Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **6**, No. 4, pp. 436-460

DOI: 10.4208/aamm.2014.4.s2
August 2014

Development of Lattice Boltzmann Flux Solver for Simulation of Incompressible Flows

C. Shu[1,*], Y. Wang[1], C. J. Teo[1] and J. Wu[2]

[1] *Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260*

[2] *Department of Aerodynamics, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*

Received 9 February 2014; Accepted (in revised version) 10 May 2014

Available online 28 May 2014

Abstract. A lattice Boltzmann flux solver (LBFS) is presented in this work for simulation of incompressible viscous and inviscid flows. The new solver is based on Chapman-Enskog expansion analysis, which is the bridge to link Navier-Stokes (N-S) equations and lattice Boltzmann equation (LBE). The macroscopic differential equations are discretized by the finite volume method, where the flux at the cell interface is evaluated by local reconstruction of lattice Boltzmann solution from macroscopic flow variables at cell centers. The new solver removes the drawbacks of conventional lattice Boltzmann method such as limitation to uniform mesh, tie-up of mesh spacing and time interval, limitation to viscous flows. LBFS is validated by its application to simulate the viscous decaying vortex flow, the driven cavity flow, the viscous flow past a circular cylinder, and the inviscid flow past a circular cylinder. The obtained numerical results compare very well with available data in the literature, which show that LBFS has the second order of accuracy in space, and can be well applied to viscous and inviscid flow problems with non-uniform mesh and curved boundary.

AMS subject classifications: 20B40

Key words: Chapman-Enskog analysis, flux solver, incompressible flow, Navier-Stokes equation, lattice Boltzmann equation.

1 Introduction

Currently, for the simulation of incompressible viscous flows, most of numerical solvers can be roughly classified into two categories. One is based on the solution of Navier-Stokes (N-S) equations, while the other is based on the solution of lattice Boltzmann

[*] Corresponding author.
Email: mpeshuc@nus.edu.sg (C. Shu)

equation (LBE). N-S equations are from the application of mass and momentum conservation laws to a control volume. They have strong physical backgrounds. LBE is from the statistical physics. Both N-S solvers and LBE solvers have their distinguished features.

In the category of incompressible N-S solvers, the dependent variables are the macroscopic pressure and velocity. One approach in this category is the artificial compressibility method [1]. This method adds a weak compressibility into the incompressible N-S equations so that the well-established compressible N-S solvers can be applied to simulate incompressible flows. Its drawback is that the artificial compressibility involves a user-specified parameter, which may not be easy to give for some cases. The most popular solver in this category is to solve incompressible N-S equations directly. However, unlike compressible N-S equations, there is no transport equation for pressure in the incompressible N-S equations. In fact, the pressure is only appeared in the momentum equation but the velocity is involved in both the continuity and momentum equations. When the velocity is obtained from the momentum equation, there is no guarantee that it will satisfy the continuity equation. To overcome this difficulty, a number of algorithms [2–9], which are termed projection or pressure correction methods, have been proposed. These methods mainly resolve the coupling problem between the pressure field and the velocity field through the fractional step process. Usually, the process involves the solution of Poisson equation for pressure or pressure correction. The slow convergence of Poisson equation degrades the computational efficiency of this kind of N-S solvers, especially for unsteady flow simulation. In addition, to properly consider the effect of pressure oscillation in the numerical simulation, the staggered grid, on which the velocity components and pressure are defined at different locations, is often adopted. The use of staggered grid brings a great inconvenience in programming. Furthermore, as N-S equations are partial differential equations, N-S solvers need to use numerical schemes such as finite difference (FD), finite volume (FV) and finite element (FE) methods to discretize the first and second order spatial derivatives, and solve the resultant ordinary differential equations or algebraic equations. It is not a trivial job.

In contrast, LBE is a discrete model. At each physical location, a finite number of fictitious particles with given velocity (provided by lattice velocity model) are distributed. The density distribution functions of these particles are taken as unknowns, which can be determined from a set of algebraic equations (lattice Boltzmann equations). Once the density distribution functions are known at a physical location, the macroscopic flow variables such as density and velocity can be computed from mass and momentum conservation. LBE was initially developed by Chen et al. [10] and Qian et al. [11]. Since then, many variants of LBE have been developed in the literature [12–24]. Basically, LBE solver has two processes: streaming and collision. The streaming process involves particle distribution functions at two physical locations while the collision process happens locally. The collision process can be approximated by a linear model with a single relaxation time (BGK model) [10,11] or multi-relaxation times (MRT model) [16]. As compared with N-S solvers, the LBE solver has following distinguished features. Firstly, the linear streaming and collision processes of fictitious particles in the LBE solver can effectively consider the nonlinear convection and diffusion effects at the macroscopic level. The manipulation of complicated nonlinear terms and treatment of high order derivatives in N-S solvers are avoided. Secondly, LBE is a set of algebraic formulations, which can be easily solved. No differential equation and solution of resultant algebraic equations are involved in the LBE solver. These appealing features attract more and more researchers to apply LBE for solving various flow problems [25–36]. On the other hand, we have to indicate that LBE solvers also suffer from some drawbacks. Due to uniformity of the lattice, the standard LBE solver is limited to the simple geometry and uniform mesh. For complex geometry and application on the non-uniform mesh, additional efforts such as supplemented interpolation [12] and built-in interpolation [19] have to be incorporated. This will increase the complexity of the solver, and requires additional computational effort and virtual storage. The second drawback is the tie-up between the time interval and mesh spacing due to the streaming process. This drawback makes the adaptive and multi-block computation of LBE solvers extremely complicated. In addition, LBE solvers need more memory to store density distribution functions than the N-S solvers. Another drawback is that LBE solvers can only be applied to simulate viscous flows but N-S solvers can be applied to solve both inviscid and viscous flows. Furthermore, the physical boundary conditions such as given velocity and pressure cannot be implemented directly in the LBE solver. They have to be converted to the boundary conditions for density distribution functions. Although bounce back rule is an efficient way to implement no-slip boundary condition, other boundary conditions may not be implemented in a simple way, especially for the three-dimensional case as it has more than 15 lattice directions.

From the above discussion, both N-S solvers and LBE solvers have their distinctive advantages and disadvantages for simulation of incompressible flows. One may ask whether we can develop a solver to combine their advantages, and in the meantime, to remove their drawbacks. This motivates the present work. To address this issue, we need to look at the relationship between N-S equations and LBE as they are mathematical models to describe the same physical problem. Indeed, the Chapman-Enskog (C-E) expansion analysis [13] is a bridge to link the two solvers, from which the macroscopic variables and fluxes in N-S equations can be computed by density distribution functions in the LBE solution. Usually, the C-E analysis is applied in the whole flow domain and at any time level. This lays the foundation that the macroscopic flow variables obtained by LBE solvers at any physical location and any time level can satisfy the N-S equations. On the other hand, it was found that the C-E analysis can be applied at a local position within a small streaming step. In fact, this idea has been well applied by Xu and his co-workers [37–40] in the development of gas kinetic scheme, where the flux at the cell interface is computed by local application of Boltzmann equation. Inspired by the work of Xu [37], in this work, the lattice Boltzmann flux solver is presented. In the solver, the finite volume method is applied to discretize the governing differential equations (N-S equations), and the flow variables at the cell center are obtained by marching in time. The fluxes at the cell interface are evaluated by local reconstruction of LBE solution with the help of C-E analysis. The present solver effectively combines the advantages of both N-S solvers and LBE solvers, and in the meantime removes their disadvantages. To be specific, only macroscopic flow variables are stored and used as dependent variables, and physical boundary conditions can be directly implemented. There is no need to approximate the second order derivatives and use staggered grid. The convective and viscous fluxes are computed simultaneously. Furthermore, the above mentioned drawbacks of LBE solvers such as limitation of simple geometry and uniform mesh, tie-up between the time interval and mesh spacing, and limitation to the viscous flow are all removed. As shown by test cases in the paper, the present solver can be effectively applied to solve both viscous and inviscid flows with the curved boundary or non-uniform mesh. It has more flexibility for applications.

2 Navier-Stokes equations, lattice Boltzmann equation and Chapman-Enskog expansion analysis

2.1 Navier-Stokes (N-S) equations

When we apply the physical conservation laws of mass and momentum to a control volume, the following continuity and momentum equations, which are often named Navier-Stokes equations can be derived

$$\frac{\partial\rho}{\partial t}+\nabla\cdot\rho\mathbf{u}=0, \tag{2.1a}$$

$$\frac{\partial\rho\mathbf{u}}{\partial t}+\nabla\cdot(\rho\mathbf{u}\mathbf{u})=-\nabla p+\mu\nabla\cdot\left[\nabla\rho\mathbf{u}+(\nabla\rho\mathbf{u})^T\right], \tag{2.1b}$$

where ρ is the fluid density, $\mathbf{u}$ is the flow velocity and p is the pressure. If the density variation is small and Mach number is low, the above governing equations can be used to simulate incompressible flows.

2.2 Lattice Boltzmann equation (LBE)

The standard lattice Boltzmann equation with BGK approximation can be written as

$$f_\alpha(\mathbf{r}+\mathbf{e}_\alpha\delta_t,t+\delta_t)=f_\alpha(\mathbf{r},t)+\frac{f_\alpha^{eq}(\mathbf{r},t)-f_\alpha(\mathbf{r},t)}{\tau},\quad \alpha=0,1,\cdots,N, \tag{2.2}$$

where $\mathbf{r}$ represents a physical location, τ is the single relaxation parameter; f_α is the density distribution function along the α direction; f_α^{eq} is its corresponding equilibrium state; δ_t is the streaming time step and $\mathbf{e}_\alpha$ is the particle velocity in the α direction; N is the number of discrete particle velocities. Once the density distribution functions at the physical location $\mathbf{r}$ are obtained, the macroscopic density ρ and momentum $\rho\mathbf{u}$ are computed by

$$\rho=\sum_{\alpha=0}^{N} f_\alpha, \tag{2.3a}$$

$$\rho\mathbf{u}=\sum_{\alpha=0}^{N} f_\alpha \mathbf{e}_\alpha. \tag{2.3b}$$

The pressure can be calculated from the equation of state,

$$p=\rho c_s^2, \tag{2.4}$$

where c_s is the sound speed. In the application of LBE (2.2), the lattice velocity $\mathbf{e}_\alpha$ has to be given first. There are a number of lattice velocity models for the two-dimensional (2D) and three-dimensional (3D) cases. For the 2D case, the most popular lattice velocity model is D2Q9 model defined in a square lattice, which can be written as

$$\mathbf{e}_\alpha=\begin{cases} 0, & \alpha=0,\\ (\cos[(\alpha-1)\pi/2],\sin[(\alpha-1)\pi/2])c, & \alpha=1,2,3,4,\\ \sqrt{2}(\cos[(\alpha-5)\pi/2+\pi/4],\sin[(\alpha-5)\pi/2+\pi/4])c, & \alpha=5,6,7,8. \end{cases} \tag{2.5}$$

Here $c=\delta_x/\delta_t$, δ_x is the lattice spacing. For the case of $\delta_x=\delta_t$, which is often used in the literature and also adopted in this work, c is taken as 1. There are two key issues in solving LBE (2.2). One is the specification of equilibrium distribution function f_α^{eq}, and the other is the determination of relaxation parameter τ. The equilibrium distribution function can be given from the truncated Taylor series expansion of Maxwellian function in terms of Mach number, which reads

$$f_\alpha^{eq}(\mathbf{r},t)=\rho w_\alpha\left[1+\frac{\mathbf{e}_\alpha\cdot\mathbf{u}}{c_s^2}+\frac{(\mathbf{e}_\alpha\cdot\mathbf{u})^2-(c_s|\mathbf{u}|)^2}{2c_s^4}\right], \tag{2.6}$$

where the coefficients w_α and the sound speed c_s depend on the lattice velocity model. For the D2Q9 model given by Eq. (2.5), they are given as: $w_0=4/9$, $w_1=w_2=w_3=w_4=1/9$ and $w_5=w_6=w_7=w_8=1/36$, $c_s=c/\sqrt{3}$. The relaxation parameter τ is linked to the kinematic viscosity of fluid through Chapman-Enskog expansion analysis by the following relationship

$$v=\left(\tau-\frac{1}{2}\right)c_s^2\delta_t. \tag{2.7}$$

Note that in the lattice Boltzmann method (LBM), the equilibrium distribution functions also satisfy the conservation of mass and momentum at any physical location. Thus, we have

$$\rho=\sum_{\alpha=0}^{N} f_\alpha^{eq}, \tag{2.8a}$$

$$\rho\mathbf{u}=\sum_{\alpha=0}^{N} f_\alpha^{eq}\mathbf{e}_\alpha. \tag{2.8b}$$

2.3 Chapman-Enskog expansion analysis

As discussed in the introduction, both N-S equations (2.1a)-(2.1b) and LBE (2.2) are mathematical models to describe the same physical problem (fluid flow). They should have some relationship. Indeed, their relationship is given by Chapman-Enskog expansion analysis [13]. Some basic formulations of this analysis are shown below.

By introducing multi-scale expansion, the density distribution function, the temporal derivative and the spatial derivative can be expanded respectively as

$$f_\alpha = f_\alpha^{(0)} + \varepsilon f_\alpha^{(1)} + \varepsilon^2 f_\alpha^{(2)}, \tag{2.9a}$$

$$\frac{\partial}{\partial t} = \varepsilon \frac{\partial}{\partial t_0} + \varepsilon^2 \frac{\partial}{\partial t_1}, \tag{2.9b}$$

$$\nabla_r = \varepsilon \nabla_{r1}, \tag{2.9c}$$

where ε is a small parameter proportional to the Knudsen number. By performing Taylor series expansion in time and space for Eq. (2.2), the following differential equation with the second order of accuracy is obtained,

$$\Big(\frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla\Big) f_\alpha + \frac{\delta_t}{2}\Big(\frac{\partial}{\partial t} + \mathbf{e}_\alpha \cdot \nabla\Big)^2 f_\alpha + \frac{1}{\tau \delta_t}\big(f_\alpha - f_\alpha^{eq}\big) + \mathcal{O}\big(\delta_t^2\big) = 0. \tag{2.10}$$

Substituting Eq. (2.9) into Eq. (2.10) gives the following 3 equations in terms of ε order,

$$\mathcal{O}(\varepsilon^0): \qquad ({f_\alpha}^{(0)} - {f_\alpha}^{eq})/(\tau\delta_t) = 0, \tag{2.11a}$$

$$\mathcal{O}(\varepsilon): \qquad \Big(\frac{\partial}{\partial t_0} + \mathbf{e}_\alpha \cdot \nabla_1\Big) f_\alpha^{(0)} + \frac{1}{\tau\delta_t} f_\alpha^{(1)} = 0, \tag{2.11b}$$

$$\mathcal{O}(\varepsilon^2): \qquad \frac{\partial f_\alpha^{(0)}}{\partial t_1} + \Big(1 - \frac{1}{2\tau}\Big)\Big(\frac{\partial}{\partial t_0} + \mathbf{e}_\alpha \cdot \nabla_1\Big) f_\alpha^{(1)} + \frac{1}{\tau\delta_t} f_\alpha^{(2)} = 0. \tag{2.11c}$$

From Eq. (2.11a), we have

$${f_\alpha}^{(0)} = {f_\alpha}^{eq}, \tag{2.12}$$

and from Eq. (2.11b), we get

$$\varepsilon f_\alpha^{(1)} = -\tau\delta_t \varepsilon \Big(\frac{\partial}{\partial t_0} + \mathbf{e}_\alpha \cdot \nabla_1\Big) f_\alpha^{eq}. \tag{2.13}$$

By taking summation of Eqs. (2.11b) and (2.11c) over α and combining the resultant formulations on the t_0 and t_1 time scales, the following equation is derived

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \Big(\sum_{\alpha=0}^{N} \mathbf{e}_\alpha f_\alpha^{eq}\Big) = 0. \tag{2.14}$$

Furthermore, by taking the first-moment summation of Eqs. (2.11b) and (2.11c) over α and combining the resultant formulations on the t_0 and t_1 time scales, the following equation is obtained:

$$\frac{\partial \rho \mathbf{u}}{\partial t}+\nabla\cdot\Pi=0, \tag{2.15}$$

where Π is the momentum flux tensor defined by

$$\Pi_{\beta\gamma}=\sum_{\alpha=0}^{N}(\mathbf{e}_\alpha)_\beta(\mathbf{e}_\alpha)_\gamma\Big[f_\alpha^{eq}+\Big(1-\frac{1}{2\tau}\Big)\varepsilon f_\alpha^{(1)}\Big]. \tag{2.16}$$

Here, $(\mathbf{e}_\alpha)_\beta$ is the component of the lattice velocity vector $\mathbf{e}_\alpha$ in the β-coordinate direction. From the expressions of Eqs. (2.14)-(2.16), it can be seen clearly that to the zero order of ε, that is, f_α is approximated by f_α^{eq} as shown by Eq. (2.9a), Eqs. (2.14) and (2.15) recover Euler equations. And to the first order of ε, in which f_α is approximated by $f_\alpha^{eq}+\varepsilon f_\alpha^{(1)}$ as shown by Eq. (2.9a), Eqs. (2.14) and (2.15) recover N-S equations (2.1a)-(2.1b) with the relationship between τ and viscosity ν given by Eq. (2.7). In this case, Eqs. (2.13) and (2.16) can be approximated by

$$f_\alpha^{neq}=f_\alpha-f_\alpha^{eq}=\varepsilon f_\alpha^{(1)}=-\tau\delta_t\Big(\frac{\partial}{\partial t}+\mathbf{e}_\alpha\cdot\nabla\Big)f_\alpha^{eq}, \tag{2.17a}$$

$$\Pi_{\beta\gamma}=\sum_{\alpha=0}^{N}(\mathbf{e}_\alpha)_\beta(\mathbf{e}_\alpha)_\gamma\Big[f_\alpha^{eq}+\Big(1-\frac{1}{2\tau}\Big)f_\alpha^{neq}\Big]. \tag{2.17b}$$

Here f_α^{neq} is the non-equilibrium distribution function. Note that Eq. (2.17a) is also applied in the gas kinetic scheme [37].

3 Lattice Boltzmann Flux Solver (LBFS)

3.1 Finite volume discretization of macroscopic differential equations

The multi-scale Chapman-Enskog expansion analysis provides a solid foundation for LBM. It guarantees that the macroscopic flow variables obtained by LBM would satisfy N-S equations. As discussed in the introduction, LBE is usually applied globally in the whole computational domain and for all time levels. As such, it causes some limitations for application. On the other hand, if we start from Eqs. (2.14) and (2.15) given from Chapman-Enskog analysis, we can combine the solution algorithms of N-S solvers with LBE solvers. To be specific, we can apply the finite volume method to discretize Eqs. (2.14) and (2.15) so that the conservative variables ρ and $\rho\mathbf{u}$ defined at the cell center can be obtained by marching in time. Unlike N-S solvers where the inviscid and viscous fluxes are approximated differently and the derivative approximation is needed, the inviscid and viscous fluxes at the cell interface in the present lattice Boltzmann flux solver (LBFS) are evaluated simultaneously by local reconstruction of LBE solution from macroscopic flow variables at cell centers. For simplicity, we will consider the 2D case to illustrate the details of LBFS. For the 2D case, the flow velocity **u** has components u and v and in the x and y direction respectively. Similarly, the particle velocity $\mathbf{e}_\alpha$ also has two components $\mathbf{e}_{\alpha x}$ and $\mathbf{e}_{\alpha y}$ in the x and y directions. With these notations, Eqs. (2.14) and (2.15) can be written as:

$$\frac{\partial \rho}{\partial t}+\frac{\partial P_x}{\partial x}+\frac{\partial P_y}{\partial y}=0, \tag{3.1a}$$

$$\frac{\partial \rho u}{\partial t}+\frac{\partial \Pi_{xx}}{\partial x}+\frac{\partial \Pi_{xy}}{\partial y}=0, \tag{3.1b}$$

$$\frac{\partial \rho v}{\partial t}+\frac{\partial \Pi_{yx}}{\partial x}+\frac{\partial \Pi_{yy}}{\partial y}=0, \tag{3.1c}$$

where

$$P_x=\sum_{\alpha=0}^{N}\mathbf{e}_{\alpha x}f_\alpha^{eq}, \tag{3.2a}$$

$$P_y=\sum_{\alpha=0}^{N}\mathbf{e}_{\alpha y}f_\alpha^{eq}, \tag{3.2b}$$

$$\Pi_{xx}=\sum_{\alpha=0}^{N}\mathbf{e}_{\alpha x}\mathbf{e}_{\alpha x}\left[f_\alpha^{eq}+\left(1-\frac{1}{2\tau}\right)f_\alpha^{neq}\right], \tag{3.2c}$$

$$\Pi_{xy}=\sum_{\alpha=0}^{N}\mathbf{e}_{\alpha x}\mathbf{e}_{\alpha y}\left[f_\alpha^{eq}+\left(1-\frac{1}{2\tau}\right)f_\alpha^{neq}\right], \tag{3.2d}$$

$$\Pi_{yx}=\sum_{\alpha=0}^{N}\mathbf{e}_{\alpha y}\mathbf{e}_{\alpha x}\left[f_\alpha^{eq}+\left(1-\frac{1}{2\tau}\right)f_\alpha^{neq}\right], \tag{3.2e}$$

$$\Pi_{yy}=\sum_{\alpha=0}^{N}\mathbf{e}_{\alpha y}\mathbf{e}_{\alpha y}\left[f_\alpha^{eq}+\left(1-\frac{1}{2\tau}\right)f_\alpha^{neq}\right]. \tag{3.2f}$$

In the above equations, τ is computed by Eq. (2.7), and the lattice velocity is given by the chosen lattice velocity model. If we define the vectors **W**, **E** and **F** as

$$\mathbf{W}=\begin{Bmatrix}\rho\\ \rho u\\ \rho v\end{Bmatrix},\quad \mathbf{E}=\begin{Bmatrix}P_x\\ \Pi_{xx}\\ \Pi_{yx}\end{Bmatrix},\quad \mathbf{F}=\begin{Bmatrix}P_y\\ \Pi_{xy}\\ \Pi_{yy}\end{Bmatrix}, \tag{3.3}$$

then Eqs. (3.1a)-(3.1c) can be put into the following form,

$$\frac{\partial \mathbf{W}}{\partial t}+\frac{\partial \mathbf{E}}{\partial x}+\frac{\partial \mathbf{F}}{\partial y}=0. \tag{3.4}$$

Integrating Eq. (3.4) over a control cell Ω_i gives

$$\frac{d\mathbf{W}_i}{dt}=-\frac{1}{dV_i}\sum_k (n_x\mathbf{E}+n_y\mathbf{F})_k dS_k, \tag{3.5}$$

where dV_i is the volume of the control cell, and dS_k is the area of the kth control surface enclosed Ω_i, n_x and n_y are the x and y components of the unit normal vector on the kth control surface. Obviously, once the fluxes at all cell interfaces are known, Eq. (3.5) can be solved by well established numerical schemes such as the 4-satge Runge-Kutta method. Thus, the evaluation of fluxes **E** and **F** at the cell interface is the key in the solution process.

3.2 Flux evaluation at cell interface by LBFS

Consider a cell interface between control cells Ω_i and Ω_{i+1} as shown in Fig. 1. We need to evaluate fluxes **E** and **F** at the interface from flow variables at 2 cell centers. As can be seen from Eqs. (3.2a)-(3.3), to evaluate fluxes **E** and **F**, we have to know the equilibrium distribution functions f_α^{eq} and non-equilibrium distribution functions f_α^{neq} at the cell interface. In the following, we will discuss how to obtain f_α^{eq} and f_α^{neq} respectively.

It is indicated in Section 2 that, to the Navier-Stokes level, f_α^{neq} can be approximated by Eq. (2.17a). Now, we assume that the physical location for the two cell centers and their interface is respectively $\mathbf{r}_i$, $\mathbf{r}_{i+1}$ and $\mathbf{r}$. Using Taylor series expansion, we have

$$f_\alpha^{eq}(\mathbf{r},t)-f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)=\delta_t\Big(\frac{\partial}{\partial t}+\mathbf{e}_\alpha\cdot\nabla\Big)f_\alpha^{eq}+\mathcal{O}(\delta_t^2). \tag{3.6}$$

From Eqs. (3.6) and (2.17a), we can get the following form

$$f_\alpha^{neq}(\mathbf{r},t)=f_\alpha^{neq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)=-\tau\Big[f_\alpha^{eq}(\mathbf{r},t)-f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)\Big]+\mathcal{O}(\delta_t^2). \tag{3.7}$$

Eq. (3.7) shows that once we have the equilibrium distribution functions at the cell interface and its surrounding points, we can have the full information of distribution function

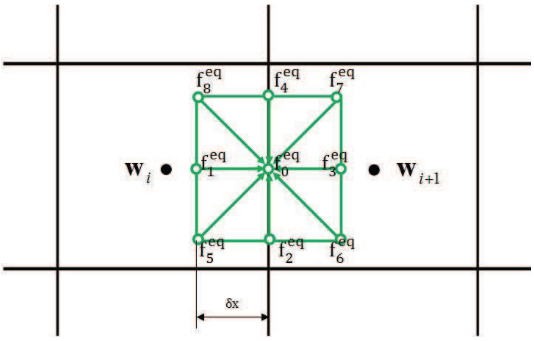


Figure 1: Local reconstruction of LBE solution at cell interface, $\mathbf{W}=(\rho,\rho u,\rho v)$.

at the interface. Note that the approximation for Eq. (3.7) is the second order of accuracy in δ_t. Using Eq. (2.6), the equilibrium distribution function f_α^{eq} can be computed from the fluid density ρ and flow velocity $\mathbf{u}$. With the given density and velocity at the cell center, the respective density and velocity at location $(\mathbf{r}-\mathbf{e}_\alpha\delta_t)$ can be easily obtained by interpolation. One of interpolation forms can be written as

$$\rho(\mathbf{r}-\mathbf{e}_\alpha\delta_t)=\begin{cases}\rho(\mathbf{r}_i)+(\mathbf{r}-\mathbf{e}_\alpha\delta_t-\mathbf{r}_i)\cdot\nabla\rho(\mathbf{r}_i), & \text{when } \mathbf{r}-\mathbf{e}_\alpha\delta_t \text{ is in the cell } \Omega_i,\\ \rho(\mathbf{r}_{i+1})+(\mathbf{r}-\mathbf{e}_\alpha\delta_t-\mathbf{r}_{i+1})\cdot\nabla\rho(\mathbf{r}_{i+1}), & \text{when } \mathbf{r}-\mathbf{e}_\alpha\delta_t \text{ is in the cell } \Omega_{i+1},\end{cases} \tag{3.8a}$$

$$\mathbf{u}(\mathbf{r}-\mathbf{e}_\alpha\delta_t)=\begin{cases}\mathbf{u}(\mathbf{r}_i)+(\mathbf{r}-\mathbf{e}_\alpha\delta_t-\mathbf{r}_i)\cdot\nabla\mathbf{u}(\mathbf{r}_i), & \text{when } \mathbf{r}-\mathbf{e}_\alpha\delta_t \text{ is in the cell } \Omega_i,\\ \mathbf{u}(\mathbf{r}_{i+1})+(\mathbf{r}-\mathbf{e}_\alpha\delta_t-\mathbf{r}_{i+1})\cdot\nabla\mathbf{u}(\mathbf{r}_{i+1}), & \text{when } \mathbf{r}-\mathbf{e}_\alpha\delta_t \text{ is in the cell } \Omega_{i+1}.\end{cases} \tag{3.8b}$$

With computed $\rho(\mathbf{r}-\mathbf{e}_\alpha\delta_t)$ and $\mathbf{u}(\mathbf{r}-\mathbf{e}_\alpha\delta_t)$ by Eqs. (3.8a) and (3.8b), $f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)$ can be given by Eq. (2.6). Now, we are only left to determine $f_\alpha^{eq}(\mathbf{r},t)$ as shown in Eq. (3.7). Again, with Eq. (2.6), the determination of $f_\alpha^{eq}(\mathbf{r},t)$ is equivalent to computing $\rho(\mathbf{r},t)$ and $\mathbf{u}(\mathbf{r},t)$. Using Eq. (2.3), the conservative variables ρ and $\rho\mathbf{u}$ can be computed by

$$\rho(\mathbf{r},t)=\sum_{\alpha=0}^{N} f_\alpha(\mathbf{r},t), \tag{3.9a}$$

$$\rho(\mathbf{r},t)\mathbf{u}(\mathbf{r},t)=\sum_{\alpha=0}^{N} f_\alpha(\mathbf{r},t)\mathbf{e}_\alpha, \tag{3.9b}$$

where $f_\alpha(\mathbf{r},t)$ is given by the lattice Boltzmann equation (2.2). As shown in Fig. 1, we can locally apply Eq. (2.2) at the cell interface, and have

$$f_\alpha(\mathbf{r},t)=f_\alpha(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)-\frac{f_\alpha(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)-f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)}{\tau}, \quad \alpha=0,1,\cdots,N. \tag{3.10}$$

Here $f_\alpha(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)$ is the initial distribution function in the reconstruction process of local LBE solver. In general, $f_\alpha(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)$ consists of two parts: equilibrium part and non-equilibrium part. That is, $f_\alpha(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)$ can be written as

$$f_\alpha(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)=f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)+f_\alpha^{neq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t). \tag{3.11}$$

Substituting Eq. (3.11) into Eq. (3.10) gives

$$f_\alpha(\mathbf{r},t)=f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)+\Big(1-\frac{1}{\tau}\Big)f_\alpha^{neq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t). \tag{3.12}$$

Furthermore, by substituting Eq. (3.7) into Eq. (3.12), we obtain

$$f_\alpha(\mathbf{r},t)=(1-\tau)f_\alpha^{eq}(\mathbf{r},t)+\tau f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t). \tag{3.13}$$

Eq. (3.13) is actually equivalent to $f_\alpha(\mathbf{r},t)=f_\alpha^{eq}(\mathbf{r},t)+f_\alpha^{neq}(\mathbf{r},t)$. Finally, application of compatibility conditions (2.3) and (2.8) gives

$$\rho(\mathbf{r},t)=\sum_{\alpha=0}^{N} f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t), \tag{3.14a}$$

$$\rho(\mathbf{r},t)\mathbf{u}(\mathbf{r},t)=\sum_{\alpha=0}^{N} f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)\mathbf{e}_\alpha, \tag{3.14b}$$

Eqs. (3.14a) and (3.14b) show a very interesting result. That is, the conservative flow variables at the cell interface are fully determined from the equilibrium distribution functions of particles at the surrounding points, which stream to the cell interface within a short streaming time step δ_t. As equilibrium distribution functions only depend on the macroscopic flow variables, there is no need to store the density distribution functions for all the time levels. In fact, at any time step, we locally construct a LBE solution at any cell interface in order to evaluate fluxes there. The reconstruction process is applied locally and repeated from one time level to another time level. Overall, the basic solution procedure of LBFS can be summarized below:

1. At first, we have to choose a lattice velocity model such as D2Q9 model. Then we need to specify a streaming time step δ_t. The choice of δ_t should satisfy the constraint that the location of $(\mathbf{r}-\mathbf{e}_\alpha\delta_t)$ must be within either the cell Ω_i or the cell Ω_{i+1}. Note that as local LBE solution is reconstructed at each cell interface, different interfaces could use different δ_t. This provides a great flexibility for application if we use non-uniform mesh or solve problems with curved boundary. Once δ_t is chosen, the single relaxation parameter τ in LBFS is calculated by Eq. (2.7).
2. For the considered interface position $\mathbf{r}$, identify its surrounding positions $(\mathbf{r}-\mathbf{e}_\alpha\delta_t)$, and then use Eqs. (3.8a) and (3.8b) to compute the macroscopic flow variables at those positions.
3. Use Eq. (2.6) to calculate the equilibrium density distribution function $f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta_t,t-\delta_t)$.
4. Compute the macroscopic flow variables at the cell interface by using Eqs. (3.14a) and (3.14b), and further calculate $f_\alpha^{eq}(\mathbf{r},t)$ by Eq. (2.6).
5. Calculate $f_\alpha^{neq}(\mathbf{r},t)$ by using Eq. (3.7).
6. Compute the fluxes at the cell interface by Eqs. (3.2a)-(3.3).
7. Once fluxes at all cell interfaces are obtained, solve ordinary differential equations (3.5) by using 4-stage Runge-Kutta scheme.

It is noted that the present LBFS can not only be used to simulate incompressible viscous flows, but also be applied to simulate incompressible inviscid flows. For the inviscid flow, we just simply set $\tau=0.5$. Another point to note is that the time marching step used in solving Eq. (3.5) and the streaming time step δ_t used in LBFS are independent. δ_t can be selected differently at different location and different time level. As shown in the following section, its effect on the solution accuracy is very little.

4 Numerical results and discussion

In this section, the developed LBFS is validated by applying it to simulate some test problems. At first, the decaying vortex problem is solved on the uniform mesh to study the order of solution accuracy. Then the driven cavity flow is simulated. For the case of $Re=100$, numerical simulation on the uniform mesh is carried out to show that the choice of streaming time step δ_t in the reconstruction of local LBE solution has no effect on the solution accuracy. After that, numerical simulation for other Reynolds numbers on the non-uniform mesh is presented to show flexibility and capability of LBFS on the non-uniform grid. The comparison of computational time required by LBFS and a conventional LBM is also shown. The third test problem is the flow past a circular cylinder. On one hand, we will use this example to test the ability of LBFS for problems with curved boundary. On the other hand, through this example, we will demonstrate that LBFS has capability to accurately simulate both viscous and inviscid flows. Note that for all test cases, D2Q9 lattice velocity model is adopted. For the steady flow, the following convergence criterion is applied,

$$\frac{\sum_{ij}\left|\left(\sqrt{u^2+v^2}\right)^{n+1}-\left(\sqrt{u^2+v^2}\right)^{n}\right|}{\sum_{ij}\left(\sqrt{u^2+v^2}\right)^{n+1}}\le 10^{-6}. \tag{4.1}$$

4.1 Simulation of decaying vortex flow

The numerical accuracy of LBFS is examined by simulating the decaying vortex flow, which has an analytic solution given by

$$u(x,y,t)=-U\cos(\pi x/L)\sin(\pi y/L)e^{-2\pi^2Ut/(ReL)}, \tag{4.2a}$$

$$v(x,y,t)=-U\sin(\pi x/L)\cos(\pi y/L)e^{-2\pi^2Ut/(ReL)}, \tag{4.2b}$$

$$\rho(x,y,t)=\rho_0-\frac{\rho_0U^2}{4c_s^2}[\cos(\pi x/L)+\sin(\pi y/L)]e^{-4\pi^2Ut/(ReL)}. \tag{4.2c}$$

In the present test, the computational domain of $[-L,L]\times[-L,L]$ is chosen where six different uniform grids ($N\times N$, $N=21,41,61,81,101$ and 161) are used. The Reynolds number is selected as $Re=UL/\nu=10$ and the relaxation parameter τ is set as $\tau=0.65$, ρ_0 is taken as 1. The solution at $t=L/U=1$ is computed and the relative error of velocity component u is measured by L_2 norm which is defined as

$$L_2(u)_{retative}=\sqrt{\frac{1}{N\times N}\sum_{k=1}^{N\times N}\left(\frac{u^{numerical}-u^{exact}}{u^{exact}}\right)^2}, \tag{4.3}$$

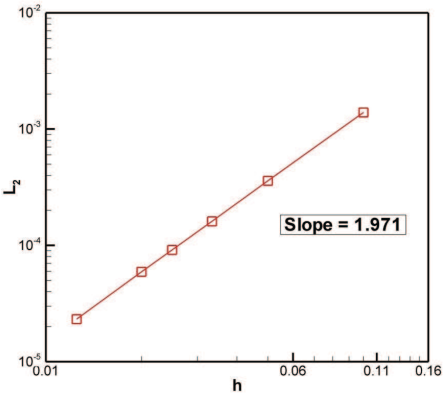


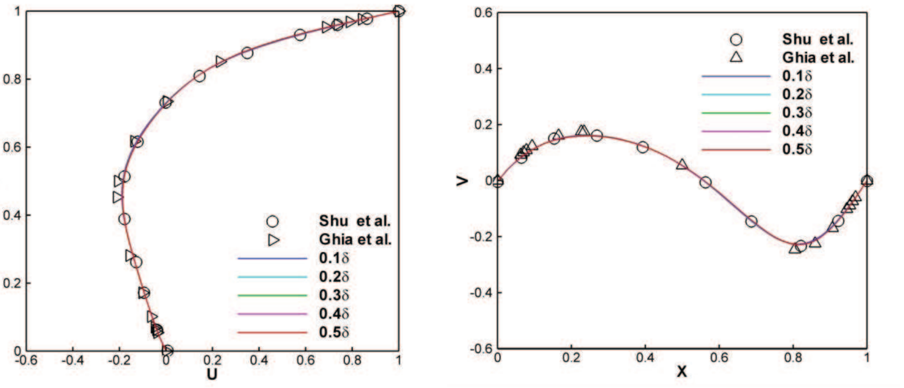
Figure 2: L_2 norm of relative error of u versus h for the decaying vortex flow.

where $u^{numerical}$ and u^{exact} represent the numerical result and the exact solution, respectively. Fig. 2 shows the L_2 norm of the relative error of u versus the mesh spacing in the log scale. As can be seen, the slope of the line is 1.971 which is close to 2. This indicates that the accuracy of the present LBFS solution is roughly the second order in space.

4.2 Simulation of lid-driven flow in a square cavity

The lid-driven flow in a square cavity is a standard test case for validating new numerical methods in simulation of incompressible viscous lows. At first, we use this example to simulate the flow at $Re=100$ on the uniform grid of 49×49 with 5 different streaming steps δ_t. The Reynolds number for the problem is defined as $Re=UL/\nu$, where U is the velocity of the top lid and L is the length of the square cavity.

In the present application of D2Q9 lattice velocity model, c is taken as 1. Thus the streaming distance δ_x and the streaming step δ_t in the LBFS have the same value, that is, $\delta_t=\delta_x$. Suppose that the mesh spacing for the uniform grid is noted as $\delta=1/48$. 5 streaming distances in the LBFS, that is, $\delta_x=0.1\delta,0.2\delta,0.3\delta,0.4\delta,0.5\delta$, are selected to study the effect of δ_t. Numerical simulation shows that all 5 different streaming distances give the same results. This can be seen clearly in Fig. 3, which depicts the velocity component u-profile along the vertical centerline and the velocity component v-profile alone the horizontal centerline with 5 different streaming distances. The results are also in good agreement with available data in the literature [19, 41]. This is a very interesting result. It clearly shows that numerical results of LBFS are independent on the choice of streaming distance. It implies that the streaming distance for any interface between 2 control cells is selective and could be different for different interfaces. This feature ensures that the present solver could be applied easily on non-uniform and body-fitted grids which removes the limitation of the standard LBM and makes the solver be more flexible. In the following, we will study the performance of LBFS for the application on non-uniform grids.



(a) (b)

Figure 3: u (a) and v (b) velocity along vertical and horizontal centerlines at Re=100 using 5 different streaming distances.

The non-uniform grid for the simulation of driven cavity flow can be generated by using the following formulation,

$$x_i=\frac{1}{2}\Big[1-\cos\Big(\frac{i-1}{N-1}\pi\Big)\Big], \qquad i=1,2,\cdots,N, \tag{4.4a}$$

$$y_j=\frac{1}{2}\Big[1-\cos\Big(\frac{j-1}{M-1}\pi\Big)\Big], \qquad j=1,2,\cdots,M, \tag{4.4b}$$

where N and M are respectively the total number of mesh points in the x and y direction. With Eq. (4.4), the non-uniform grids of 61×61 for Re=400, 81×81 for Re=1000, 121×121 for $Re=5000$ and 10000 are used respectively. For these non-uniform grids, the mesh spacing near the wall is taken very small to capture the thin boundary layer, and in the middle region, mesh spacing is relatively large. This distribution on one hand can well capture the physics, and on the other hand, can reduce the computational effort, especially for high Reynolds number cases. We have to indicate that when the LBM is applied globally to solve this problem with non-uniform grids, the streaming distance in the LBE solver is restricted by the minimum mesh spacing in the whole domain, which greatly degrades the computational efficiency. In contrast, in the LBFS, LBM is applied locally, and the local streaming distance is pegged to the local mesh spacing. In other words, in the fine mesh region, we can use a small streaming distance, while in the coarse mesh region, we can take a larger streaming distance. As a result, less time steps and computational effort could be needed. Table 1 compares the converged iteration numbers and run time (s) on Lenovo Laptop (2.53GHz and 4G RAM) required respectively by LBFS and Taylor series expansion- and least square-based lattice Boltzmann method (TLLBM) [19]. The case is for $Re=1000$, and 3 non-uniform grids of 81×81, 101×101, 121×121 are used. It can be seen clearly from Table 1 that to satisfy the convergence criterion (4.1), LBFS needs less iteration numbers than TLLBM. In addition, the run time per iteration number by LBFS is also less than that by TLLBM. This test case well demonstrates the computational

Table 1: Comparison of computational time by LBFS and TLLBM for lid-driven cavity flow at $Re=1000$.

Grid Size	Iteration numbers		Run time (s) on Lenovo Laptop (2.53GHz)		
	LBFS	TLLBM	LBFS	TLLBM	LBFS/TLLBM
81×81	42924	66722	130.75	471.34	27.74%
101×101	51884	76821	252.44	764.36	33.03%
121×121	60920	87356	441.69	1145.66	38.55%

Table 2: Locations of primary vortex centers at different Reynolds numbers.

Re	Vortex center	
	Ghia et al. [41]	LBFS
400	(0.5547,0.6055)	(0.5571,0.6047)
1000	(0.5313,0.5625)	(0.5320,0.5662)
5000	(0.5117,0.5352)	(0.5162,0.5364)
10000	(0.5117,0.5333)	(0.5136,0.5323)

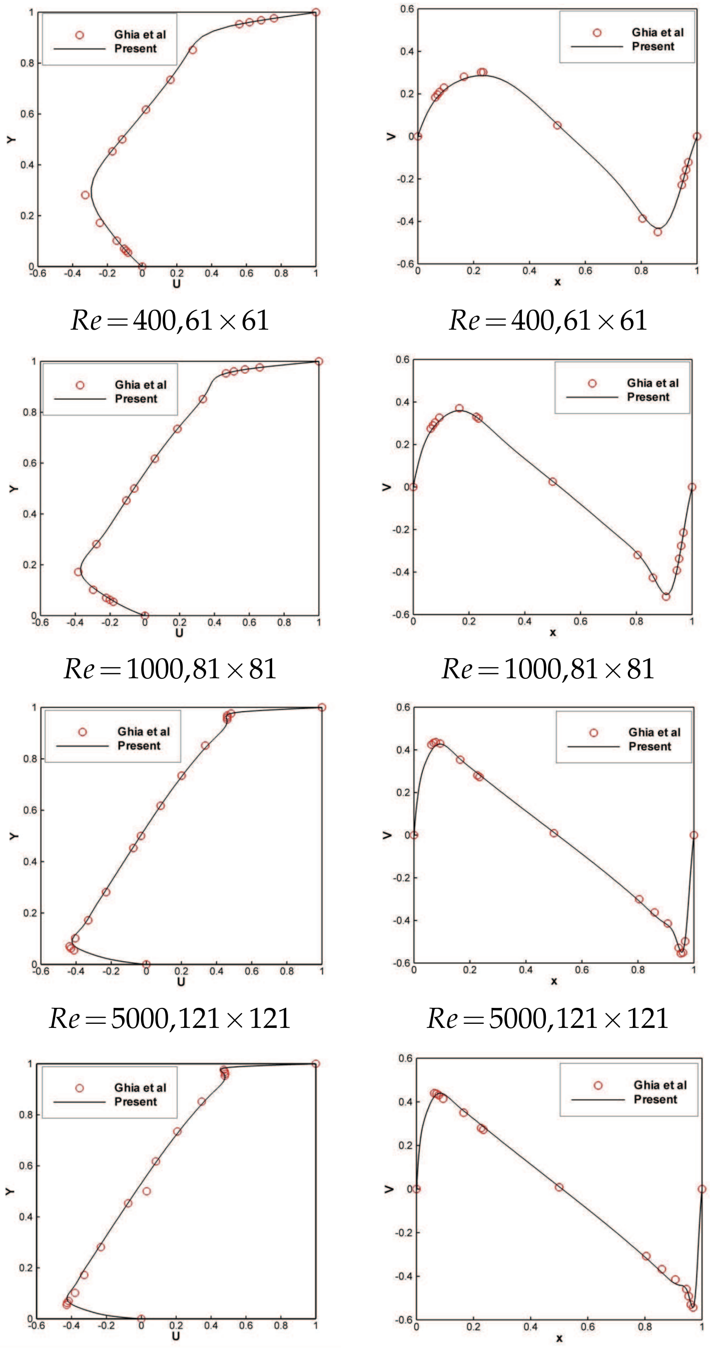
efficiency of LBFS.

Table 2 compares the locations of the primary vortex centers at different Reynolds numbers obtained by LBFS with those given by Ghia et al. [41]. As shown in the table, the vortex center moves toward the cavity center as Re increases and the maximum relative error between present solutions and those of Ghia et al. [41] is less than 0.9%. Fig. 4 displays u-velocity profile along the horizontal centerline and v-velocity profile along the vertical centerline at $Re=400$, 1000, 5000 and 10000. As can be seen from this figure, the present results agree very well with those of Ghia et al. [41]. Fig. 5 shows the streamlines for the four Reynolds numbers. The most striking aspect of this figure is that the Reynolds number apparently has unique effect on flow patterns. Secondary and tertiary vortices appear and evolve into larger ones as Re becomes large. These results and observations are in good agreement with those of Ghia et al. [41].

4.3 Simulation of viscous flow past a circular cylinder

Although the complex lid-driven cavity flows have been successfully simulated to validate the present solver, the geometry of the cavity which only involves straight boundaries is nevertheless simple. To further illustrate the capability of LBFS for problems with curved boundary, the flow past a circular cylinder is simulated. This problem is very attractive and has been investigated extensively. There are an increasing number of numerical and experimental results available in the literature.

The flow behaviors for this problem are characterized by the Reynolds number which is defined as $Re=U_\infty D/\nu$, where U_∞ is the free stream velocity; D is the diameter of the cylinder and ν is the kinematic viscosity of the fluid. The drag and lift coefficients are useful parameters and commonly used to check the accuracy of numerical results. In order to compute these two coefficients, the drag and lift forces should be computed first



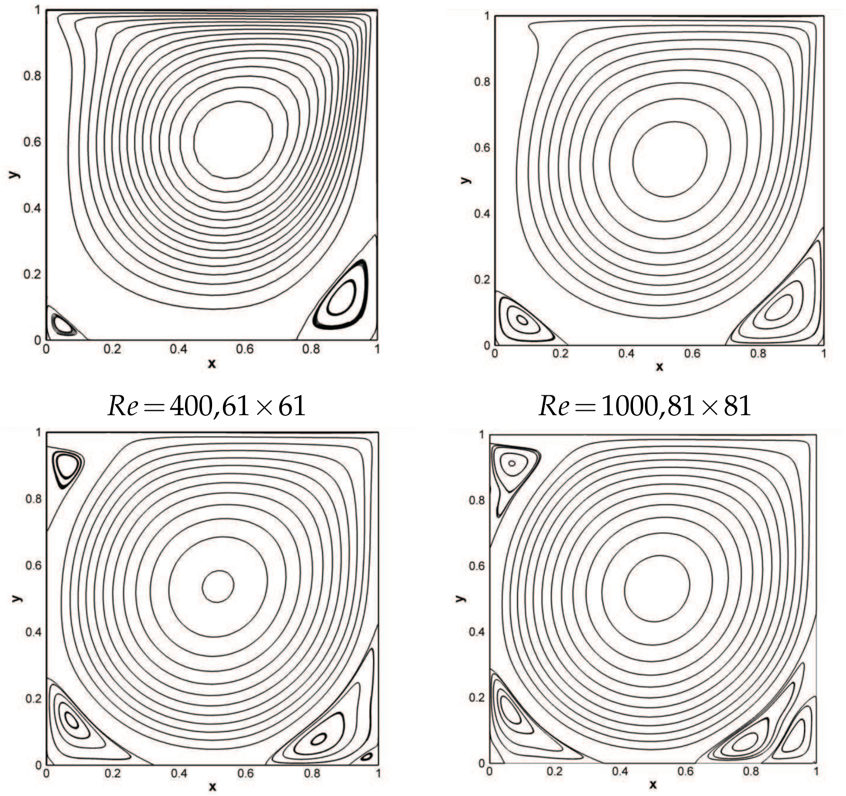
$Re=400, 61\times 61$ $Re=400, 61\times 61$

$Re=1000, 81\times 81$ $Re=1000, 81\times 81$

$Re=5000, 121\times 121$ $Re=5000, 121\times 121$

$Re=10000, 121\times 121$ $Re=10000, 121\times 121$

Figure 4: u and v velocity profiles along horizontal and vertical centerlines for a lid-driven cavity flow at various Reynolds numbers.



$Re = 400, 61\times 61$ $Re = 1000, 81\times 81$

$Re = 5000, 121\times 121$ $Re = 10000, 121\times 121$

Figure 5: Streamlines for a lid-driven cavity flow at various Reynolds numbers.

by integrating the momentum equations over the surface of the circular cylinder,

$$F_d = -\oint \Big[\Big(p-\mu\frac{\partial u}{\partial x}\Big)n_x - \mu\frac{\partial u}{\partial y}n_y\Big]ds, \tag{4.5a}$$

$$F_l = -\oint \Big[\mu\frac{\partial v}{\partial x}n_x - \Big(p-\mu\frac{\partial v}{\partial y}\Big)n_y\Big]ds, \tag{4.5b}$$

where F_d and F_l are the drag force and lift force respectively; $n = (n_x, n_y)$, is the outer normal vector on the cylinder surface; μ is the dynamic viscosity. Then the drag and lift coefficients can be calculated by

$$C_d = \frac{F_d}{\frac{1}{2}\rho_\infty U_\infty^2}, \tag{4.6a}$$

$$C_l = \frac{F_l}{\frac{1}{2}\rho_\infty U_\infty^2}. \tag{4.6b}$$

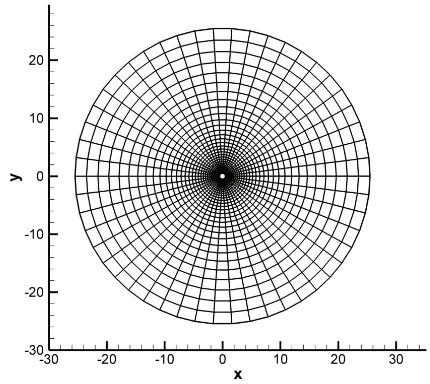


Figure 6: Typical O-type grid.

The pressure coefficient is defined as

$$c_p = \frac{p_w - p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2}, \tag{4.7}$$

where ρ_∞ and p_∞ are the free stream density and pressure respectively, p_w is the pressure on the cylinder surface.

Another important parameter for the unsteady flow is the Strouhal number which examines the vortex shedding frequency from the cylinder. As a non-dimensional parameter, it can be defined as

$$St = \frac{f_q D}{U_\infty}. \tag{4.8}$$

Here, f_q is the vortex shedding frequency.

In the present study, the typical O-type structured grid is adopted which is shown in Fig. 6. For simulation of steady and unsteady flows, the computational grids are set, respectively, as 301×201 and 301×501. The far-field boundaries are put respectively at 25.5 diameters and 55.5 diameters away from the center of the cylinder. In order to well capture the boundary layer and vortex structures near the solid boundary and in the meantime to reduce the computational cost, the grid is fine around the cylinder and coarse near the far field. The free stream density ρ_∞ is set as 1.0 and the free stream velocity U_∞ is taken as 0.1. Initially, the flow properties are set the same as the free stream values.

Numerical simulation shows that when Re=20 and 40, the flow reaches a steady state. The streamlines of these two cases are displayed in Fig. 7. As shown in this figure, the streamlines are symmetric about the x axis, and a pair of stationary recirculation eddies are attached behind the cylinder. The scales and strength of these eddies are enlarged as the Reynolds number increases. To show geometrical quantities of the eddies, the recirculation length l_s, which is set as the distance between the rearmost point of the cylinder

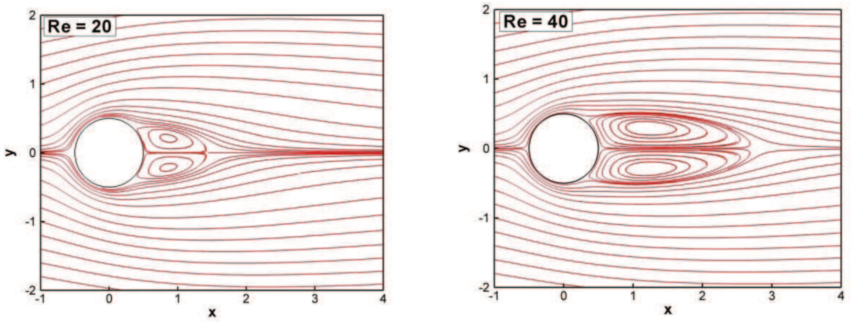


Figure 7: Streamlines for the steady flow past a circular cylinder at $Re=20,40$.

and the end of the wake, and separation angle, θ_s, are measured. The drag coefficient C_d is also an essential parameter for the steady flow. Table 3 gives the detailed comparison of these three parameters between present solutions and those of previous studies (Dennis and Chang [42], Nieuwstadt and Keller, [43], He and Doolen, [44], Shukla et al. [45]). Obviously, for both Reynolds numbers of 20 and 40, the present results agree well with the solutions in the literature. To further demonstrate the performance of present solver, the pressure coefficient distribution around the cylinder surface at $Re=40$ is depicted in Fig. 8. The numerical solutions obtained by interpolation-based LBM on curvilinear coordinates of He and Doolen [44] and the experimental data of Park et al. [46] are also included in this figure. The orientation angle θ is measured in degree from the leading stagnation point to the trailing stagnation point. Due to the fact that the flow flied is symmetric about the x-axis, only the pressure coefficient distribution on the upper surface of the cylinder is shown. As can be seen in the figure, good agreement between the present results and those of previous studies is achieved.

For the case of Re=100 and 200, the flow is unsteady and eventually reaches a periodic state. The temporal evolution of the lift and drag coefficients of $Re=100$, 200 is shown in Fig. 9. As can be seen from this figure, both the lift and drag coefficients show periodic

Table 3: Comparison of drag coefficient, recirculation length and separation angle for steady flow past a circular cylinder at $Re=20,40$.

Re	References	C_d	l_s/D	θ_s
20	Dennis and Chang [42]	2.05	0.94	43.7
	Nieuwsdadt and Keller [43]	2.053	0.893	-
	He and Doolen [44]	2.152	0.921	42.96
	Shukla et al. [45]	2.07	0.92	43.3
	Present	2.062	0.935	42.94
40	Dennis and Chang [42]	1.52	2.35	53.8
	Nieuwsdadt and Keller [43]	1.54	2.18	-
	He and Doolen [44]	1.499	2.245	52.84
	Shukla et al. [45]	1.55	2.34	52.7
	Present	1.53	2.240	52.69

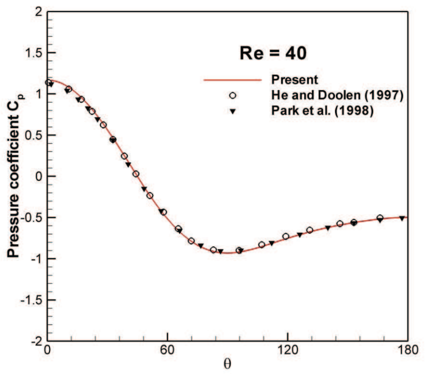
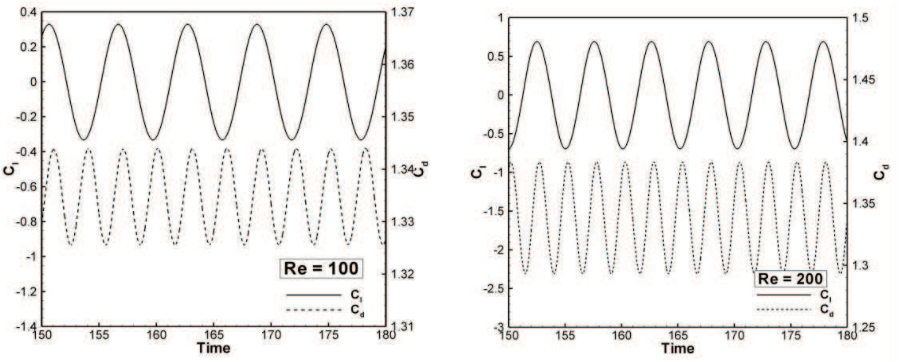
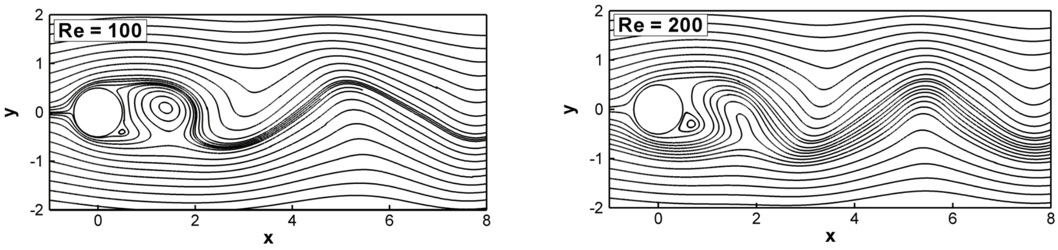


Figure 8: Comparison of pressure coefficient distribution on cylinder surface at $Re=40$.



(a) (b)

Figure 9: Evolution of lift and drag coefficients for the flow past a cylinder at $Re=100$ and 200.



(a) (b)

Figure 10: Instantaneous streamlines for unsteady flow past a circular cylinder at $Re=100$ and 200.

feature and the period of the lift coefficient is twice of that of drag coefficient for the two Reynolds numbers. The periodic distribution clearly shows that the flow field reaches the periodic state. Table 4 shows the quantitative comparison of the lift and drag coefficients as well as the Strouhal number. It can be seen from this table that, the present results are compared well with those of Braza et al. [47], Benson et al. [48] and Ding et al. [49].

Table 4: Comparison of dynamic parameters for unsteady flow past a circular cylinder at $Re=100,200$.

Re	References	C_l	C_d	St
100	Braza et al. [47]	±0.30	1.28 ±0.02	0.16
	Benson et al. [48]	±0.38	1.46 ±0.01	0.17
	Ding et al. [49]	±0.28	1.325 ±0.008	0.164
	Present	±0.33	1.334 ±0.009	0.164
200	Braza et al. [47]	±0.78	1.38 ±0.07	0.19
	Benson et al. [48]	±0.65	1.45 ±0.04	0.193
	Ding et al. [49]	±0.60	1.327 ±0.045	0.196
	Present	±0.69	1.338 ±0.045	0.197

Another distinctive feature of the results in Table 4 is that, the effect of Reynolds number from 100 to 200 on the mean values of both lift and drag coefficients is not obvious, but the effect on amplitude of these coefficients is substantial. This feature could also be seen from Fig. 9. Fig. 10 depicts the instantaneous flow flied. As shown in this figure, the well-known Karman vortex street can be clearly seen from the streamlines.

4.4 Simulation of inviscid flow past a circular cylinder

It is well known that the standard LBM is only applicable for the viscous flow due to stability condition, that is, τ cannot be taken as 0.5. As shown in Eqs. (3.2a)-(3.2f), in the present LBFS, when τ is set as 0.5, the contribution from the non-equilibrium distribution function is vanished, and only the equilibrium distribution function has contribution to the flux calculation. In fact, the present LBFS can be well applied to both viscous and inviscid flows. The performance of LBFS for simulation of viscous flows has been tested for the decaying vortex flow, driven cavity flow and the flow past a circular cylinder. In the following, we will further test LBFS for simulation of inviscid flows by setting $\tau=0.5$.

The test problem we consider is the inviscid flow past a circular cylinder. Since there is no boundary layer around the cylinder surface for this case, the computational grid used is much coarser than that for the viscous flow. In the present simulation, the grid size of 121×51 is used. The far-field boundary is taken 20.5 diameters away from the geometrical center of the cylinder. The no-penetration condition is applied at the cylinder surface.

Fig. 11 shows the pressure coefficient distribution around the cylinder surface. Also included in the figure is the theoretical result. The horizontal axis is the orientation angle θ which is measured from the trailing stagnation point along the counterclockwise direction. As shown in Fig. 11, good agreement is achieved between the present result and theoretical solution. The streamlines past the circular cylinder are depicted in Fig. 12, which clearly shows that the streamlines pass through the cylinder smoothly and no vortex emerges. This feature for inviscid flows is quite different from that for viscous flows. Fig. 13 shows the pressure flied for this inviscid flow. As can be seen from the figure, the pressure field is basically symmetric about x axis and y axis. All these observations agree

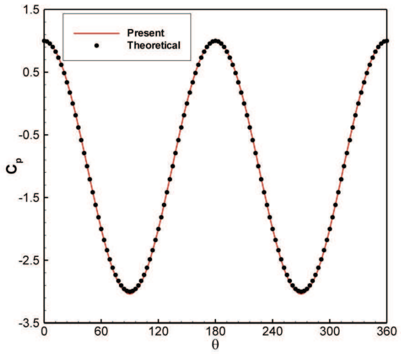


Figure 11: Comparison of pressure coefficient distribution on cylinder surface for the inviscid flow past a circular cylinder.

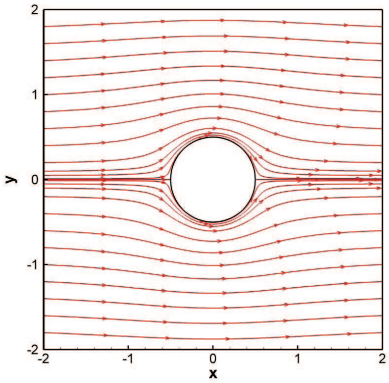


Figure 12: Streamlines of the inviscid flow past a circular cylinder.

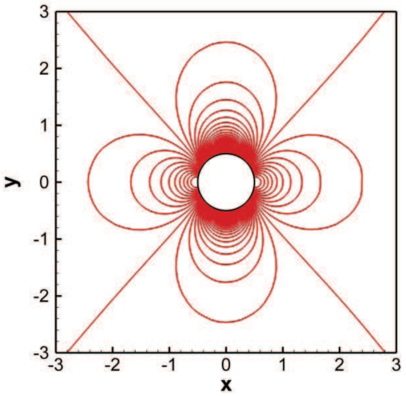


Figure 13: Pressure contours for the inviscid flow past a circular cylinder.

well with theoretical results. This well demonstrates the capability of LBFS for simulation of inviscid flows.

5 Conclusions

In this paper, a lattice Boltzmann flux solver (LBFS) is presented for simulation of incompressible viscous and inviscid flows. The work is based on Chapman-Enskog expansion analysis, which links Navier-Stokes equations and lattice Boltzmann equation. By applying the finite volume method to discretize the Navier-Stokes equations recovered from lattice Boltzmann equation through the Chapman-Enskog analysis, the conservative flow variables at cell centers can be updated by marching in time. The key step in LBFS is to evaluate the inviscid and viscous fluxes simultaneously at cell interface by local reconstruction of LBE solution within a short streaming step from macroscopic flow variables at cell centers.

The present LBFS is well validated by its application to simulate the viscous decaying vortex flow, the driven cavity flow and the flow past a circular cylinder, as well as the inviscid flow past a circular cylinder. Numerical results show that LBFS has the second order of accuracy in space, and the selected streaming distance in local reconstruction of LBE solution has no effect on the solution accuracy. For the driven cavity flow at $Re=1000$, when the same non-uniform mesh is used, LBFS requires less iteration numbers and about one-third of computational time on Lenovo Laptop (2.53GHz and 4G RAM) than the Taylor series expansion- and least square-based lattice Boltzmann method (TLLBM). The solver can be well applied to problems with non-uniform mesh and curved boundary. It can also be used to simulate both the viscous and inviscid flows. It is believed that LBFS has a great potential for solving various incompressible flow problems in practice.

References

[1] A. J. Chorin, *A numerical method for solving incompressible viscous flow problems*, J. Comput. Phys., 135 (1997), pp. 118–125.

[2] A. J. Chorin, *Numerical solution of the Navier-Stokes equation*, Math. Comput., 22 (1971), pp. 745–762.

[3] S. V. Patankar and D. B. Spadling, *A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows*, Int. J. Heat Mass Trans., 15 (1972), pp. 1787–1806.

[4] S. V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere, New York, 1981.

[5] J. P. Van Doormaal and G. D. Raithby, *Enhancement of the simple method for predicting incompressible fluid flows*, Numer. Heat Trans., 7 (1984), pp. 147–163.

[6] R. I. Issa, *Solution of the implicit discretized fluid flow equations by operator splitting*, J. Comput. Phys., 62 (1985), pp. 40–65.

[7] J. Kim and P. Moin, *Application of a fractional-step method to incompressible Navier-Stokes equations*, J. Comput. Phys., 59 (1985), pp. 308–323.

[8] J. B. Bell, P. Colella and H. M. Glaz, *A second order projection method for the incompressible Navier-Stokes equations*, J. Comput. Phys., 85 (1989), pp. 257–283.

[9] D. L. Brown, R. Cortez and M. L. Minion, *Accurate projection methods for the incompressible Navier-Stokes equations*, J. Comput. Phys., 168 (2001), pp. 464–499.

[10] S. Chen, H. Chen, D. O. Martinez and W. H. Matthaeus, *Lattice Boltzmann model for simulation of magnetohydrodynamics*, Phys. Rev. Let., 67 (1991), pp. 3776–3779.

[11] Y. H. Qian, D. d'Humières and P. Lallemand, *Lattice BGK models for Navier-Stokes equation*, Europhys. Lett., 17 (1992), pp. 479–484.

[12] X. He, L. S. Luo and M. Dembo, *Some progress in lattice Boltzmann method: Part I. Nonuniform mesh grids*, J. Comput. Phys., 129 (1996), pp. 357.

[13] S. Chen and G. Doolen, *Lattice Boltzmann method for fluid flows*, Ann. Rev. Fluid Mech., 30 (1998), pp. 329–364.

[14] L. S. Luo, *Unified theory of the lattice Boltzmann models for nonideal gases*, Phys. Rev. Lett., 81 (1998), pp. 1618–1621.

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

[16] P. Lallemand and L. S. Luo, *Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, Galilean invariance, and stability*, Phys. Rev. E, 61 (2000), pp. 6546–6562.

[17] Z. L. Guo, B. C. Shi and N. C. Wang, *Lattice BGK model for incompressible Navier-Stokes equation*, J. Comput. Phys., 165 (2000), pp. 288–306.

[18] C. Shu, Y. T. Chew and X. D. Niu, *Least square-based LBM: a meshless approach for simulation of flows with complex geometry*, Phys. Rev. E, 64 (2001), 045701.

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

[20] S. Succi, *Mesoscopic modeling of slip motion at fluid-solid interfaces with heterogeneous catalysis*, Phys. Rev. Lett., 89 (2002), 064502.

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

[22] Y. H. Zhang, X. J. Gu, R. W. Barber and D. R. Emerson, *Capturing Knudsen layer phenomena using a lattice Boltzmann model*, Phys. Rev. E, 74 (2006), 046704.

[23] X. Shan, X. F. Yuan and H. Chen, *Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation*, J. Fluid Mech., 550 (2006), pp. 413–441.

[24] C. K. Aidun and J. R. Clausen, *Lattice-Boltzmann method for complex flows*, Ann. Rev. Fluid Mech., 42 (2010), pp. 439–472.

[25] M. R. Swift, E. Orlandini, W. Osborn and J. M. Yoemans, *Lattice Boltzmann simulations of liquid-gas and binary fluid systems*, Phys. Rev. E, 54 (1996), pp. 5041–5052.

[26] X. He, S. Chen and G. D. Doolen, *A novel thermal model for the lattice Boltzmann method in incompressible limit*, J. Comput. Phys., 146 (1998), pp. 282–300.

[27] T. Kataoka and M. Tsutahara, *Lattice Boltzmann method for the compressible Navier-Stokes equations with flexible specific-heat ratio*, Phys. Rev. E, 69 (2004), R035701.

[28] Y. H. Zhang, R. Qin and D. R. Emerson, *Lattice Boltzmann simulation of rarefied gas flows in microchannels*, Phys. Rev. E, 71 (2005), 047702.

[29] Z. L. Guo, P. Asinari and C. G. Zheng, *Lattice Boltzmann equation for microscale gas flows of binary mixtures*, Phys. Rev. E, 79 (2009), 026702.

[30] M. Mendoza, B. M. Boghosian, H. J. Herrmann1 and S. Succi, *Fast lattice Boltzmann solver for relativistic hydrodynamics*, Phys. Rev. Lett., 105 (2010), 014502.

[31] M. R. Wang and Q. J. Kang, *Modeling electrokinetic flows in microchannels using coupled lattice Boltzmann method*, J. Comput. Phys., 229 (2010), pp. 728–744.

[32] J. Wu and C. Shu, *A solution-adaptive lattice Boltzmann method for two-dimensional incompressible viscous flows*, J. Comput. Phys., 230 (2011), pp. 2246–2269.

[33] X. J. Li, R. G. Zhao and C. W. Zhong, *Novel immersed boundary-lattice Boltzmann method based on feedback law*, Transactions of Nanjing University of Aeronautics and Astronautics, 29 (2012), pp. 179–186.

[34] H. B. Huang, Y. F. Wu and X. Y. Lu, *Shear viscosity of dilute suspensions of ellipsoidal particles with a lattice Boltzmann method*, Phys. Rev. E, 86 (2012), 046305.

[35] L. Q. Zhao, J. H. Sun and C. Y. Xu, *Flow field analyses of plane jet at low Reynolds number using lattice Boltzmann method*, Transactions of Nanjing University of Aeronautics and Astronautics, 29 (2012), pp. 199–206.

[36] H. H. Liu, Y. H. Zhang and A. J. Valocchi, *Modeling and simulation of thermocapillary flows using lattice Boltzmann method*, J. Comput. Phys., 231 (2012), pp. 4433–4453.

[37] K. Xu, *Gas-kinetic schemes for unsteady compressible flow simulations*, VKI Report, 1998-03 (1998).

[38] H. Z. Tang and K. Xu, *A high-order gas-kinetic method for multidimensional ideal magnetohydrodynamics*, J. Comput. Phys., 165 (2000), pp. 69–88.

[39] K. Xu, *A gas-kinetic BGK scheme for the Navier-Stokes equations, and its connection with artificial dissipation and Godunov method*, J. Comput. Phys., 171 (2001), pp. 289–335.

[40] S. Z. Chen, K. Xu, C. B. Li and Q. D. Cai, *A unified gas kinetic scheme with moving mesh and velocity space adaptation*, J. Comput. Phys., 231 (2012), pp. 6643–6664.

[41] U. Ghia, K. N. Chia and C. T. Shin, *High-Re solutions for incompressible flow using the Navier-Stokes equations a multigrid method*, J. Comput. Phys., 48 (1982), pp. 387–411.

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

[43] F. Nieuwstadt and H. B. Keller, *Viscous flow past circular cylinders*, Comput. Fluids, 1 (1973), pp. 59–71.

[44] X. He and G. D. Doolen, *Lattice Boltzmann method on curvilinear coordinates system: flow around a circular cylinder*, J. Comput. Phys., 134 (1997), pp. 306–315.

[45] 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 (2007), pp. 1064–1094.

[46] J. Park, K. Kwon and H. Choi, *Numerical solutions of flow past a circular cylinder at Reynolds number up to 160*, KSME Int. J., 12 (1998), pp. 1200.

[47] M. Braza, P. Chassaing and H. H. Minh, *Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder*, J. Fluid Mech., 165 (1986), pp. 79–130.

[48] M. G. Benson, P. G. Bellamy-Knights, J. H. Gerrard and I. Gladwell, *A viscous splitting algorithm applied to low Reynolds number flows round a circular cylinder*, J. Fluids Struct., 3 (1989), pp. 439–479.

[49] H. Ding, C. Shu, K. S. Yeo and D. Xu, *Simulation of incompressible viscous flows past a circular cylinder by hybrid FD scheme and meshless least square-based finite difference method*, Comput. Meth. Appl. Mech. Eng., 193 (2004), pp. 727–744.