Finite volume scheme for the lattice Boltzmann method on unstructured meshes

Gongwen Peng, Haowen Xi, and Comer Duncan
Department of Physics and Astronomy, Bowling Green State University, Bowling Green, Ohio 43403

So-Hsiang Chou
Department of Mathematics and Statistics, Bowling Green State University, Bowling Green, Ohio 43403
(Received 9 September 1998)

A finite volume scheme for the lattice Boltzmann method is developed for unstructured triangular meshes in two dimensions. The accuracy of this new scheme is demonstrated by comparing the numerical results with the exact solutions to the Navier-Stokes equations for Taylor vortex flow, shear flow between two parallel plates, shear flow between two rotating cylinders, and Poiseuille flow. The agreement between the numerical and analytical results is very good for each of these tests. [S1063-651X(99)07804-6]

PACS number(s): 47.10.+g, 47.11.+j, 05.20.Dd

I. INTRODUCTION

In recent years the lattice Boltzmann method (LBM) has attracted much interest in the physics and engineering communities. As a different approach from the conventional computational fluid dynamics (CFD), the LBM has been demonstrated to be successful in simulations of fluid flow and other types of complex physical systems [1–4]. In particular, this method is promising for simulations of multiphase and multicomponent fluid flow involving complex interfacial dynamics [5–9]. Unlike the conventional CFD that directly simulates evolution of the macroscopic Navier-Stokes equations, the LBM is based on the mesoscopic kinetic equation for the single particle distribution function. It has been proven that the Navier-Stokes equations can be recovered from the LBM at the macroscopic level [1,2]. The obvious advantages of using LBM are the simplicity of programming, the parallelism of the algorithm, and the capability of incorporating complex microscopic interactions.

Historically, the LBM was developed from the lattice gas automaton (LGA) [10] model. In an LGA model, the dynamics of particles consists of two steps: (1) particles at the same site collide according to a set of hard-sphere particle collision rules that conserve mass, momentum, and energy (for multispeed models) at each lattice site; (2) after collision, particles advance to the next lattice site in the direction of their velocities. The small number of discrete velocities allowed in the LGA models is tightly coupled with the spatial lattice structure. At its earliest development stage, the LBM was a floating-point version of the LGA model where the particle distribution function in the LBM was interpreted as the floating-number counterpart of the Boolean particle occupation in the LGA. Two important improvements to enhance the computational efficiency were made later. They were the linearization of the collision operator [11] and the adoption of the single time relaxation approximation [12] (or the BGK approximation [13]). Nevertheless, the requirement of using uniform spatial lattice structure was still unchanged until very recently.

In the commonly used LBM models, as in their LGA precursors, the discrete velocity directions are associated with the structure of the underlining spatial lattice. For example, in a two-dimensional square lattice one uses nine velocities—eight velocities pointing to the nearest neighbors plus one rest velocity; in a two-dimensional triangular lattice seven velocities are used—six directions for the velocities point to the nearest neighbors along with a rest velocity. This arrangement of the discrete velocities prescribes that particle density distributions move exactly from one lattice point to an adjacent point.

To remove this constraint, during the past few years several workers have extended the LBM models to use irregular lattices. Succi and his collaborators [14] were the first to propose a finite-volume formulation of the lattice Boltzmann equation (LBE) using the idea of a finite-volume method. They begin from the differential form of the LBE and apply Gauss' theorem to a set of macrocells covering the spatial domain. For each cell, a volume-averaged ''coarse-grain'' particle distribution is defined. By using either piecewise linear or piecewise constant interpolation procedures they obtain equations of the ''coarse-grain'' distribution. He, Luo, and Dembo [15] have proposed a model for an arbitrary but logically rectangular mesh. In this model, collisions still take place on the grid nodes as in the ordinary LBM models. After a collision, the density distributions move along their respective velocities to points which in general will not be exactly on the grid nodes. An interpolation step is thus introduced in this model to determine the density distributions at the grid nodes for the next time step and the above procedures are repeated.

In the above-mentioned approaches of using nonuniform lattices, the lattice connectivity is still restricted to be structured. For example, in the two-dimensional simulations of previous workers, logically rectangular mesh structures were mostly used [14,15] in association with nine discrete velocities, though the meshes were not the regular square lattice. This is in contrast to the situation in the modern CFD techniques which are generally capable of accommodating fairly complex meshes. In this paper we describe a computational scheme based on two-dimensional unstructured meshes from the point of view of finite volume methods.

This paper is organized as follows. In Sec. II we specify the finite-volume LBM scheme (a brief description of this model was reported in Ref. [16]). As examples of tests of this new scheme, numerical simulations for two-dimensional Taylor vortex flow, shear flows between two parallel plates

and two rotating cylinders as well as Poiseuille flow will be presented in Sec. III. Section IV contains some comparisons with other methods and concluding remarks.

II. FINITE VOLUME LBM MODEL

Our starting point is the LBE. Recently, it is shown [17] that the LBE can be directly derived from the Boltzmann equation by discretization in phase space without borrowing the concept of particles jumping from site to site as in the LGA model. The commonly used LBM models can be regarded as specific discretizations of the LBE on regular lattices. More general finite difference discretizations of the LBE were studied in Ref. [18] and have been extended to efficient parallel schemes [19]. The flexibility gained in unlocking the spatial and velocity lattices from each other provides us with an important degree of freedom in designing our finite-volume scheme.

The LBE reads as follows after discretizing the velocity space:

$$\frac{\partial f_i}{\partial t}+\mathbf{v}_i\cdot\nabla f_i=\Omega_i, \tag{1}$$

where f_i is the particle distribution function associated with motion along the ith direction in velocity space, $\mathbf{v}_i$ the velocity in the ith direction, $i=1,2,\ldots,m$ with m the number of different velocities in the model, and Ω_i is the collision operator. One commonly used model is the lattice BGK model [13], which uses the single time relaxation approximation, i.e.,

$$\Omega_i=-\frac{1}{\tau}(f_i-f_i^{eq}), \tag{2}$$

where f_i^{eq} is the local equilibrium distribution and τ is the relaxation time. f_i^{eq} is carefully constructed so that the velocity moments of the LBE reproduce the Navier-Stokes equations.

It should be pointed out that in the phase space the space variable $\mathbf{x}$ and the velocity variable $\mathbf{v}$ are independent. In the LBM, only a small set of discrete velocities are used to approximate the Boltzmann kinetics of the continuum velocity. In the original formulations of the LBM, it was understood that the discretization of momentum space is coupled with that of real space. But as emphasized in Refs. [20,17,18], this coupling is not necessary and both discretizations can be done independently. Here we will completely decouple these two discretization procedures by choosing the nine velocities as in the nine-bit model [12] for the velocity discretization while using arbitrary triangular meshes for the spatial discretization.

The nine discrete velocities are defined by

$$\mathbf{v}_i=\begin{cases}(0,0), & i=0,\\ (\cos[(i-1)\pi/2],\sin[(i-1)\pi/2]), & i=1,2,3,4,\\ \sqrt{2}(\cos[(i-5)\pi/2+\pi/4],\sin[(i-5)\pi/2+\pi/4]), & i=5,6,7,8\end{cases} \tag{3}$$

and the equilibrium distribution f_i^{eq} is given by [12]

$$f_i^{eq}=w_i\rho[1+\tfrac{3}{2}(\mathbf{v}_i\cdot\mathbf{u})+\tfrac{9}{2}(\mathbf{v}_i\cdot\mathbf{u})^2-\tfrac{3}{2}|\mathbf{u}|^2], \tag{4}$$

where

$$\rho=\sum_i f_i, \tag{5}$$

$$\mathbf{u}=\sum_i f_i\mathbf{v}_i/\rho \tag{6}$$

are the density and velocity, respectively, and

$$w_i=\begin{cases}4/9, & i=0,\\ 1/9, & i=1,2,3,4,\\ 1/36, & i=5,6,7,8.\end{cases} \tag{7}$$

We choose two-dimensional triangular meshes to illustrate how the finite-volume scheme is constructed. Figure 1(a) shows a generic situation in which triangular elements surround an interior grid node. The scheme we report here is a finite-volume method of the cell-vertex type [21]. In this type of formulation, the f_is at the nodes are the unknowns. When we need to calculate f_is at non-node positions, these values would be interpolated from the f_is at the nodes using standard interpolation procedures [22] according to the element types in use. For example, linear and bilinear interpolations could be applied to the triangular and quadrilateral elements, respectively.

We choose the control volume as the polygon around the node P as shown in Fig. 1(a), known as the dual mesh. Two sides of the polygon, CE and ED, are labeled in the figure. Here E is the middle point of edge PP_2 and C is the geometric center of element PP_1P_2 with coordinates

$$\mathbf{x}_E=(\mathbf{x}_P+\mathbf{x}_{P_2})/2,\quad \mathbf{x}_C=(\mathbf{x}_P+\mathbf{x}_{P_1}+\mathbf{x}_{P_2})/3. \tag{8}$$

Likewise, D is the center of element PP_2P_3. The integration volume consists of triangles PCE, PED, etc., taken in counterclockwise order. In the following we focus on integration over the triangle PCE. Similar integrations would be done over all such triangles centered on P and the results summed.

The integration of the first term in Eq. (1) is approximated as

$$\int_{PCE}\frac{\partial f_i}{\partial t}d\sigma=\frac{\partial f_i(P)}{\partial t}A_{PCE}, \tag{9}$$

where A_{PCE} is the area of triangle PCE and $f_i(P)$ is the f_i value at node P. From now on, the node index is indicated in

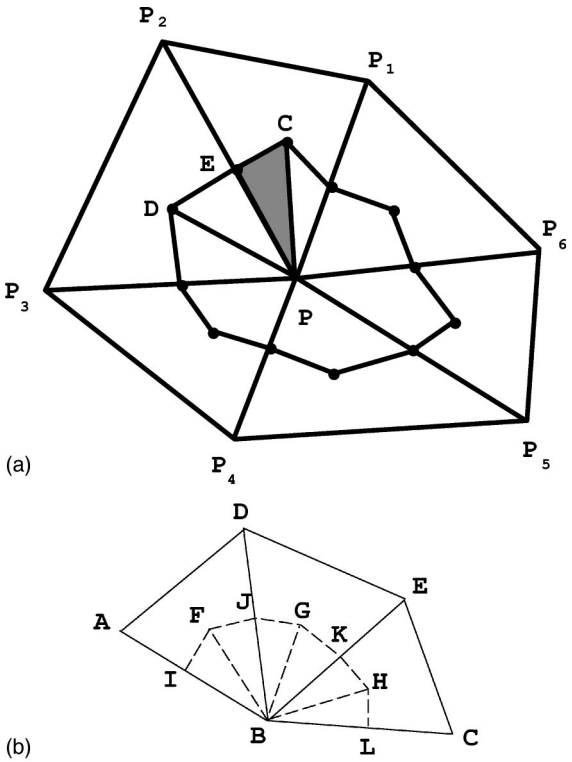

FIG. 1. (a) Diagram of finite elements sharing one common node. Here $P,P_1,P_2,\ldots,P_6$ stand for the mesh grid points. CE and ED are two boundary edges of the control volume (polygon) over which integration of the PDE of Eq. (1) is performed. (b) Diagram illustrating the half-covolume boundary condition method. A,B,C are three nodes at walls, D and E are two fluid interior nodes linked with wall node B. F, G, and H are the geometric center of triangles BDA, BED, and BCE, respectively, while I, J, K, and L are the center points of the corresponding edges. The covolume consists of the dashed triangles.

parentheses following the f values. In the above equation, we have made an approximation that f_i is constant over the triangle PCE to prevent us from solving a large set of equations if f_is were assumed to be linear. This kind of ''lumping'' is commonly used in the finite volume method applications [21].

Integration of the convection term of Eq. (1) gives fluxes though the three edges PC, CE, and EP:

$$\int_{PCE}\mathbf{v}_i\cdot\nabla f_i d\sigma=\mathbf{v}_i\cdot\int_{CE}f_i d\mathbf{l}+I_s \tag{10}$$

where $d\mathbf{l}$ is normal to the integration edge CE, and I_s is the fluxes through internal edges (e.g., PC, PE). Since we will sum over all the triangles like PCE, PED, the net flux through internal edges (e.g., PC, PE) will cancel out and not enter into the calculation. On the assumption that each f_i varies linearly over each triangular element, the right hand side of Eq. (10) becomes

$$\int_{PCE}\mathbf{v}_i\cdot\nabla f_i d\sigma=\mathbf{v}_i\cdot\mathbf{n}_{CE}l_{CE}[f_i(C)+f_i(E)]/2+I_s \tag{11}$$

where $\mathbf{n}_{CE}$ is the unit vector normal to the edge CE, and l_{CE} is the length of CE.

Since we use triangular elements, $f_i(C)$ and $f_i(E)$ must be interpolated from the values at the three nodes of the element PP_1P_2. The values at C and E are then linearly interpolated as

$$f_i(C)=[f_i(P)+f_i(P_1)+f_i(P_2)]/3, \tag{12}$$

$$f_i(E)=[f_i(P)+f_i(P_2)]/2. \tag{13}$$

For other types of elements, the above interpolation would be replaced with an appropriate interpolation. For example, one could use bilinear interpolation for a discretization into quadrilateral elements.

The integration over the third term of Eq. (1) [i.e., Eq. (2)] results in the following formula, assuming the linearity of f_i and f_i^{eq} over the triangular element PCE,

$$-\int_{PCE}\frac{1}{\tau}(f_i-f_i^{eq})d\sigma=-\frac{A_{PCE}}{\tau}\{[f_i(P)-f_i^{eq}(P)]+[f_i(C)-f_i^{eq}(C)]+[f_i(E)-f_i^{eq}(E)]\}/3, \tag{14}$$

where $f_i^{eq}(C)$ and $f_i^{eq}(E)$ are interpolated from the equilibrium values at the three nodes at element PP_1P_2 as follows:

$$f_i^{eq}(C)=[f_i^{eq}(P)+f_i^{eq}(P_1)+f_i^{eq}(P_2)]/3, \tag{15}$$

$$f_i^{eq}(E)=[f_i^{eq}(P)+f_i^{eq}(P_2)]/2. \tag{16}$$

With these, the integration of Eq. (1) over the triangle PCE is complete. The integration over the whole control volume is just the sum of all these terms over different triangles such as PCE, PED, etc. Therefore, a first-order accuracy time-stepping scheme for the update of f_i at node P is given as follows:

$$f_i(P,t+dt)=f_i(P,t)+\frac{dt}{A_P}\left(\sum\text{ collisions}-\sum\text{ fluxes}\right) \tag{17}$$

where A_P is the total area of the control volume around node P. The terms ''collisions'' and ''fluxes'' refer respectively to the finite-volume-integrated contributions from the collision term and the flux term. The summation is over different triangles PCE, PED, etc., associated with node P.

In all the computations we present in this paper, the update of the f_is at boundary nodes is similar to that for interior nodes except at the boundary the corresponding covolumes are half-covolumes. Let A,B,C in Fig. 1(b) boundary nodes and D, E interior fluid nodes linked with node B. We focus on the update of f_is at node B. As for interior fluid nodes, we update f_is at B by covolume integrals. However, the covolume is now not complete in the 2π directions, as shown in Fig. 1(b) where F, G, and H are the geometric center of triangles BDA, BED, and BCE, respectively, I, J, K, and L are the center points of the corresponding edges [see Fig. 1(b) and compare with Fig. 1(a)]. As for interior nodes, integrals of Eq. (1) over the covolume are carried out over the smaller triangles BLH, $BHK\ldots$ and BFI one by one. There is only one difference, which occurs when integrating the second term of Eq. (1) over triangles BLH and BFI. The flux terms over edges BL and BI, which we omitted in the

case of interior nodes [i.e., I_s in Eq. (11)] as they were internal fluxes, must now be included in the calculation. They are actually easy to evaluate as shown in Eq. (11) for fluxes over edges.

III. NUMERICAL TESTS

We have conducted simulations using the finite-volume scheme for two-dimensional Taylor vortex flow, shear flows between two parallel plates and between two rotating cylinders, and Poiseuille flow.

A. Two-dimensional Taylor vortex flow

As a first example, we simulate the evolution of the two-dimensional Taylor vortex flow in a square domain with periodic boundary conditions in both x and y directions. The system has an initial state with velocity at position (x,y) given by $u_x(x,y,0) = -u_0 \cos(k_1 x)\sin(k_2 y)$ and $u_y(x,y,0) = u_0(k_1/k_2)\sin(k_1 x)\cos(k_2 y)$, where k_1 and k_2 are given by $k_1 = 2\pi m/L$ and $k_2 = 2\pi n/W$. Here L and W are the length and width of the system and m and n can be any integers. There is no driving force presented in the system and the velocities will decay their magnitudes as a function of time due to the viscous nature of the fluid. The velocity evolution is characterized by

$$u_x(x,y,t) = -u_0 \exp[-\nu t(k_1^2+k_2^2)] \cos(k_1 x)\sin(k_2 y), \tag{18}$$

$$u_y(x,y,t) = u_0(k_1/k_2)\exp[-\nu t(k_1^2+k_2^2)] \sin(k_1 x)\cos(k_2 y) \tag{19}$$

according to the Navier-Stokes equation. Here ν is the kinematic viscosity of the fluid.

Since the present model is expected to yield a faithful solution to the continuous LBE, we expect it will give the same viscosity as the LBE. From Eq. (18) one can see that the decaying flow can be used to numerically determine the kinematic viscosity. We have simulated the flow using three different types of meshes, depicted in Figs. 2(a)–2(c). We focus on the velocity change at one chosen node and plot its velocity decaying in the course of time in Fig. 3(a). In this example, the mesh used is of the type of Fig. 2(b). The system has a size of $L=32$ and $W=128$ and the mesh size is 1.0. Here $\rho=1.0$, $\tau=1.0$ and $dt=0.25$. We use $k_1 = (2\pi/L)$, $k_2 = 4(2\pi/W)$ and $u_0 = 0.01$. The linearity in the semilogarithmic plot shows clearly that the velocity decays exponentially and from the slope of the straight line in Fig. 3(a) one can obtain the viscosity. Here we have chosen a node whose coordinates are $x=L/4$ and $y=W/2$. We find that the kinematic viscosity equal to $\tau/3$ for all the meshes we have used, independent of the mesh sizes and mesh types. One should note that this result is nontrivial since other finite-volume schemes may introduce some numerical viscosity. Figure 3(b) shows the velocity $u_y(x)$ at time $t=50$ for all the nodes with coordinate $y=W/2$ in a simulation with $\rho=1.0$, $\tau=0.1$, and $dt=0.01$. Here we take the same system sizes and K_1, K_2 as in Fig. 3(a). The solid line in Fig. 3(b) is the analytical solution [Eq. (18)] with $\nu=\tau/3$, showing excellent agreement between the numerical and analytical results.

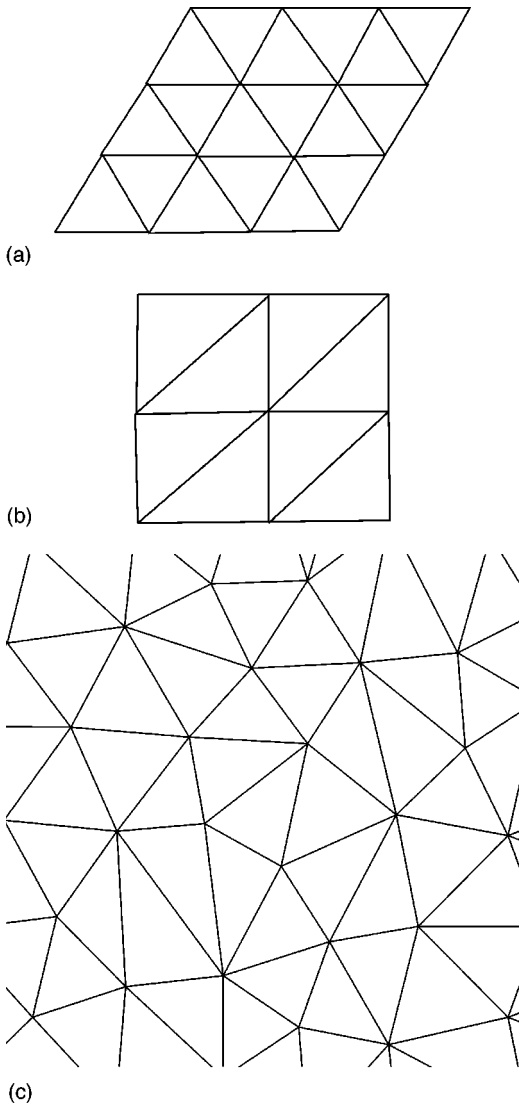

FIG. 2. Three types of meshes (a) triangular mesh, (b) Cartesian mesh, (c) irregular mesh. Both (a) and (b) are structured meshes while (c) is unstructured.

B. Shear flows

Shear flow between two parallel plates is simulated using the finite volume scheme. The system is a two-dimension square domain with periodic boundary condition in the x direction. Shear velocity along the x direction is applied at the two boundaries located at $y=0$ and $y=W$. We start from an initial state where the velocity is zero everywhere in the system. The system is then exposed the external shear and the velocity field is calculated. Figure 4 shows the x component of velocity after the system reaches the steady state. We plot three sets of data corresponding to three types of meshes shown in Figs. 2(a)–2(c) (they are collapsing on top of each other). Here we perform the simulations with a system of width $W=31$ and length $L=31$. The mesh size is in the order of 1.0 for all these meshes. The shear velocities are taken to be 0.05 at $y=W$ and -0.05 at $y=0$. We use $\tau=0.5$ and $dt=0.01$, and find that the steady state is reached after $T = 50\,000$ time steps in the three cases. From Fig. 4 it is clear

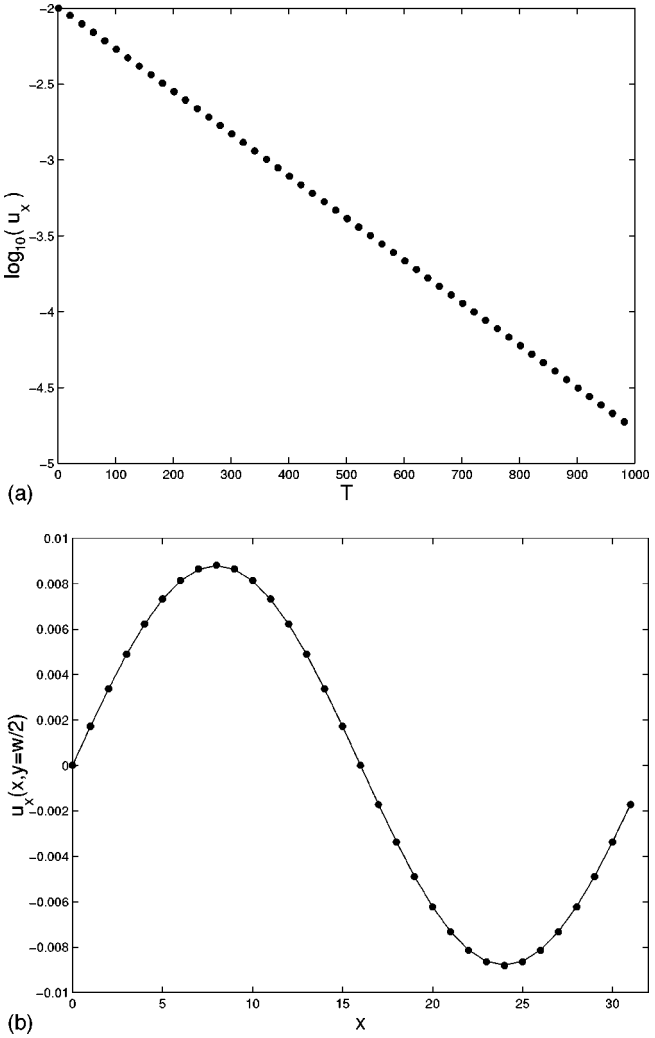

FIG. 3. (a) Semilogarithmic plot of the velocity u_x $[\log_{10}(u_x)]$ decaying with time for one chosen node at $x=L/4$ and $y=W/2$. The mesh used is of type Fig. 2(b). The system has a size of $L=32$ and $W=128$ and the mesh size is 1.0. Here $\rho=1.0$, $\tau=1.0$, $dt=0.25$, $k_1=2\pi/L$, $k_2=8\pi/W$ and $u_0=0.01$. (b) $u_x(x,y=W/2)$ at $t=50$, compared with exact solution Eq. (18). Here $\rho=1.0$, $\tau=0.1$, $dt=0.01$, $k_1=2\pi/L$, $k_2=8\pi/W$ with $L=32$ and $W=128$. The mesh is of type Fig. 2(b) with grid size of 1.0. $\nu=\tau/3$ was used in the solid line.

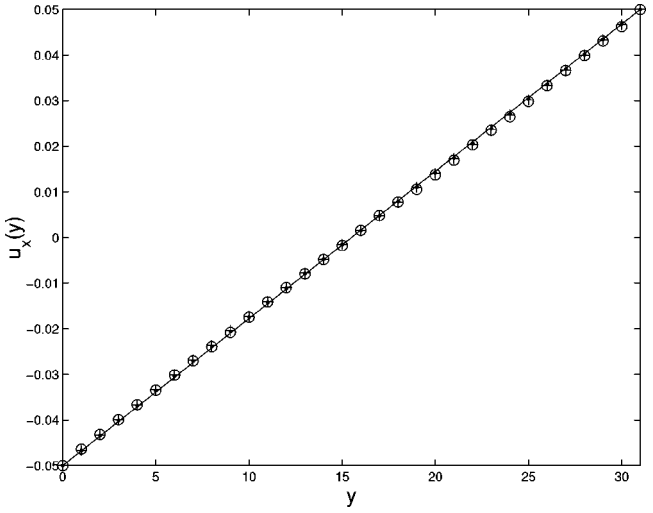

FIG. 4. $u_x(y)$ in the steady state for shear flow between two parallel plates located at $y=0$ and $y=W=31$ with shear velocities $u_x(0)=-0.05$ and $u_x(W)=0.05$. Three sets of data (dot, plus, and circle) are collapsing on top of each other and they correspond to the three types of meshes depicted in Figs. 2(a)–2(c), respectively. The straight line is the analytical solution to the Navier-Stokes equation. For irregular mesh, each point is an average of velocity at nodes whose y-coordinates are between $y-1/2$ and $y+1/2$ with y integers.

that the computed velocities agree very well with the linear velocity profile of the exact solution to the Navier-Stokes equation.

To demonstrate the flexibility of the finite volume scheme we present another example of simulation of shear flow between two coaxial cylinders that also possesses an exact solution for comparison. Figure 5 illustrates an irregular mesh (actually we used much denser grids in the simulation than in Fig. 5). Note that for any irregular boundary geometry, it is always possible to cover the domain using triangular elements as we have done here for the two cylinders. The macroscopic velocity field between two cylinders is taken to be zero initially. The outer cylinder then suddenly begins to rotate with a constant angular velocity Ω while the inner cylinder is kept at rest for all times. Note that this particular problem has a rotational symmetry so that an appropriate finite difference scheme in cylindrical coordinates can also handle it [18]. The finite volume scheme reported here needs no assumption of symmetry and thus is capable of handling a wide variety of complex geometries without modification. In Fig. 6 we show the velocity profile in the angular direction from the simulation when the system reaches a steady state, compared with the theoretical solution to the Navier-Stokes equation. The radii of the two cylinders are $R_1=50$ and $R_2=100$ and the angular velocity of the outer cylinder is $\Omega=0.0005$ rad per unit time. The mesh spacing is at the scale of 1 and the relaxation time τ is taken to be 0.1. The average density ρ_0 is set to be 1.0. We updated the system for 3×10^6 time steps with $dt=0.01$ and observed it to reach a steady state. From Fig. 6 one can see the agreement between

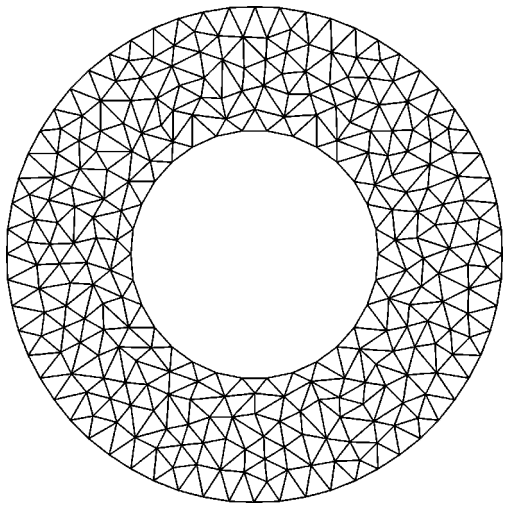

FIG. 5. An irregular mesh between two coaxial cylinders.

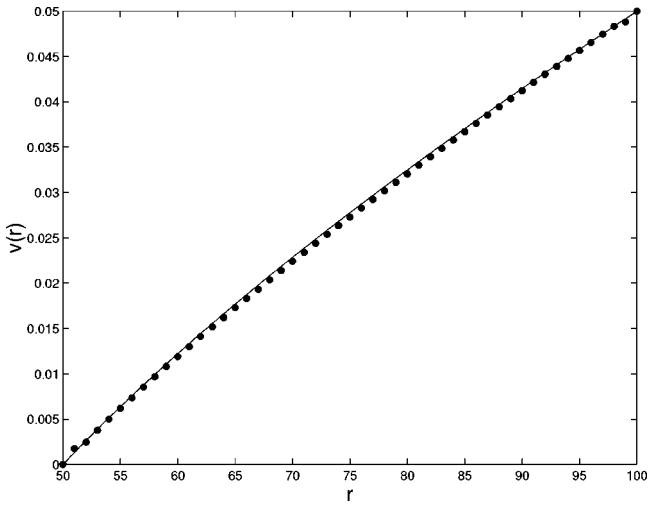

FIG. 6. The static velocity profile of flow between two coaxial cylinders (points), compared with the theoretic solution (curve) of the Navier-Stokes equation $v(r)=ar-b/r$ with $a=\frac{2}{3}\times10^{-3}$ and $b=\frac{5}{3}$. Here each point is the average of the angular velocities at the nodes in a circular layer between $r-1/2$ and $r+1/2$ with r integers.

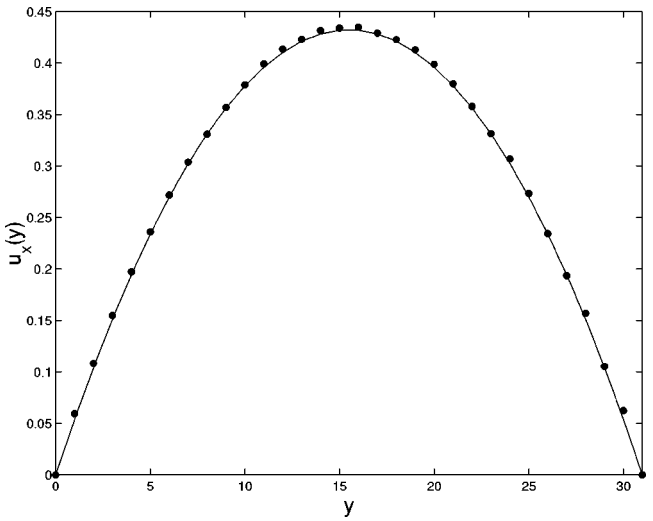

FIG. 7. Velocity profile for Poiseuille flow in a system of size $L=W=31$. The mesh type is of Fig. 2(c) and mesh size is in the order of 1.0. The forcing level is $F=0.000\,12$. We use $\rho=1.0$, $\tau=0.1$ and $dt=0.01$. The steady state is reached after $T=1.5\times10^6$ time steps. Here each point is an average of velocity over nodes the y coordinates of which are between $y-1/2$ and $y+1/2$ with y integers. The solid line represents the exact solution to the Navier Stokes equation $u_x^{(exact)}(y)=u_{max}(1-(y-c)^2/b)$ where $y=c$ is the location of the center of the channel b is the half width of the channel and $u_{max}=FW^2/(8\rho\nu)$ with ν the kinematic viscosity. For our system, $b=c=W/2=15.5$ and $\nu=\tau/3$.

the computed and the theoretical results is quite good. The relative global difference between the computed velocity and the exact velocity solution to the Navier-Stokes equation is 1.0% where the exact solution [23] is $u_r^{(exact)}=ar-b/r$ with $a=\Omega R_2^2/(R_2^2-R_1^2)=\frac{2}{3}\times10^{-3}$ and $b=\Omega R_1^2R_2^2/(R_2^2-R_1^2)=\frac{5}{3}$.

C. Poiseuille flow

We have also simulated a forced two-dimensional channel flow (Poiseuille flow) using the finite volume scheme. Here we present the results from simulations on the irregular mesh depicted in Fig. 2(c).

The LBE for the forced system now in general reads as

$$\frac{\partial f_i}{\partial t}+\mathbf{v}_i\cdot\nabla f_i=-\frac{1}{\tau}(f_i-f_i^{eq})+\alpha\mathbf{v}_i\cdot\mathbf{F}, \tag{20}$$

where F is the body force applied in the +x direction and the coefficients α equals to $1.0/\Sigma_i v_{ix}^2=1.0/\Sigma_i v_{iy}^2=1/6$. The last term in the above equation can be taken into consideration easily by the finite volume scheme by integrating over the control volume.

Figure 7 is the x-component velocity profile obtained from our simulation. The system has a length of $L=31$ and a width of $W=31$ with mesh size in the order of 1.0. The forcing level is $F=0.000\,12$. We started the computation from an initial state with zero macroscopic velocity in the system. The initial density is $\rho=1.0$. The system is found to have reached a steady state after we update the system for $T=1.5\times10^6$ time steps. Here the relaxation time τ is equal to 0.1 and the time difference between two successive time steps is $dt=0.01$. In Fig. 7 the solid line represents the exact solution to the Navier-Stokes equation $u_x^{(exact)}(y)=u_{max}(1-(y-c)^2/b)$ where $y=c$ is the location of the center of the channel and b is the half width of the channel, and $u_{max}=FW^2/(8\rho\nu)$ with ν the kinematic viscosity. For our system, $b=c=W/2=15.5$ and $\nu=\tau/3$. The error between the computed and the exact velocity profile is less than 1.0%.

IV. DISCUSSION AND CONCLUDING REMARKS

In this paper we have used the nine-bit model along with triangular mesh, thus our model is a hybrid model. One can also use the seven-bit model with the triangular mesh by replacing Eqs. (3), (4), and (7) with the corresponding ones used in the conventional LBM models [17], leaving all the others unchanged. In the special case of using seven-bit velocities and regular triangular lattice, one can reach a numerical scheme close to the so-called D2Q7 model. It is thus interesting to make a comparison between this special case and the D2Q7 model. When using regular triangular lattice, for each lattice point $\mathbf{x}$ we have six nearest neighbors (NN's). The seven-bit velocities are equal to the six bond vectors ($\mathbf{e}_i$) connecting one lattice site to its NN (given the ''speed of light'' to be unity), along with a zero velocity. They are

$$\mathbf{v}_i=\mathbf{e}_i=\begin{cases}(0,0), & i=0,\\ (\cos[(i-1)\pi/3],\sin[(i-1)\pi/3]), & i=1,2,\ldots,6.\end{cases} \tag{21}$$

Using the above update procedures described in Sec. II, we can write down the explicit update rule for the special case of using regular triangular lattice. It reads

$$f_i(\mathbf{x},t+dt)=f_i(\mathbf{x},t)-\frac{dt}{3}\sum_{\alpha=0}^{\alpha=6}(\mathbf{v}_i\cdot\mathbf{e}_\alpha)f_i(\mathbf{x}+\mathbf{e}_\alpha,t) - \frac{dt}{108\tau}\left[66\delta f_i(\mathbf{x},t)+7\sum_{\alpha=1}^{\alpha=6}\delta f_i(\mathbf{x}+\mathbf{e}_\alpha,t)\right], \tag{22}$$

where $\delta f_i=f_i-f_i^{eq}$. For comparison, we have the update rule for the D2Q7 model,

$$f_i(\mathbf{x},t+1)=f_i(\mathbf{x}-\mathbf{e}_i,t)-\frac{1}{\tau}\delta f_i(\mathbf{x}-\mathbf{e}_i,t). \tag{23}$$

One can see there are several differences between the two models. Firstly, the D2Q7 model is an upwind model, which means that the f_is at one lattice site $\mathbf{x}$ depends only on the f_i at the upstream site $\mathbf{x}-\mathbf{e}_i$, given a velocity direction $\mathbf{v}_i$ (i.e., $\mathbf{e}_i$). In our model, contributions from all directions are included for a given velocity direction. f_i at one lattice site gathers information from all the six NN sites as well as from its own site at previous time step. Secondly, the first term of the right-hand side (rhs) of Eq. (23) can be expressed as $f_i(\mathbf{x},t)-[f_i(\mathbf{x},t)-f_i(\mathbf{x}-\mathbf{e}_i,t)]$, which should be compared with the first two terms of the rhs of Eq. (22). In our update equation, the flux term [the second term of the rhs of Eq. (22)] can be viewed as a weighted average of three differences of f_i at opposite directions $[f_i(\mathbf{x}+\mathbf{e}_1,t)-f_i(\mathbf{x}+\mathbf{e}_4,t)$, $f_i(\mathbf{x}+\mathbf{e}_2,t)-f_i(\mathbf{x}+\mathbf{e}_5,t)$, and $f_i(\mathbf{x}+\mathbf{e}_3,t)-f_i(\mathbf{x}+\mathbf{e}_6,t)]$. The flux term in the D2Q7 model is just an upwind difference of f_is. Similarly, our collision term is a weighted average of δf_i at the NN sites as well as at its own site, while in the D2Q7 model it is simply a δf_i at the upwind site. Thirdly, by our ''average'' update rule, the viscosity is no longer proportional to $\tau-1/2$ as in the D2Q7 model, but to τ. This means the negative (unphysical) part of viscosity does not result from our update rule. In the finite difference model [18], it was also observed that the negative part of viscosity disappear in a lattice model by taking account the downwind operator.

We should admit that there is a speed slowdown of computation compared with the classical LBM models which lead to fast numerical implementation and short codes. Here we have to look at a map of the nearest neighbors for one chosen site in the computation. This is due to the lack of regularity in the connectivity of mesh sites. Apart from this, the program is still simple to code. If we do not use unstructured meshes, but use the structured (but not regular) meshes such the ones used in Refs. [14,15], the speed is two to three times slower than the classical LBM models.

It is also interesting to make a comparison between the above scheme and the finite-volume scheme of Succi and co-workers [14]. First, their model is a cell-centered finite-volume method while our scheme is a cell-vertex model. In the model of Succi and co-workers [14], the ''coarse grain'' density distribution is the unknown for each macrocell. So, to calculate the flux term they have to extrapolate the f values from the ''coarse grain'' density distributions according to the arrangement of macrocells using piecewise constant or piecewise linear extrapolation. Second, in their model, the piecewise constant extrapolation for the streaming operator causes serious problems of numerical diffusion. Even for the piecewise linear interpolation, numerical diffusion does not disappear. To minimize this numerical diffusion, a free parameter is then needed to be adjusted for each problem on a case by case basis. Third, even for the meshes with simple connectivity of rectangular lattice, their empirical formulas for the streaming coefficients are very complex and one could imagine the difficulty of using irregular meshes with arbitrary connectivity.

Some difficulty in using unstructured meshes also exists in the model of He, Luo, and Dembo [15]. It is not clear how to make a unique interpolation on a unstructured mesh from density distributions not on the grid points to obtain the density distributions on grid points for the next time step within this model.

By comparison, our scheme was proposed based on the standard finite-volume methods. It involves a minimum of approximation and does not need to introduce any free parameters. We have not observed any numerical diffusion problems in our finite-volume scheme. The scheme does not require a special mesh connectivity and it is easy to apply to other kinds of meshes (such as quadrilateral elements in 2D and tetrahedral and hexahedral elements in 3D) by replacing the standard interpolation we used here for triangular elements with other standard interpolation procedure suitable for the relevant elements. The Courant-Friedricks-Lewey (CFL) condition in the current finite volume scheme is found to be of the form $v_i dtc/h\leqslant 1$, where h is a minimum length scale of the control volume and c is a constant depending on the shape of the control volume. Thus one can avoid the instability by decreasing the integration time, or by changing the grid length at some positions.

To conclude, we have proposed a finite volume scheme for the LBM that is flexible to use on any irregular meshes. Complex boundary geometries can be handled easily with the finite-volume lattice Boltzmann scheme. We have tested that the finite-volume scheme works well for two-dimensional Taylor vortex flow, shear flows between two parallel plates and between two cylinders as well as Poiseuille flow. The application of this scheme to a variety of complex geometries is thus expected. Further applications and developments of the finite-volume scheme is under way.

ACKNOWLEDGMENTS

This work was supported in part by NSF Grant No. ASC-9418357 for the Pharoh MetaCenter Regional Alliance, in part by PRF under Contract No. 33160-GB9 and the Research Corporation under Grant No. CC4250. The simulations were performed on the Cray T3E at the Ohio Supercomputer Center.

[1] *Lattice Gas Method for Partial Differential Equations*, edited by G. D. Doolen (Addison-Wesley, Redwood City, CA, 1990); *Lattice Gas Methods: Theory, Applications and Hardware*, edited by G. D. Doolen (MIT, Cambridge, MA 1991).
[2] R. Benzi, S. Succi, and M. Vergassola, Phys. Rep. **222**, 145 (1992).
[3] J. Stat. Phys. **81**, 10 (1995), special issue on lattice-based models and related topics, edited by J. L. Lebowitz, S. A. Orszag, and Y. H. Qian.
[4] S. Chen and G. D. Doolen, Annu. Rev. Fluid Mech. **30**, 329 (1998).
[5] A. K. Gunstensen, D. H. Rothman, S. Zaleski, and G. Zanetti, Phys. Rev. A **43**, 4320 (1991).
[6] E. G. Flekkøy, U. Oxaal, J. Feder, and T. Jøssang, Phys. Rev. E **52**, 4952 (1995); E. G. Flekkøy, *ibid.* **47**, 4247 (1993).
[7] X. Shan and H. Chen, Phys. Rev. E **47**, 1815 (1993); X. Shan and G. Doolen, J. Stat. Phys. **81**, 379 (1995).
[8] H. Xi and C. Duncan (unpublished).
[9] E. G. Flekkøy and H. J. Herrmann, Physica A **199**, 1 (1993).
[10] U. Frisch, B. Hasslacher, and Y. Pomeau, Phys. Rev. Lett. **56**, 1505 (1986).
[11] F. J. Higuera, S. Succi, and R. Benzi, Europhys. Lett. **9**, 345 (1989); F. J. Higuera and J. Jeménez, *ibid.* **9**, 663 (1989).
[12] H. Chen, S. Chen, and W. H. Mattaeus, Phys. Rev. A **45**, R5339 (1991); Y. H. Qian, D. d'Humières, and P. Lallemand, Europhys. Lett. **17**, 479 (1992).
[13] P. L. Bhatnagar, E. P. Gross, and M. Krook, Phys. Rev. **94**, 511 (1954).
[14] S. Succi, G. Amati, and R. Benzi, J. Stat. Phys. **81**, 5 (1995); F. Nannelli and S. Succi, *ibid.* **68**, 401 (1992); G. Amati, S. Succi, and R. Benzi, Fluid Dyn. Res. **19**, 289 (1997).
[15] X. He, L.-S. Luo, and M. Dembo, J. Comput. Phys. **129**, 357 (1996).
[16] G. Peng, H. Xi, C. Duncan, and S.-H. Chou, Phys. Rev. E **58**, R4124 (1998).
[17] X. He and L.-S. Luo, Phys. Rev. E **55**, R6333 (1997); *ibid.* **56**, 6811 (1997).
[18] N. Cao, S. Chen, S. Jin, and D. Martínez, Phys. Rev. E **55**, R21 (1997).
[19] Z. Weng, C. Duncan, and H. Xi (unpublished).
[20] S. Chen, D. Martinez, and R. Mei, Phys. Fluids **8**, 2527 (1996).
[21] C. Hirsch, *Numerical Computation of Internal and External Flows, Vol. I: Fundamentals of Numerical Discretization* (Wiley, Chichester, 1988).
[22] J. E. Akin, *Application and Implementation of Finite Element Methods* (Academic, New York, 1982).
[23] D. J. Tritton, *Physical Fluid Dynamics* (Clarendon, Oxford, 1988).