Taylor-series expansion and least-squares-based lattice Boltzmann method: Two-dimensional formulation and its applications

C. Shu, X. D. Niu, and Y. T. Chew
Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 117576
(Received 1 August 2001; published 6 March 2002)

An explicit lattice Boltzmann method (LBM) is developed in this paper to simulate flows in an arbitrary geometry. The method is based on the standard LBM, Taylor-series expansion, and the least-squares approach. The final formulation is an algebraic form and essentially has no limitation on the mesh structure and lattice model. Theoretical analysis for the one-dimensional (1D) case showed that the version of the LBM could recover the Navier-Stokes equations with second order accuracy. A generalized hydrodynamic analysis is conducted to study the wave-number dependence of shear viscosity for the method. Numerical simulations of the 2D lid-driven flow in a square cavity and a polar cavity flow as well as the ''no flow'' simulation in a square cavity have been carried out. Favorable results were obtained and compared well with available data in the literature, indicating that the present method has good prospects in practical applications.

DOI: 10.1103/PhysRevE.65.036708 PACS number(s): 45.50.−j, 47.11.+j

I. INTRODUCTION

The development of the lattice Boltzmann method (LBM) as an alternative computational fluid dynamics approach has attracted more and more attention in recent years [1–8]. However, because of the essential restriction of the standard lattice Boltzmann equation (LBE) to the lattice uniformity, broad application of the LBM in engineering has been greatly hampered. For many practical problems, an irregular grid or a meshless structure is always preferable due to the fact that curved boundaries can be described more accurately, and that computational resources can be used more efficiently with it.

The drawback of the standard LBE restricting to the lattice uniformity comes from its precursor—the lattice-gas cellular automata (LGCA) [9–10]. In the LGCA, the symmetry of the lattice, which guarantees the isotropy of the fourth tensor consisting of particle velocities, is an essential condition to obtain the Navier-Stokes equations. By this condition, a particle at one lattice node must move to its neighboring node in one time step. This is the condition of lattice uniformity. Although the LBE [11–12] with the Bhatnagar-Gross-Krook (BGK) [13] model has made great improvements to the LGCA, it also inherits the feature of lattice uniformity, which makes it macroscopically similar to a uniform Cartesian-grid solver.

Theoretically, the feature of lattice uniformity is not necessary to keep because the distribution functions are continuous in physical space. Currently, there are two ways to improve the standard LBM so that it can be applied to complex problems. One is the interpolation-supplemented LBM (IS-LBM) proposed by He, Luo, and Dembo and He and Doolen [14–16]. The other is based on the solution of a differential lattice Boltzmann equation (LBE). For complex problems, the differential LBE can be solved by the finite difference (FDLBE) method with the help of coordinate transformation [17] or by the finite volume (FVLBE) approach [18–21]. Numerical experience has shown that these methods have a good capability in real applications. However, the interpolation-supplemented lattice Boltzmann equation (IS-LBE) requires an extra computational effort for interpolation at every time step and it also has a strict restriction on the selection of interpolation points, which requires upwind nine points for two-dimensional problems and upwind 27 points for three-dimensional problems if a structured mesh is used. For the FDLBE and FVLBE methods, one needs to select efficient approaches such as upwind schemes to do numerical discretization in order to get the stable solution. As a consequence, the computational efficiency greatly depends on the selected numerical scheme. In addition, the numerical diffusion may affect the accuracy of the results, especially in the region where the flow gradient is large.

In order to implement the LBE more efficiently for flows with arbitrary geometry, we propose in this paper a version of LBM, which is based on the standard LBM, the well-known Taylor-series expansion, the idea of developing the Runge-Kutta method [22], and the least-squares approach [23]. The final form of our method is an algebraic formulation, in which the coefficients only depend on the coordinates of mesh points and lattice velocity, and are computed in advance. The method is also free of lattice models. To validate the proposed method, some theoretical analysis and a generalized hydrodynamic analysis are presented. Numerical simulations include ''no flow'' in the closed square cavity, a lid-driven flow in a square cavity, and a polar cavity flow. All simulations use a nonuniform mesh with mesh points strongly clustering to the boundary. The obtained numerical results are very accurate. Numerical experience showed that the present method is an efficient and flexible approach for practical application.

II. TAYLOR-SERIES EXPANSION- AND LEAST-SQUARES-BASED LBM

The method developed in this paper is based on the well-known fact that the density distribution function is a continuous function in physical space and can be well defined in any mesh system. Let us start with the standard LBM. The two-

dimensional, standard LBE with BGK approximation can be written as

$$f_\alpha(x+e_{\alpha x}\delta t, y+e_{\alpha y}\delta t, t+\delta t) = f_\alpha(x,y,t) + \frac{f_\alpha^{\mathrm{eq}}(x,y,t) - f_\alpha(x,y,t)}{\tau},$$

$$\alpha = 0.1, \ldots, N, \tag{1}$$

where τ is the single relaxation time; f_α is the density distribution function along the α direction; f_α^{eq} is its corresponding equilibrium state, which depends on the local macroscopic variables such as density ρ and velocity $\mathbf{U}(u,v)$; $\boldsymbol{\delta}_t$ is the time step and $\mathbf{e}_\alpha(e_{\alpha x}, e_{\alpha y})$ is the particle velocity in the α direction; N is the number of discrete particle velocities. Obviously, the standard LBE consists of two steps: collision and streaming. The macroscopic density ρ and momentum density $\rho\mathbf{U}$ are defined as

$$\rho = \sum_{\alpha=0}^{N} f_\alpha, \quad \rho\mathbf{U} = \sum_{\alpha=0}^{N} f_\alpha \mathbf{e}_\alpha. \tag{2}$$

Suppose that a particle is initially at the grid point (x, y, t). Along the α direction, this particle will stream to the position $(x+e_{\alpha x}\delta t, y+e_{\alpha y}\delta t, t+\delta t)$. For a uniform lattice, $\delta x = e_{\alpha x}\delta_t$, $\delta y = e_{\alpha y}\delta_t$. So, $(x+e_{\alpha x}\delta_t, y+e_{\alpha y}\delta_t)$ is at the grid point. In other words, Eq. (1) can be used to update the density distribution functions exactly at the grid points. However, for a nonuniform grid, $(x+e_{\alpha x}\delta_t, y+e_{\alpha y}\delta_t)$ is usually not at the grid point $(x+\delta x, y+\delta y)$. In the numerical simulation, we are only interested in the density distribution function at the mesh point for all the time levels. So, the macroscopic properties such as the density and flow velocity can be evaluated at every mesh point. To get the density distribution function at the grid point $(x+\delta x, y+\delta y)$ and the time level $t+\delta_t$, we need to apply the Taylor-series expansion or other interpolation techniques such as the one used by He, Luo, and Dembo [14]. In this paper, the Taylor-series expansion is used. Note that the time level for the position $(x+e_{\alpha x}\delta_t, y+e_{\alpha y}\delta_t)$ and the grid point $(x+\delta x, y+\delta y)$ is the same, that is, $t+\delta_t$. So, the expansion in the time direction is not necessary. As shown in Fig. 1, for simplicity, we let point A represent the grid point (x_A, y_A, t), point A' represent the position $(x_A+e_{\alpha x}\delta_t, y_A+e_{\alpha y}\delta_t, t+\delta t)$, and point P represent the position $(x_P, y_P, t+\delta t)$ with $x_P = x_A+\delta x$, $y_P = y_A+\delta y$. So Eq. (1) gives

$$f_\alpha(A', t+\delta t) = f_\alpha(A,t) + [f_\alpha^{\mathrm{eq}}(A,t) - f_\alpha(A,t)]/\tau. \tag{3}$$

For the general case, A' may not coincide with the mesh point P. At first, we consider the Taylor-series expansion with truncation to the first-order derivative terms. So, $f_\alpha(A', t+\delta t)$ can be approximated by the corresponding function and its derivatives at the mesh point P as

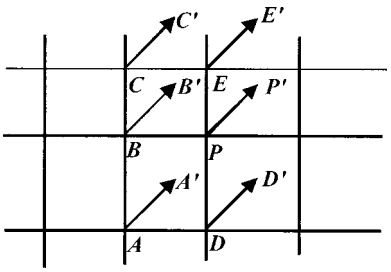


FIG. 1. Configuration of particle movement along the α direction.

$$f_\alpha(A', t+\delta t) = f_\alpha(P, t+\delta t) + \Delta x_A \frac{\partial f_\alpha(P, t+\delta t)}{\partial x} + \Delta y_A \frac{\partial f_\alpha(P, t+\delta t)}{\partial y} + O[(\Delta x_A)^2, (\Delta y_A)^2], \tag{4}$$

where $\Delta x_A = x_A + e_{\alpha x}\delta t - x_P$, $\Delta y_A = y_A + e_{\alpha y}\delta t - y_P$. Note that the above approximation has a truncation error of the second order. Substituting Eq. (4) into Eq. (1) gives

$$f_\alpha(P, t+\delta t) + \Delta x_A \frac{\partial f_\alpha(P, t+\delta t)}{\partial x} + \Delta y_A \frac{\partial f_\alpha(P, t+\delta t)}{\partial y} = f_\alpha(A,t) + \frac{f_\alpha^{\mathrm{eq}}(A,t) - f_\alpha(A,t)}{\tau}. \tag{5}$$

It is indicated that Eq. (5) is a first-order differential equation, which only involves two mesh points A and P. When a uniform grid is used, $\Delta x_A = \Delta y_A = 0$ and Eq. (5) is reduced to the standard LBE (1). Solving Eq. (5) can provide the density distribution functions at all the mesh points. In this paper, we try to develop an explicit formulation to update the distribution function. In fact, our development is inspired from the Runge-Kutta method [22]. As we know, the Runge-Kutta method is developed to improve the Taylor-series method in the solution of ordinary differential equations (ODEs). Like Eq. (5), the Taylor series method involves evaluation of different orders of derivatives to update the functional value at the next time level. For a complicated expression of given ODEs, this application is very difficult. To improve the Taylor-series method, the Runge-Kutta method evaluates the functional values at some intermediate points and then combines them (through the Taylor-series expansion) to form a scheme with the same order of accuracy. With this idea in mind, we look at Eq. (5). We know that at the time level $t+\delta t$, the density distribution function and its derivatives at the mesh point P are all unknowns. So, Eq. (5) has three unknowns in total. To solve for the three unknowns, we need three equations. However, Eq. (5) just provides one equation. We need an additional two equations to close the system. As shown in Fig. 1, we can see that along the α direction, the particles at two mesh points P, B at the time level t will stream to the positions P', B' at the time level $t+\delta t$. The distribution functions at these positions can be computed through Eq. (1), which are given below

$$f_\alpha(P',t+\delta t)=f_\alpha(P,t)+[f_\alpha^{\mathrm{eq}}(P,t)-f_\alpha(P,t)]/\tau, \quad (6)$$

$$f_\alpha(B',t+\delta t)=f_\alpha(B,t)+[f_\alpha^{\mathrm{eq}}(B,t)-f_\alpha(B,t)]/\tau. \quad (7)$$

Using Taylor-series expansion with truncation to the first-order derivative terms, $f_\alpha(P',t+\delta t)$, $f_\alpha(B',t+\delta t)$ in above equations can be approximated by the function and its derivatives at the mesh point P. As a result, Eqs. (6),(7) can be reduced to

$$f_\alpha(P,t+\delta t)+\Delta x_P\frac{\partial f_\alpha(P,t+\delta t)}{\partial x}+\Delta y_P\frac{\partial f_\alpha(P,t+\delta t)}{\partial y}=f_\alpha(P,t)+\frac{f_\alpha^{\mathrm{eq}}(P,t)-f_\alpha(P,t)}{\tau}, \quad (8)$$

$$f_\alpha(P,t+\delta t)+\Delta x_B\frac{\partial f_\alpha(P,t+\delta t)}{\partial x}+\Delta y_B\frac{\partial f_\alpha(P,t+\delta t)}{\partial y}=f_\alpha(B,t)+\frac{f_\alpha^{\mathrm{eq}}(B,t)-f_\alpha(B,t)}{\tau}, \quad (9)$$

where

$$\Delta x_P=e_{\alpha x}\delta t,\ \ \Delta y_P=e_{\alpha y}\delta t,$$

$$\Delta x_B=x_B+e_{\alpha x}\delta t-x_P,\Delta y_B=y_B+e_{\alpha y}\delta t-y_P.$$

Equations (5) and (8),(9) form a system to solve for three unknowns. The solution of this system gives

$$f_\alpha(P,t+\delta t)=\Delta_P/\Delta, \quad (10)$$

where

$$\Delta=\Delta x_A\Delta y_B-\Delta x_B\Delta y_A+\Delta x_B\Delta y_P-\Delta x_P\Delta y_B+\Delta x_P\Delta y_A-\Delta x_A\Delta_P,$$

$$\Delta_P=(\Delta x_A\Delta y_B-\Delta x_B\Delta y_A)g_P+(\Delta x_B\Delta y_P-\Delta x_P\Delta y_B)g_A+(\Delta x_P\Delta y_A-\Delta x_A\Delta y_P)g_B,$$

$$g_P=f_\alpha(P,t)+[f_\alpha^{\mathrm{eq}}(P,t)-f_\alpha(P,t)]/\tau,$$

$$g_A=f_\alpha(A,t)+[f_\alpha^{\mathrm{eq}}(A,t)-f_\alpha(A,t)]/\tau,$$

$$g_B=f_\alpha(B,t)+[f_\alpha^{\mathrm{eq}}(B,t)-f_\alpha(B,t)]/\tau.$$

Note that g_P, g_A, g_B are actually the postcollision state of the distribution function at the time level t and the mesh point P, A, and B respectively. Equation (10) has second order of truncation error, which may introduce a large numerical diffusion. To improve the accuracy of numerical computation, we need to truncate the Taylor-series expansion to the second-order derivative terms. For the two-dimensional case, this expansion involves six unknowns, that is, one distribution function, two first-order derivatives, and three second-order derivatives at the time level $t+\delta t$. To solve for these unknowns, we need six equations to close the system. This can be done by applying the second-order Taylor-series expansion at six points. As shown in Fig. 1, the particles at six mesh points P, A, B, C, D, and E at the time level t will stream to positions P', A', B', C', D', and E' at the time level $t+\delta t$. The distribution functions at these positions can be computed through Eq. (1). Then by using the second-order Taylor-series expansion at these positions in terms of the distribution function and its derivatives at the mesh point P, we can get the following equation system:

$$g_i=\{s_i\}^T\{V\}=\sum_{j=1}^{6}s_{i,j}V_j,i=P,A,B,C,D,E, \quad (11)$$

where

$$g_i=f_\alpha(x_i,y_i,t)+\lfloor f_\alpha^{\mathrm{eq}}(x_i,y_i,t)-f_\alpha(x_i,y_i,t)\rfloor/\tau,$$

$$\{s_i\}^T=\{1,\Delta x_i,\Delta y_i,(\Delta x_i)^2/2,(\Delta y_i)^2/2,\Delta x_i\Delta y_i\},$$

$$\{V\}=\{f_\alpha,\partial f_\alpha/\partial x,\partial f_\alpha/\partial y,\partial^2 f_\alpha/\partial x^2,\partial^2 f_\alpha/\partial^2 y,\partial^2 f_\alpha/\partial x\partial y\}^T.$$

g_i is the postcollision state of the distribution function at the ith point and the time level t, $\{s_i\}^T$ is a vector with six elements formed by the coordinates of mesh points, $\{V\}$ is the vector of unknowns at the mesh point P and the time level $t+\delta t$, which also has six elements, $s_{i,j}$ is the jth element of the vector $\{s_i\}^T$ and V_j is the jth element of the vector $\{V\}$. Our target is to find its first element $V_1=f_\alpha(P,t+\delta t)$. Equation system (11) can be put into the following matrix form:

$$[S]\{V\}=\{g\}, \quad (12)$$

where

$$\{g\}=\{g_P,g_A,g_B,g_C,g_D,g_E\}^T,$$

$$[S]=[s_{i,j}]=\begin{bmatrix}\{s_P\}^T\\ \{s_A\}^T\\ \{s_B\}^T\\ \{s_C\}^T\\ \{s_D\}^T\\ \{s_E\}^T\end{bmatrix}=\begin{bmatrix}1 & \Delta x_P & \Delta y_P & (\Delta x_P)^2/2 & (\Delta y_P)^2/2 & \Delta x_P\Delta y_P\\ 1 & \Delta x_A & \Delta y_A & (\Delta x_A)^2/2 & (\Delta y_A)^2/2 & \Delta x_A\Delta y_A\\ 1 & \Delta x_B & \Delta y_B & (\Delta x_B)^2/2 & (\Delta y_B)^2/2 & \Delta x_B\Delta y_B\\ 1 & \Delta x_C & \Delta y_C & (\Delta x_C)^2/2 & (\Delta y_C)^2/2 & \Delta x_C\Delta y_C\\ 1 & \Delta x_D & \Delta y_D & (\Delta x_D)^2/2 & (\Delta y_D)^2/2 & \Delta x_D\Delta y_D\\ 1 & \Delta x_E & \Delta y_E & (\Delta x_E)^2/2 & (\Delta y_E)^2/2 & \Delta x_E\Delta y_E\end{bmatrix},$$

$$\Delta x_C=x_C+e_{\alpha x}\delta t-x_P,\ \ \Delta y_C=y_C+e_{\alpha y}\delta t-y_P,$$

$$\Delta x_D=x_D+e_{\alpha x}\delta t-x_P,\ \ \Delta y_D=y_D+e_{\alpha y}\delta t-y_P,$$

$$\Delta x_E=x_E+e_{\alpha x}\delta t-x_P,\ \ \Delta y_E=y_E+e_{\alpha y}\delta t-y_P.$$

The expressions of Δx_P, Δy_P, Δx_A, Δy_A, Δx_B, Δy_B have been given previously. Since $[S]$ is a (6×6)-dimensional matrix, it is very difficult to get an analytical expression for the solution of equation system (12). We need to use a numerical algorithm to get the solution. Note that the matrix $[S]$ only depends on the coordinates of mesh points, which can be computed once and stored for the application of Eq. (12) at all time levels.

In practical applications, it was found that the matrix $[S]$ might be singular or ill conditioned. To overcome this difficulty and make the method be more general, we introduce the least-squares approach to optimize the approximation by Eq. (11). Equation (11) has six unknowns (elements of the vector $\{V\}$). If Eq. (11) is applied at more than six mesh points, then the system is over determined. For this case, the unknown vector can be decided from the least-squares method. For simplicity, let the mesh point P be represented by the index $i=0$, and its adjacent points be represented by index $i=1, 2, \ldots, M$, where M is the number of neighboring points around P and it should be larger than five. At each point, we can define an error in terms of Eq. (11), that is,

$$\mathrm{err}_i = g_i - \sum_{j=1}^{6} s_{i,j} V_j, \quad i=0,1,2,\ldots,M. \tag{13}$$

The square sum of all the errors are defined as

$$E = \sum_{i=0}^{M} \mathrm{err}_i^2 = \sum_{i=0}^{M} \left(g_i - \sum_{j=1}^{6} s_{i,j} V_j \right)^2. \tag{14}$$

To minimize the error E, we need to set $\partial E/\partial V_k = 0, k = 1, 2, \ldots, 6$, which leads to

$$[S]^T[S]\{V\} = [S]^T\{g\}, \tag{15}$$

where $[S]$ is a $(M+1)\times 6$-dimensional matrix, which is given as

$$[S] = \begin{bmatrix} 1 & \Delta x_0 & \Delta y_0 & (\Delta x_0)^2/2 & (\Delta y_0)^2/2 & \Delta x_0 \Delta y_0 \\ 1 & \Delta x_1 & \Delta y_1 & (\Delta x_1)^2/2 & (\Delta y_1)^2/2 & \Delta x_1 \Delta y_1 \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ 1 & \Delta x_M & \Delta y_M & (\Delta x_M)^2/2 & (\Delta y_M)^2/2 & \Delta x_M \Delta y_M \end{bmatrix}_{(M+1)\times 6}$$

and $\{g\} = \{g_0, g_1, \ldots, g_M\}^T$.

The Δx and Δy values in the matrix $[S]$ are given as

$$\Delta x_0 = e_{\alpha x}\delta t, \quad \Delta y_0 = e_{\alpha y}\delta t, \tag{16a}$$

$$\Delta x_i = x_i + e_{\alpha x}\delta t - x_0, \quad \Delta y_i = y_i + e_{\alpha y}\delta t - y_0, \quad \text{for } i=1,2,\ldots,M. \tag{16b}$$

Clearly, when the coordinates of mesh points are given, and the particle velocity and time step size are specified, the matrix $[S]$ is determined. Then from Eq. (15), we obtain

$$\{V\} = ([S]^T[S])^{-1}[S]^T\{g\} = [A]\{g\}. \tag{17}$$

Note that $[A]$ is a $6\times(M+1)$-dimensional matrix. From Eq. (17), we can have

$$f_\alpha(x_0, y_0, t+\delta t) = V_1 = \sum_{k=1}^{M+1} a_{1,k}^{\alpha} g_{k-1}^{\alpha}, \tag{18}$$

where $a_{1,k}^{\alpha}$ are the elements of the first row of the matrix $[A]$, which are precomputed before the LBM is applied. Therefore, little computational effort is introduced as compared with the standard LBE. Note that the function g is evaluated at the time level t. So, Eq. (18) is actually an explicit form to update the distribution function at the time level $t+\delta t$ for any mesh point. In the above process, there is no requirement for the selection of neighboring points. In other words, Eq. (18) has nothing to do with the mesh structure. It only needs to know the coordinates of the mesh points. Thus, we can say that Eq. (18) is basically a meshless form.

III. D2Q7 SPEED MODEL AND IMPLEMENTATION OF BOUNDARY CONDITIONS ON THE WALL

It can be seen that Eq. (18) is applied along the α direction. Here, α can be any direction. This implies that Eq. (18) can be uniformly applied to different lattice models. In this paper, we select the D2Q7 model. The configuration of this model is shown in Fig. 2. The discrete velocity of this model is defined as

$$\mathbf{e}_\alpha = \begin{cases} (0,0), & \alpha = 0 \\ \{\cos[(\alpha-1)\pi/3], \sin[(\alpha-1)\pi/3]\}c, & \alpha = 1,2,\ldots,6. \end{cases} \tag{19}$$

The parameter c is the particle streaming speed. The fluid kinetic viscosity is given by

$$\nu = \frac{(2\tau-1)}{8} c^2 \delta_t. \tag{20}$$

The equilibrium density distribution f_α^{eq} is chosen to be

$$f_\alpha^{\mathrm{eq}} = \rho\left[\frac{1}{2} + \frac{1}{6}\left(2\frac{\mathbf{e}_\alpha\cdot\mathbf{U}}{c^2} + 4\left(\frac{\mathbf{e}_\alpha\cdot\mathbf{U}}{c^2}\right)^2 - \frac{\mathbf{U}^2}{c^2}\right)\right]. \tag{21}$$

The speed of sound of this model is $c_s = c/2$, and the equation of state is $P = \rho c_s^2$ for an ideal gas. Although the proposed method has a meshless feature, it is recommended to use a structured grid. This is because in our method, only the coordinates of mesh points are involved. When a structured grid is used, it is much easier for us to define the coordinates of mesh points. In our application, we use a structured grid, and take M as eight for convenience. As shown in Fig. 3, for an internal mesh point (i, j) [noted as "0" in Eq. (18)], the eight neighboring points are taken as $(i-1, j-1)$; $(i-1, j)$; $(i-1, j+1)$; $(i, j-1)$; $(i, j+1)$, $(i+1, j-1)$; $(i+1, j)$; and $(i+1, j+1)$. Therefore, at each mesh point, we only need to store nine coefficients $a_{1,k}$, $k = 1,2,\ldots,9$ before Eq. (18) is applied.

Implementation of boundary conditions is an essential issue in LBM. Many kinds of boundary treatments have been published in the past few years [24–31]. Although these treatments are successful in a number of physical situations, we intend to believe that a complete halfway wall bounce-back condition [30–31] is the most simple and efficient in practicle applications.

The complete halfway wall bounce-back condition, which originated from LGCA, assigns each f_α the value of the f_α in its opposite direction with no relaxation on the bounce-back points. The treatment is independent of the direction, which gives us more convenience in treating complicated boundary problems. The complete halfway wall bounce-back condition has second order of accuracy because macroscopic quantities such as stress force is evaluated on the halfway wall between the bounce-back row and the first flow row. As shown in Fig. 2, for the D2Q7 model, at a boundary point, f_4, f_5, and f_6 point to the flow field from the wall, which will be determined from the boundary condition. f_1, f_2, and f_3 are computed by streaming from points inside the flow field. So, using the halfway wall bounce-back condition, f_4, f_5, and f_6 are evaluated as

$$f_4 = f_1, \qquad f_5 = f_2, \qquad f_6 = f_3. \tag{22}$$

IV. SOME THEORETICAL ANALYSIS OF THE PRESENT METHOD

To illustrate the accuracy of the scheme and its applicable range, a mathematical analysis is carried out in this section, and it will serve as a theoretical justification for the present method. For simplicity, we take the one-dimensional model to illustrate our analysis. Under this consideration, the standard LBE becomes

$$f_\alpha(x + e_\alpha\delta t,\ t+\delta t) = f_\alpha(x,t) + \frac{f_\alpha^{\mathrm{eq}}(x,\ t) - f_\alpha(x,\ t)}{\tau}. \tag{23}$$

Using Taylor-series expansion, we have

$$\begin{aligned} &f_\alpha(x + e_\alpha\delta t,\ t+\delta t) \\ &\quad = f_\alpha(x,t) + \delta t\frac{\partial f_\alpha}{\partial t} + e_\alpha\delta t\frac{\partial f_\alpha}{\partial x} + \frac{\delta t^2}{2}\frac{\partial^2 f_\alpha}{\partial t^2} \\ &\qquad + \frac{(e_\alpha\delta t)^2}{2}\frac{\partial^2 f_\alpha}{\partial x^2} + e_\alpha\delta t^2\frac{\partial^2 f_\alpha}{\partial t\partial x} + \frac{\delta t^3}{6} \\ &\qquad \times\left(\frac{\partial^3 f_\alpha}{\partial t^3} + 3e_\alpha\frac{\partial^3 f_\alpha}{\partial t^2\partial x} + 3e_\alpha^2\frac{\partial^3 f_\alpha}{\partial t\partial x^2} + e_\alpha^3\frac{\partial^3 f_\alpha}{\partial x^3}\right) + \cdots. \end{aligned} \tag{24}$$

With Eq. (24), the standard LBE is equivalent to

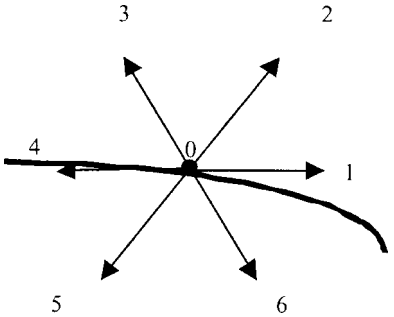


FIG. 2. Schematic plot of the D2Q7 model on a wall boundary (thick curve).

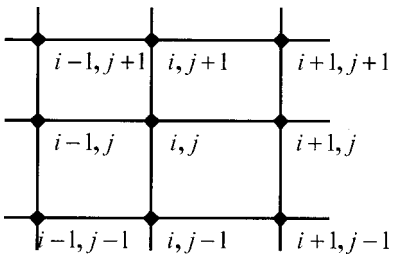


FIG. 3. Schematic plot of neighboring point distribution around the point (i, j).

$$\delta t\frac{\partial f_\alpha}{\partial t}+e_\alpha\delta t\frac{\partial f_\alpha}{\partial x}+\frac{\delta t^2}{2}\left(\frac{\partial^2 f_\alpha}{\partial t^2}+2e_\alpha\frac{\partial^2 f_\alpha}{\partial t\partial x}+e_\alpha^2\frac{\partial^2 f_\alpha}{\partial x^2}\right)=\frac{f_\alpha^{\mathrm{eq}}-f_\alpha}{\tau}+O(\delta t^3). \tag{25}$$

Equation (25) will be used to analyze the present method. We start with the Taylor-series expansion which truncates to the second-order derivative terms. With Taylor-series expansion, Eq. (23) can be written as

$$f_\alpha(x+\delta x,t+\delta t)+(e_\alpha\delta t-\delta x)\frac{\partial f_\alpha}{\partial x}+\frac{(e_\alpha\delta t-\delta x)^2}{2}\frac{\partial^2 f_\alpha}{\partial x^2}=f_\alpha(x,t)+\frac{f_\alpha^{\mathrm{eq}}(x,t)-f_\alpha(x,t)}{\tau}. \tag{26}$$

As shown previously, the truncation error of Eq. (26) is third order. Equation (26) consists of three unknowns, which should be determined by three equations. Suppose that a local mesh point x_i is considered. The three equations can be obtained by applying the Taylor-series expansion at three positions streamed, respectively, from mesh points x_{i-1}, x_i, x_{i+1}. Solving the three equations, we can get the solution

$$f_\alpha(x_i,t+\delta t)=a_i g_i(t)+a_{i-1}g_{i-1}(t)+a_{i+1}g_{i+1}(t), \tag{27}$$

where

$$g_i(t)=f_\alpha(x_i,t)+\frac{f_\alpha^{\mathrm{eq}}(x_i,\ t)-f_\alpha(x_i,t)}{\tau},$$

$$a_i=-\frac{(1-r_i)(1+r_{i+1})}{r_i r_{i+1}},\quad a_{i-1}=\frac{(1+r_{i+1})}{r_i(r_{i+1}+r_i)},$$

$$a_{i+1}=\frac{(1-r_{i+1})}{r_{i+1}(r_{i+1}+r_i)},$$

$$r_i=(x_i-x_{i-1})/(e_\alpha\delta t),\quad r_{i+1}=(x_{i+1}-x_i)/(e_\alpha\delta t).$$

Using second-order Taylor-series expansion, Eq. (27) can be reduced to

$$f_\alpha(x_i,t)+\delta t\frac{\partial f_\alpha}{\partial t}+\frac{\partial t^2}{2}\frac{\partial^2 f_\alpha}{\partial t^2}=g_i(t)-e_\alpha\delta t\frac{\partial g_i}{\partial x}+\frac{(e_\alpha\delta t)^2}{2}\frac{\partial^2 g_i}{\partial x^2}+O[\delta t^3]. \tag{28}$$

On the other hand, from Eq. (23) and Taylor-series expansion, we have

$$g_i(t)=f_\alpha(x_i,t)+\frac{f_\alpha^{\mathrm{eq}}(x_i,t)-f_\alpha(x_i,t)}{\tau}=f_\alpha(x_i+e_\alpha\delta t,t+\delta t)=f_\alpha(x_i,t)+\delta t\frac{\partial f_\alpha}{\partial t}+e_\alpha\delta t\frac{\partial f_\alpha}{\partial x}+O(\delta t^2). \tag{29}$$

Differentiating Eq. (29) with respect to x gives

$$e_\alpha\delta t\frac{\partial g_i}{\partial x}=e_\alpha\delta t\frac{\partial f_\alpha}{\partial x}+e_\alpha\delta t^2\frac{\partial^2 f_\alpha}{\partial t\partial x}+(e_\alpha\delta t)^2\frac{\partial^2 f_\alpha}{\partial x^2}+O(\delta t^3), \tag{30a}$$

$$(e_\alpha\delta t)^2\frac{\partial^2 g_i}{\partial x^2}=(e_\alpha\delta t)^2\frac{\partial^2 f_\alpha}{\partial x^2}+O(\delta t^3). \tag{30b}$$

Finally, by substituting Eqs. (29) and (30) into Eq. (28), we obtain exactly the same differential equation as Eq. (25). As shown in [12] and [32], when the Chapman-Enskog expansion is applied to Eq. (25) with two time scales, the Navier-Stokes (NS) equations can be recovered with second order of accuracy. This indicates that our Taylor-series expansion form can recover the NS equations with second order of accuracy even when a nonuniform mesh ($r_i\neq r_{i+1}$) is used.

Next, we will consider the Taylor-series expansion- and least-squares-based form. For the one-dimensional problem, the second-order Taylor-series expansion has three unknowns, that is, the distribution function and its first- and second-order derivatives at the mesh point x_i and the time level $t+\delta t$. As shown above, to solve for these three unknowns, we need to have three equations, which are obtained by applying the Taylor-series expansion at three positions streamed from three mesh points x_{i-1}, x_i, x_{i+1}. To apply the least-squares approach, the Taylor-series expansion is applied at four positions, which are streamed, respectively, from four mesh points x_{i-1}, x_i, x_{i+1}, x_{i-2}. So, we can get four equations for three unknowns. As shown in the previous section, by using the least-squares approach, the final equation system can be obtained as

$$\begin{pmatrix} 4 & a & \frac{1}{2}b \\ a & b & \frac{1}{2}c \\ \frac{1}{2}b & \frac{1}{2}c & \frac{1}{4}d \end{pmatrix}\begin{pmatrix} f_\alpha(x_i,t+\delta t) \\ e_\alpha\delta t\dfrac{\partial f_\alpha(x_i,t+\delta t)}{\partial x} \\ (e_\alpha\delta t)^2\dfrac{\partial^2 f_\alpha(x_i,t+\delta t)}{\partial x^2} \end{pmatrix}=\begin{pmatrix} \sum_{k=1}^{4} g_{i+k-3} \\ \sum_{k=1}^{4}\delta_k g_{i+k-3} \\ \sum_{k=1}^{4}\delta_k^2 g_{i+k-3} \end{pmatrix}, \tag{31}$$

where $a=\sum_{k=1}^{4}\delta_k$, $b=\sum_{k=1}^{4}\delta_k^2$, $c=\sum_{k=1}^{4}\delta_k^3$, $d=\sum_{k=1}^{4}\delta_k^4$,

$$\delta_1=\frac{e_\alpha\delta t-(x_i-x_{i-2})}{e_\alpha\delta t}=1-r_{i-2},$$

$$\delta_2=\frac{e_\alpha\delta t-(x_i-x_{i-1})}{e_\alpha\delta t}=1-r_i,$$

$$\delta_3=\frac{e_\alpha\delta t}{e_\alpha\delta t}=1,$$

$$\delta_4=\frac{e_\alpha\delta t-(x_i-x_{i+1})}{e_\alpha\delta t}=1+r_{i+1}.$$

The function g is defined as before. The solution of Eq. (31) gives

$$f_\alpha(x_i,t+\delta t)=\frac{1}{\Delta}\sum_{k=1}^{4}[(bc-ad)(b-a\delta_k)-(b^2-ac)(c-a\delta_k^2)]g_{i+k-3}, \quad (32)$$

where $\Delta=(bc-ad)(4b-a^2)-(b^2-ac)(4c-ab)$. Using Taylor-series expansion, g_{i+k-3} can be expressed as

$$\begin{aligned}g_{i+k-3}&=g_i+(x_{i+k-3}-x_i)\frac{\partial g}{\partial x}+\frac{(x_{i+k-3}-x_i)^2}{2}\frac{\partial^2 g}{\partial x^2}\\&\quad+O[(x_{i+k-3}-x_i)^3]\\&=g_i+(\delta_k-1)e_\alpha\delta t\frac{\partial g_i}{\partial x}\\&\quad+\frac{(\delta_k-1)^2}{2}(e_\alpha\delta t)^2\frac{\partial^2 g_i}{\partial x^2}+O(\delta t^3).\end{aligned} \quad (33)$$

By substituting Eq. (33) into Eq. (32), we obtain

$$f_\alpha(x_i,t+\delta t)=s_1g_i+s_2e_\alpha\delta t\frac{\partial g_i}{\partial x}+s_3\frac{(e_\alpha\delta t)^2}{2}\frac{\partial^2 g_i}{\partial x^2}+O[\delta t^3], \quad (34)$$

where

$$s_1=\frac{1}{\Delta}\sum_{k=1}^{4}[(bc-ad)(b-a\delta_k)-(b^2-ac)(c-a\delta_k^2)],$$

$$s_2=\frac{1}{\Delta}\sum_{k=1}^{4}[(bc-ad)(b-a\delta_k)-(b^2-ac)(c-a\delta_k^2)]\times(\delta_k-1),$$

$$s_3=\frac{1}{\Delta}\sum_{k=1}^{4}[(bc-ad)(b-a\delta_k)-(b^2-ac)(c-a\delta_k^2)](\delta_k-1)^2.$$

Furthermore, from the definition of a, b, c, and d, we have

$$\begin{aligned}s_1&=\frac{1}{\Delta}\left[(bc-ad)\sum_{k=1}^{4}(b-a\delta_k)-(b^2-ac)\sum_{k=1}^{4}(c-a\delta_k^2)\right]\\&=\frac{1}{\Delta}[(bc-ad)(4b-a^2)-(b^2-ac)(4c-ab)]=1,\end{aligned} \quad (35a)$$

$$\begin{aligned}s_2&=\frac{1}{\Delta}\left[(bc-ad)\sum_{k=1}^{4}[-b+(b+a)\delta_k-a\delta_k^2]-(b^2-ac)\times\sum_{k=1}^{4}(c\delta_k-c-a(\delta_k^3-\delta_k^2))\right]\\&=\frac{1}{\Delta}[(bc-ad)[-4b+(b+a)a-ab]-(b^2-ac)\times[ac-4c-a(c-b)]]=-1,\end{aligned} \quad (35b)$$

$$\begin{aligned}s_3&=\frac{1}{\Delta}\left[(bc-ad)\sum_{k=1}^{4}\{b(\delta_k^2-2\delta_k+1)-a(\delta_k^3-2\delta_k^2+\delta_k)\}-(b^2-ac)\sum_{k=1}^{4}\{c(\delta_k^2-2\delta_k+1)-a(\delta_k^4-2\delta_k^3+\delta_k^2)\}\right]\\&=\frac{1}{\Delta}[(bc-ad)\{b(b-2a+4)-a(c-2b+a)\}-(b^2-ac)\{c(b-2a+4)-a(d-2c+b)\}]=1.\end{aligned} \quad (35c)$$

The above results show that Eq. (34) can be reduced to exactly the same form as Eq. (28). Equation (28) can recover the NS equation with second order of accuracy. This means that our least-squares-based form can also recover the NS equation with second order of accuracy no matter what the mesh is, uniform or nonuniform.

V. NUMERICAL TESTS

In this section, the proposed Taylor-series expansion- and least-squares-based LBM is validated by some test cases. First, the wave-number dependence of shear viscosity of the present method is studied using a generalized hydrodynamic analysis [33] for a sinusoidal shear wave. The second test case is the ''no flow'' simulation in a square cavity, which is used to check the detailed balance condition of the present method with the use of nonuniform meshes. The other two test cases are the lid-driven flows in a square cavity and in a polar cavity. In these problems, the fluid is bounded by the cavity and is driven by a uniform translation of the lid. The cavity flow cases show rich vortex phenomena at many scales depending on the Reynolds numbers, and there is abundant literature to study the flow configuration. Thus, these problems are ideal test cases for numerical methods devised to simulate viscous flows. In the present paper, the numerical work given by Ghia, Ghia, and Keller [34] for the square cavity flow case, and numerical and experimental work obtained by Fuchs and Tillmark [35] for the polar cav-

ity flow case, are used as a benchmark to evaluate the present results.

Unless otherwise mentioned, all simulations were carried out by using nine mesh points for least-squares optimization. The configuration of these nine mesh points is shown in Fig. 3.

A. Wave-number dependence of shear viscosity

As shown in the previous theoretical analysis, our method does not affect the viscosities in the limit of wave number $\mathbf{k}=0$. Here, it is interesting to study the wave-number dependence of shear viscosity at finite value of $\mathbf{k}$. We will follow the work of Lallemand and Luo [33] to do this study. As shown in [33], the shear viscosity $\nu(\mathbf{k})$ can be computed by

$$\nu(\mathbf{k}) = -\frac{1}{k^2}\mathrm{Re}[\ln z_T(\mathbf{k})], \tag{36}$$

where $\mathbf{k}=(k_x,k_y)$, $k=\sqrt{k_x^2+k_y^2}$, $z_T(\mathbf{k})$ is an eigenvalue corresponding to the hydrodynamic mode of the linearized evolution operator L, which is given as [33]

$$L=\overline{A}^{-1}[I+M^{-1}CM]. \tag{37}$$

In Eq. (37), M is the transformation matrix from the discrete velocity space to the moment space, C is the linearized collision operator, $\overline{A}$ is the advection vector which is a diagonal matrix. For the present method, the transformation matrix M and the collision operator C remain the same as for the standard LBE, but the advection operator $\overline{A}$ is changed. For the D2Q9 model, Lallemand and Luo [33] gave the matrices M and C as

$$M=\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{pmatrix}, \tag{38}$$

$$C=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_2\alpha_2/4 & -s_2 & 0 & s_2\gamma_2V_x/3 & 0 & s_2\gamma_2V_y/3 & 0 & 0 & 0 \\ s_3\alpha_3/4 & 0 & -s_3 & s_3\gamma_4V_x/3 & 0 & s_3\gamma_4V_x/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_5c_1/2 & -s_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & s_7c_1/2 & -s_7 & 0 & 0 \\ 0 & 0 & 0 & 3s_8\gamma_1V_x & 0 & -3s_8\gamma_1V_y & 0 & -s_8 & 0 \\ 0 & 0 & 0 & 3s_9\gamma_3V_y/2 & 0 & 3s_9\gamma_3V_x/2 & 0 & 0 & -s_9 \end{pmatrix}. \tag{39}$$

When the standard LBE is used, the advection operator $\overline{A}$ is given as [33]

$$\overline{A}=\mathrm{diag}(1,p,q,1/p,1/q,pq,q/p,1/pq,p/q), \tag{40}$$

where $p=e^{ik_x}$, $q=e^{ik_y}$. For the present method, the advection operator $\overline{A}$ can be obtained through Eq. (18), and written as

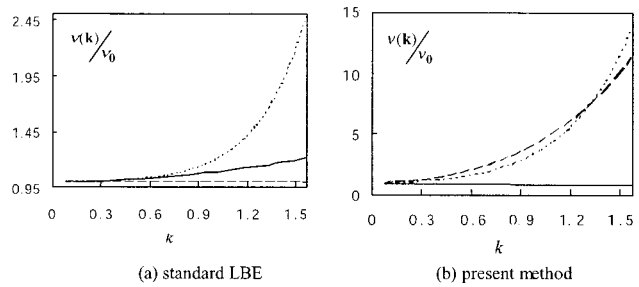
$$\overline{A}=\mathrm{diag}(1,c_1,c_2,c_3,c_4,c_5,c_6,c_7,c_8), \tag{41}$$

where

$$c_\alpha=\sum_{j=1}^{M+1} a_1^\alpha e^{i\mathbf{k}\cdot\Delta\mathbf{r}_j}, \quad \alpha=1,2,\ldots,8, \tag{42}$$

$\Delta\mathbf{r}_j=\mathbf{r}_j-\mathbf{r}_0$. In this paper, we use a uniform mesh with square grids. So, $\Delta\mathbf{r}_j=\mathbf{e}_\alpha\ \delta t=\mathbf{e}_\alpha$, where δt is taken as one. Thus, Eq. (42) can be reduced to

$$c_\alpha=\sum_{j=1}^{M+1} a_{1,j}^\alpha e^{i\mathbf{k}\cdot\mathbf{e}_\alpha}, \quad \alpha=1,2,\ldots,8. \tag{43}$$



(a) standard LBE (b) present method

FIG. 4. **k** dependence of viscosities for the standard LBE and the present method. The solid lines, dotted lines, and dashed lines correspond to $\theta=0$, $\pi/8$, and $\pi/4$, respectively.

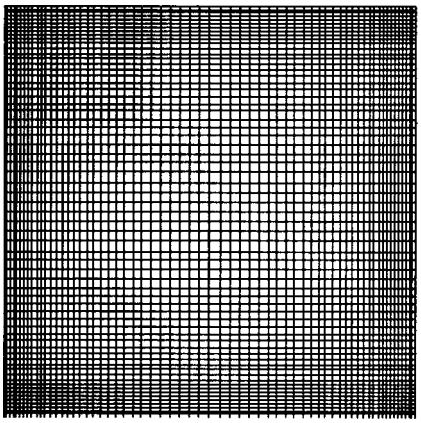
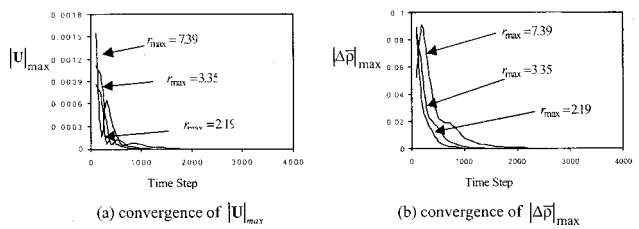
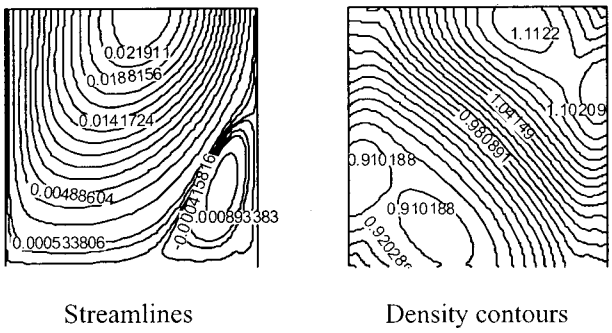


FIG. 5. A Typical nonuniform mesh in a square cavity (65 $\times$65).



(a) convergence of $|\mathbf{U}|_{max}$ (b) convergence of $|\Delta\bar{\rho}|_{max}$

FIG. 6. Convergence history of the maximum velocity magnitude and the relative difference of density for ''no flow'' simulation in a square cavity.



Streamlines Density contours

FIG. 7. Streamlines and density contours at time step=100 for ''no flow'' simulation by using a nonuniform mesh of 65$\times$65 and $r_{\max}=7.39$.

The adjustable parameters and relaxation parameters that appeared in the matrix C are taken from the work of Lallemand and Luo [33], which are

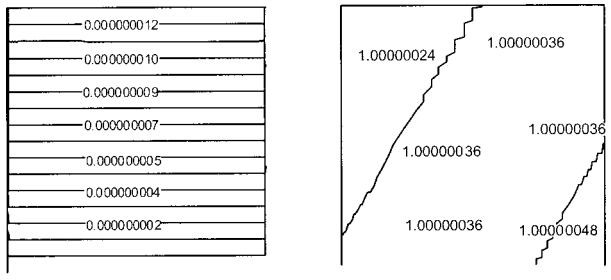
$$V_x=V_y=0,\ \alpha_2=-8,\ \alpha_3=4,\ c_1=-2,\ \gamma_1=2/3,$$

$$\gamma_2=18,\ \gamma_4=-18,$$

$$s_2=1.64,\ s_3=1.54,\ s_5=s_7=1.9,\ s_8=s_9=1.99,$$

$$\mathbf{k}=(k\cos\theta,k\sin\theta).$$

With these parameters, the eigenvalues of the operator L can be computed, and the shear viscosity can be determined from, Eq. (36). The **k** dependence of the normalized shear viscosity $\nu(\mathbf{k})/\nu_0$ for the standard LBE and the present method is shown in Fig. 4. Three orientations of **k** are chosen as: $\theta=0$ (solid line), $\pi/8$ (dotted line), and $\pi/4$ (dashed line). As shown in Fig. 4, both the standard LBE and the present method create some numerical viscosities as k increases, but the present method generates much higher numerical viscosity. For example, at $k=\pi/2$ and in the direction of $\theta=\pi/8$, the normalized shear viscosity of the standard LBE is about 2.5 while the corresponding value of the present method is about 13.5. This may imply that the present method may not be able to give accurate results at large value of k.



Streamlines Density contours

FIG. 8. Streamlines and density contours at converged state for ''no flow'' simulation by using a nonuniform mesh of 65$\times$65 and $r_{\max}=7.39$.

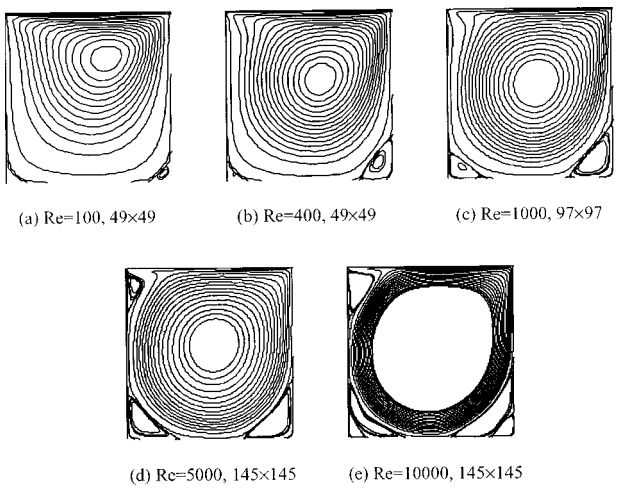


FIG. 9. Streamlines for the flow in a lid-driven square cavity. (a) Re=100, 49×49; (b) Re=400, 49×49; (c) Re=1000, 97×97; (d) Re=5000, 145×145; (e) Re=10 000, 145×145.

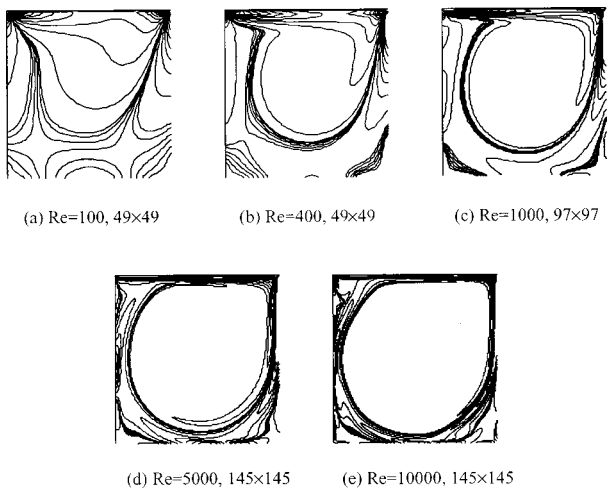


FIG. 10. Vorticity contours for the flow in a lid-driven square cavity. (a) Re=100, 49×49; (b) Re=400, 49×49; (c) Re=1000, 97×97; (d) Re=5000, 145×145; (e) Re=10 000, 145×145.

B. "No flow" simulation in a square cavity

One of the crucial measures in the application of the LBM is the so-called detailed balance condition. Violation of this condition will lead to spurious currents. The detailed balance condition can be studied by the "no flow" simulation in a region, in which no external forces are introduced, and all the boundaries are the solid walls. When a uniform density and zero velocity distribution are set up in the whole field at the beginning, the detailed balance condition demands that the uniform density and zero velocity throughout the space remain for all times; or numerically, the maximum velocity magnitude and relative difference of density are kept very small for all times.

In this paper, the "no flow" simulation in a square cavity is conducted by using three nonuniform meshes. The mesh size of the three nonuniform meshes is the same, that is 65 $\times$65. The difference is on the stretch ratio $r_{\max}$, which is defined as the ratio of the maximum mesh spacing over the minimum mesh spacing. The stretch ratios of three nonuniform meshes are, respectively, 2.19, 3.35, and 7.39. A typical mesh setup with $r_{\max}=7.39$ is shown in Fig. 5. The convergence criterion for the computation is set as

$$\frac{\|U_{t+\delta_t}-U_t\|}{\|U_{t+\delta_t}\|}\leqslant 10^{-8}. \tag{44}$$

We start our computation by assigning a uniform density ($\rho_0=1$) and zero velocity distribution in the whole domain. The viscosity is set as 1/12, and a no-slip boundary condition is imposed on the four boundaries. It was found that for all the following time steps, the maximum velocity magnitude, defined as $|U|_{\max}=\max_{i,j}(\sqrt{u^2+v^2})$, and the maximum relative difference of density, defined as $|\Delta\overline{\rho}|_{\max}=\max_{i,j}(|\rho-\rho_0|/\rho_0)$, are remained in the order of 10^{-8} and 10^{-7}, respectively, by three nonuniform meshes. This result showed that the nonuniform mesh has some effect on the detailed balance condition, but the effect is so small that it can be ignored.

We have also conducted the "no flow" simulation by giving a velocity disturbance at the beginning, and found that the flow field would turn to be static at the converged state, where the maximum velocity magnitude and the relative difference of the density are kept in the order of 10^{-8} and 10^{-7}, respectively. By giving a velocity disturbance of $u=0.1\sin(\pi x)\sin(\pi y)$ and $v=0.1\sin(\pi x)\sin(\pi y)$ at the beginning, the convergence history of $|U|_{\max}$ and $|\Delta\overline{\rho}|_{\max}$ for three nonuniform meshes is shown in Fig. 6. It can be seen from Fig. 6 that all three nonuniform meshes can converge to the static solution, but the time steps required for convergence is different. The required time steps for convergence are, respectively, 3400 for $r_{\max}=2.19$, 3500 for $r_{\max}=3.35$, and 4000 for $r_{\max}=7.39$. The need of more time steps for a converged solution by a nonuniform mesh with larger stretch ratio is due to the use of smaller time step size. Figure 7

TABLE I. Comparison for locations of primary vortex center at different Reynolds numbers.

	Vortex center		
Reynolds number	Ghia *et al.* [34]	Present method	CPU sec by present method
100	(0.61, 0.73)	(0.61, 0.73)	195.521
400	(0.56, 0.61)	(0.56, 0.60)	600.3833
1000	(0.54, 0.56)	(0.54, 0.56)	3567.650
5000	(0.52, 0.54)	(0.53, 0.56)	20443.85
10000	(0.51, 0.51)	(0.51, 0.52)	64401.57

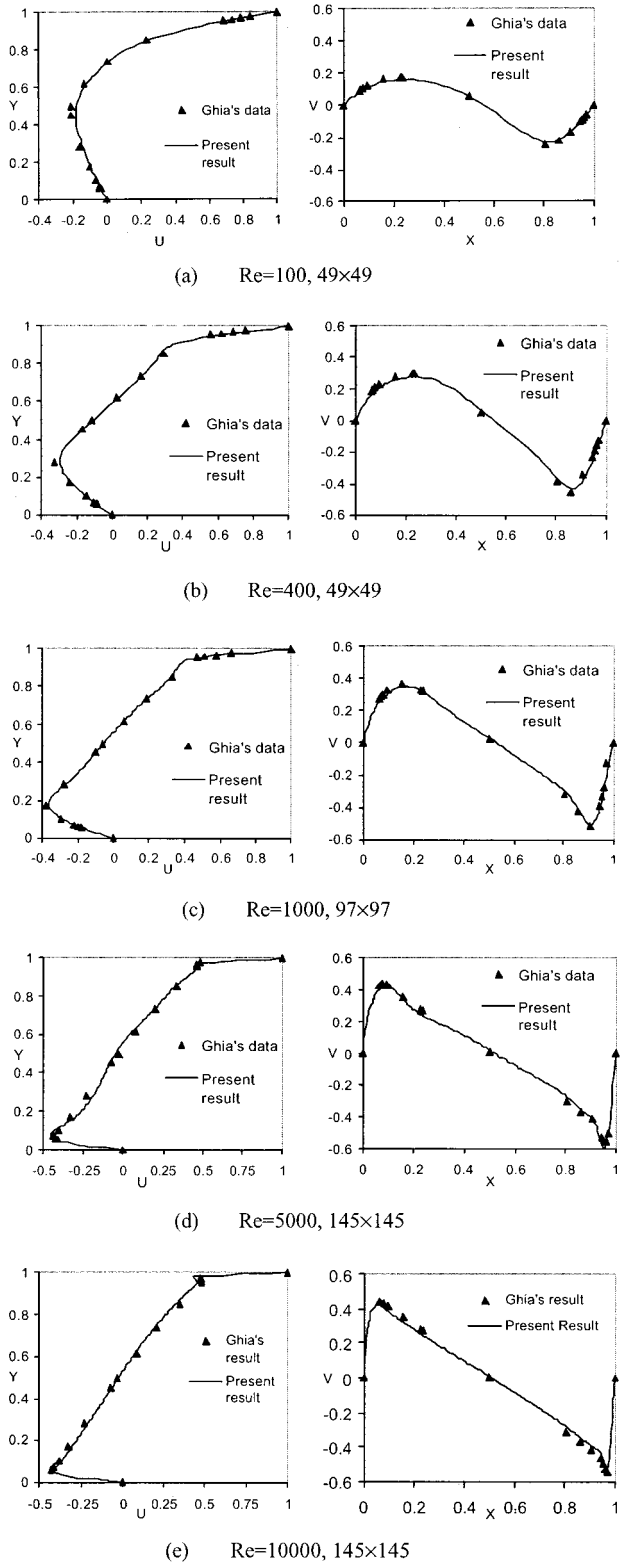
shows the streamlines and density contours obtained by using a nonuniform mesh of $r_{max}=7.39$ after 100 time steps. The effect of initial disturbance on the streamline and density is clearly observed in the figure. However, when the computation is converged, the effect of initial disturbance is removed, and a static result is obtained. This can be seen clearly in Fig. 8, which shows the streamlines and density contours by using the same nonuniform mesh at the converged state.

C. Lid-driven flows in a square cavity

In this case, the top lid moves from left to right with a constant velocity U. Nonuniform meshes of 49×49 for Re $=100$ and 400, 97×97 for Re$=1000$, 145×145 for Re $=5000$ and 10 000 are used, respectively, for the calculation. The Reynolds number is defined as $\mathrm{Re}=UL/\nu$ (based on the lid velocity and the length of the square cavity). The use of nonuniform mesh is desirable, especially for the high Reynolds number case. This is because the thin boundary layer is attached to the solid boundaries. So, to capture the thin boundary layer, the mesh spacing near the wall should be very small. Apart from the solid wall, relatively large mesh spacing can be used. In this manner, we can capture the thin boundary layer, and in the meantime, we can save the computational effort.

Initially a constant density $\rho=1$ is prescribed in the whole field, and the velocities in the interior of the cavity are set to zero. On the top, the x-component velocity is U, which is set to 0.15, and the y-component velocity is zero. At the end of each time step, the density distribution function f_α at the top is set to the equilibrium state. The whole halfway wall bounce-back boundary conditions are used on the other three solid walls. For the upper two corners between the stationary wall and the moving wall, which are singular points, it is found that treatment with the moving wall or the stationary wall points has little difference in our simulations. In order to make a comparison with the results of Ghia, Ghia, and Keller [34], the present results are normalized according to the length of the cavity and the velocity of the top lid. Note that the results of Ghia, Ghia, and Keller [34] were obtained by using the multigrid finite difference method to solve the vorticity-stream function formulation.

Figure 9 shows the streamlines for different Reynolds numbers. The effect of the Reynolds number on the flow pattern and the structure of the steady recirculating eddies is clearly observed in this figure. Apparently, the flow structures match very well with those of Ghia, Ghia, and Keller [34]. Figure 10 shows the vorticity contours inside the cavity. From this figure, one can easily see that the scale of the primary vortex increases with the Reynolds number, and the magnitude of the vorticity in the central region becomes larger and larger with the increase of the Reynolds number. Table I gives the detailed comparison for locations of the vortex center obtained by the present method and by Ghia, Ghia, and Keller [34]. The relative errors between the two solutions are less than 4%. In this table, we also give the CPU time (sec) that is spent in the present computation on the PIII 866 personal computer. Although other information about this aspect is lacking, we still believe, from our numerical experiences, that the computational efficiency of the present method is competitive as compared with the traditional CFD tools. The U and V velocities along their respec-



(a) Re=100, 49×49

(b) Re=400, 49×49

(c) Re=1000, 97×97

(d) Re=5000, 145×145

(e) Re=10000, 145×145

FIG. 11. U (left) and V (right) velocity profiles along vertical and horizontal central lines for different Reynolds numbers.

TABLE II. Grid sensitivity with different stretch ratios (Re=400).

Method	Mesh size (stretch ratio)	Primary vortex center	$\psi_{\min}$	$\omega_{v.c.}$	CPU (sec)
Ghia *et al.* [34]	257×257 (1.0)	(0.5547, 0.6055)	−0.113 909	2.294 69	
D2Q9 (Standard LBE)	193×193 (1.0)	(0.5573, 0.6042)	−0.113 539	2.288 82	7688.94
	97×97 (1.0)	(0.5625, 0.6042)	−0.107 039	2.187 27	1081.55
D2Q9 (Present)	97×97 (3.35)	(0.5570, 0.6061)	0.113 642	2.291 51	5696.45
	49×49 (2.01)	(0.5611, 0.6004)	−0.110 376	2.367 18	463.01
D2Q7 (Present)	49×49 (3.35)	(0.5597, 0.6050)	−0.112 574	2.304 63	600.38
	97×97 (3.35)	(0.5570, 0.6061)	−0.113 629	2.291 50	4564.69

tive central line are displayed in Fig. 11 for different Reynolds numbers. Obviously, our simulation results are in good agreement with those of Ghia, Ghia, and Keller [34].

To test the mesh sensitivity, numerical simulations for Re=400 on mesh sizes of 49×49 and 97×97 with different stretch ratios were carried out. The results of the D2Q9 model using the present method on the nonuniform mesh of 97×97 ($r_{\max}$=3.35) and the standard LBE on the uniform mesh of 97×97 are also presented for comparison. The detailed comparison of the primary vortex center, the minimum stream function value $\Psi_{\min}$ and the vorticity value at the primary vortex center $\omega_{v.c.}$, and the CPU time required on PIII 866 are listed in Table II. As shown in this table, the accuracy of the present method increases with the increase of the grid stretch ratio. At the same level of the grid stretch ratio, the accuracy of the present method is improved with the increase of the mesh points. It is also seen from this table that our method needs about five times the computational time of the standard LBE when the same number of mesh points is used. However, this undesirable feature of the present method can be compensated for by using fewer mesh points that, for the standard LBE, cannot give satisfactory accuracy. As shown in Table II, for Re=400, the standard LBE needs at least 193×193 mesh points to reach the same order of accuracy as the present method using the mesh size of 97×97. As a result, the standard LBE requires 7688.94 sec CPU time while the present method only needs 5696.45 sec CPU time. The high efficiency of the present method is obvious. In addition, Table II reveals that the present method with the D2Q7 model saves about 20% of computational time as compared with the D2Q9 model when the same mesh points and distributions are used.

D. Flow in a polar cavity

The polar cavity case is used to show the capability of the present method in treating the flow problem with complex geometry. The geometry with a nonuniform mesh is given in Fig. 12. The Reynolds number (based on the lid velocity and the radius of the inner circle) is 350. Initially, a clockwise velocity of $U=0.15$ is set on the inner lid and other conditions are the same as those in the square cavity case. All results are normalized in terms of the lid velocity and the radius of the inner circle.

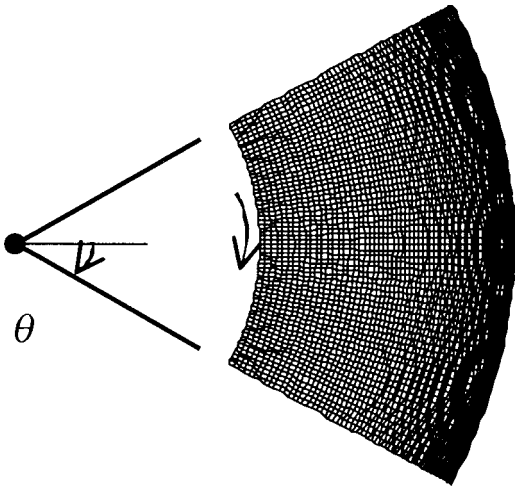


FIG. 12. Geometry and a typical nonuniform mesh for the flow in a polar lid-driven cavity.

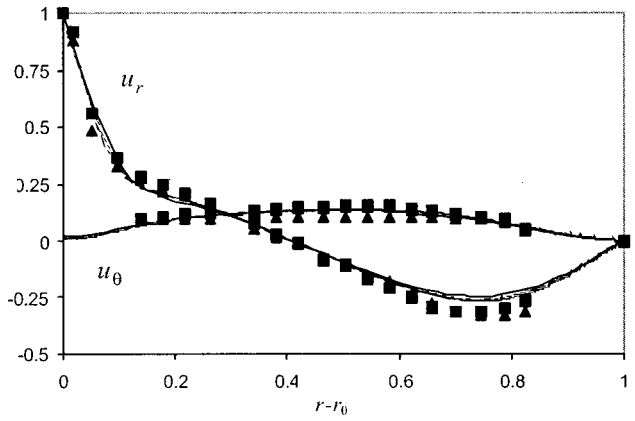


FIG. 13. Comparison of radial (u_r) and azimuthal (u_θ) velocity profiles along the line of $\theta=0°$ with Re=350 (■, numerical data by Fuchs and Tillmark; ▲, experimental data by Fuchs and Tillmark; –, present result of 49×49; − − −, present result of 65×65; ———, present result of 81×81).

Figure 13 shows the steady-state azimuthal and radial velocity profiles along the line of $\Theta = 0$. Different results, which are obtained on 49×49, 65×65, and 81×81 nonuniform meshes, together with the experimental and numerical results of Fuchs and Tillmark [35] are included in the figure for comparison. The results obtained by the present method agree well with those of Fuchs and Tillmark's numerical simulation [35]. The present solutions also compare well with the experimental data and the discrepancy between them may be attributed to the three-dimensional effect in the experiments in which three-dimensional flow structures are observed. Further increase of the mesh points can lead to more accurate results by the present method.

VI. CONCLUSIONS

An explicit Taylor-series expansion- and least-squares-based lattice Boltzmann method was developed in this paper. Theoretical analysis for a one-dimensional case showed that the present method could recover the Navier-Stokes equations with second order of accuracy. The successful numerical simulations for the lid-driven cavity flows showed that the present method is prospective in practical applications.

The beauty of the present method is that it still keeps the local and explicit features of the standard lattice Boltzmann method. Therefore, it is able to exploit fully the power of parallel computing. The other advantage of the present method is that it is easy for application to flow problems with complex geometry. Furthermore, the computational efficiency of the present method is competitive as compared with that of the standard LBE and of the conventional CFD solvers.

Although the test problems shown in this paper are relatively simple, the proposed method can be applied to any other complex problems with the use of nonuniform mesh and different lattice models. The present paper shows that the proposed method gives much larger shear viscosity at large value of wave number k. This result is based on a uniform mesh and the use of symmetric mesh point distribution used in Eq. (18). Further study may be needed for the k dependence of shear viscosity at large value of k when the nonuniform mesh and upwind mesh point distribution are used in Eq. (18).

[1] S. Succi, E. Foti, and F. Higuera, Europhys. Lett. **10**, 433 (1989).
[2] S. Chen, Z. Wang, X. Shan, and G. D. Doolen, J. Stat. Phys. **68**, 379 (1992).
[3] F. J. Alexander, S. Chen, and D. W. Grunau, Phys. Rev. B **48**, 634 (1993).
[4] F. J. Alexander, S. Chen, and J. D. Sterling, Phys. Rev. E **47**, 2249 (1993).
[5] D. O. Martinez, W. H. Matthaeus, S. Chen, and D. C. Montgomery, Phys. Fluids **6**, 1285 (1994).
[6] C. K. Aidun and Y.-N. Lu, J. Stat. Phys. **81**, 49 (1995).
[7] L.-S. Luo, Int. J. Mod. Phys. C **8**, 859 (1997).
[8] J. Huang, Ph.D. thesis, Drexel University, 1998.
[9] U. Frisch, B. Hasslacher, and Y. Pomeau, Phys. Rev. Lett. **56**, 1505 (1986).
[10] S. Wolfram, J. Stat. Phys. **45**, 471 (1986).
[11] J. M. V. A. Koelman, Europhys. Lett. **15**, 603 (1991).
[12] Y. H. Qian, D. d'Humieres, and P. Lallemand, Europhys. Lett. **17**, 479 (1992).
[13] P. L. Bhatnagar, E. P. Gross, and M. Krook, Phys. Rev. **94**, 511 (1954).
[14] X. He, L.-S. Luo, and M. Dembo, J. Comp. Physiol. **129**, 357 (1996).
[15] X. He and G. D. Doolen, J. Comp. Physiol. **134**, 306 (1997).
[16] X. He and G. D. Doolen, Phys. Rev. E **56**, 434 (1997).
[17] R. Mei and W. Shyy, J. Comp. Physiol. **134**, 306 (1997).
[18] H. Chen, Phys. Rev. E **58**, 3955 (1998).
[19] F. Nanelli and S. Succi, J. Stat. Phys. **68**, 401 (1992).
[20] G. Peng, H. Xi, and S. H. Chou, Int. J. Mod. Phys. C **10**, 1003 (1999).
[21] G. Peng, H. Xi, C. Duncan, and S. H. Chou, Phys. Rev. E **59**, 4675 (1999).
[22] L. F. Shampine, *Numerical Solution of Ordinary Differential Equations* (Chapman and Hall, London, 1994).
[23] L. James, L. Buchanan, and Peter R. Turner, *Numerical Methods and Analysis* (McGraw-Hill, New York, 1992).
[24] H. Chen, S. Chen, and W. H. Matthaeus, Phys. Rev. A **45**, 5339 (1991).
[25] P. A. Skordos, Phys. Rev. E **48**, 4823 (1993).
[26] D. P. Ziegler, J. Stat. Phys. **71**, 1171 (1993).
[27] D. R. Noble, S. Chen, J. G. Georgiadis, and R. O. Buckius, Phys. Fluids **7**, 203 (1995).
[28] T. Inamuro, M. Yoshino, and F. Ogino, Phys. Fluids **7**, 2928 (1995).
[29] S. Chen, D. Martinez, and R. Mei, Phys. Fluids **8**, 2527 (1996).
[30] R. Cornubert, D. d'Humieres, and D. Levermore, Physica D **47**, 241 (1991).
[31] I. Ginzbourg and D. d'Humieres, J. Stat. Phys. **84**, 927 (1996).
[32] Dieter A. Wolf-Gladrow, *Lecture Notes in Mathematics* (Springer, New York, 2000).
[33] P. Lallemand and L.-S. Luo, Phys. Rev. E **61**, 6546 (2000).
[34] U. Ghia, K. N. Ghia, and H. B. Keller, J. Comp. Physiol. **48**, 387 (1982).
[35] L. Fuchs and N. Tillmark, Int. J. Numer. Methods Fluids **5**, 311 (1985).