersed interface method for incompressible Navier-Stokes equations," *SIAM J. Sci. Comput.*, vol. 25, no. 3, pp. 832-856, 2003.
- [26] Y. G. Cheng and H. Zhang, "Immersed boundary method and lattice Boltzmann method coupled FSI simulation of mitral leaflet flow," *Comput. & Fluids*, vol. 39, no. 5, pp. 871-881, 2010.



Y. G. Chen received his M.Sc in computational mathematics from Chongqing University, China. He is a Ph.D student in School of Water Resources and Hydropower Engineering at Wuhan University. He is currently studying the fluid dynamics using a lattice Boltzmann-immersed boundary method. He is also currently an assistant professor of College of Mathematics and Computer Science at Wuhan Textile University.