PHYSICAL REVIEW E **68**, 016303 (2003)

Three-dimensional lattice Boltzmann model for compressible flows

Chenghai Sun* and Andrew T. Hsu†

Department of Mechanical Engineering, Indiana University-Purdue University, Indianapolis, Indiana 46202-5132, USA

(Received 4 December 2001; revised manuscript received 18 December 2002; published 11 July 2003)

A three-dimensional compressible lattice Boltzmann model is formulated on a cubic lattice. A very large particle-velocity set is incorporated in order to enable a greater variation in the mean velocity. Meanwhile, the support set of the equilibrium distribution has only six directions. Therefore, this model can efficiently handle flows over a wide range of Mach numbers and capture shock waves. Due to the simple form of the equilibrium distribution, the fourth-order velocity tensors are not involved in the formulation. Unlike the standard lattice Boltzmann model, no special treatment is required for the homogeneity of fourth-order velocity tensors on square lattices. The Navier-Stokes equations were recovered, using the Chapman-Enskog method from the Bhatnagar-Gross-Krook (BGK) lattice Boltzmann equation. The second-order discretization error of the fluctuation velocity in the macroscopic conservation equation was eliminated by means of a modified collision invariant. The model is suitable for both viscous and inviscid compressible flows with or without shocks. Since the present scheme deals only with the equilibrium distribution that depends only on fluid density, velocity, and internal energy, boundary conditions on curved wall are easily implemented by an extrapolation of macroscopic variables. To verify the scheme for inviscid flows, we have successfully simulated a three-dimensional shock-wave propagation in a box and a normal shock of Mach number 10 over a wedge. As an application to viscous flows, we have simulated a flat plate boundary layer flow, flow over a cylinder, and a transonic flow over a NACA0012 airfoil cascade.

DOI: 10.1103/PhysRevE.68.016303 PACS number(s): 47.40.−x, 51.20.+d

I. INTRODUCTION

The lattice Boltzmann method (LBM) [1–3] is a relatively new numerical approach for simulating complex flow and transport phenomena in cases where direct solution of the Navier-Stokes equations is not practical. Unlike conventional computational fluid dynamics (CFD) methods based on macroscopic continuum equations, the LBM uses a mesoscopic equation, i.e., the Boltzmann equation, to determine macroscopic fluid dynamics. The LBM is flexible, has broad applicability, and may be easily adapted for parallel computing. It has been successfully applied to multiphase and multicomponent fluids, flows through porous media, and solid particle suspensions.

The LBM originated from a Boolean model known as the lattice gas automata (LGA) [4,5]. In a LGA method, the local equilibrium distribution is described by the Fermi-Dirac statistics. As a result, LGA has several shortcomings: high statistical noises, the violation of Galilean transformation invariance in their resulting hydrodynamics equations, and the failure in high Reynolds number computations. To eliminate noise, the Boltzmann equation was used to simulate the lattice-gas automata [6,7]; however, other problems, i.e., non-Galilean invariance and low Reynolds number, remained. These difficulties led to the development of the LB method [1–3]. Higuera, Succi, and Benzi and Benzi, Succi, and Vergassola [3] simplified the collision term by a linear operator. Chen, Chen, and Matthaeus [1] and Qian, d'Humiéres, and Lallemand [2] used a simpler collision operator of the Bhatnagar-Gross-Krook (BGK) type [8]. The equilibrium distribution was an approximation of the Maxwellian equilibrium distribution. Galilean invariance was guaranteed in these LB models. The LB models of BGK type [1,2] have only a single ratio of viscosity to thermal conductivity, while the models of linear collision operator [3] allow for independently varying viscosities and thermal conductivities. The LB models have been successfully applied to various physical problems, such as single component hydrodynamics, multiphase and multicomponent fluid flows, magnetohydrodynamics, reaction-diffusion systems, flows through porous media, and other complex systems at small Mach numbers [9,10].

Unfortunately, as a new CFD tool, the general LB method developed in the past suffered from the constraint of small Mach number because the particle velocities belong to a finite set, and the resulting macroscopic velocity is always much smaller than the speed of sound calculated from the microscopic diffusion velocity.

Efforts have been made to increase the allowable Mach number range and to incorporate the effects of temperature into lattice Boltzmann simulations. Choosing a modified equilibrium distribution, Alexander, Chen, and Doolen [11] replicated the Burger's equation with a controllable sound speed. Yu and Zhao [12] introduced an attractive force to reduce the sound speed and to alleviate the small Mach number restriction; however, the energy equation was not recovered in their formulation. Palmer and Rector [13] formulated a thermal model that can simulate temperature variations in a flow, but high Mach number effects were not included in that study.

---

*Present address: Steele Lab., Department of Radiation Oncology, Massachusetts General Hospital and Harvard Medical School, Cox 7, Fruit St. Boston, Massachusetts 02114, USA. Email address: csun@hms.harvard.edu

†Corresponding author: FAX: 317-274-9744; email address: anhsu@iupui.edu

1063-651X/2003/68(1)/016303(14)/$20.00

Two schemes related to the LB method, the gas-kinetic theory [14,15] and the discrete-velocity model [16,17], had been used to simulate the compressible Euler equations. In both of the above works, the flux-splitting approach with total variation diminishing (TVD) flux limitation was employed to determine the mean flux to neighboring cells. Nadiga [16,17] introduced an adaptive-velocity concept in the discrete-velocity model for compressible inviscid flows. Huang *et al.* [18] similarly used adaptive discrete velocities to simulate one-dimensional shock waves. Only under special circumstances, the Boltzmann equation used in these methods is equivalent to the lattice Boltzmann equation [19], but the lattice Boltzmann equation is much easier to solve.

Recently, we proposed a locally adaptive LB model on hexagonal lattice [20] based on a large particle-velocity set, so that the mean flow may have a high velocity; however, the support set of the equilibrium distribution is quite small and similar to the adaptive velocities of Nadiga's Euler solver [17]. This model is suitable for a wide range of Mach numbers and does not consume much computer resource. Compressible Navier-Stokes equations including the energy equation are derived from the BGK lattice Boltzmann equation; therefore, this model can simulate compressible viscous flows that include heat transfer [21,22]. If the viscous terms are considered as discretization error and a slip wall condition is employed, the solution can be compared with compressible Euler solutions. The numerical simulations showed that the model has the capability of solving compressible Euler flows with strong shocks [20,23,24] and has high parallel computing efficiency [25,26]. This locally adaptive LB model has been also formulated on a two-dimensional square lattice [27]. All the previous simulations were carried out for periodical or flat wall boundary conditions, or a combination of the two.

In LBM, the boundary conditions have been directly adopted from the lattice-gas automaton method. A common method of modeling no-slip walls in LBM simulations is to use the bounce-back boundary condition in which particles that stream into the walls ‘‘bounce back’’ and exit the wall in the direction from which they came. It has been noted that the bounce-back boundary condition is second order for walls aligned with the lattice; however, it gives only first-order accuracy at the curved boundaries [28,29]. Several boundary treatments have been proposed for achieving second-order accuracy for no-slip velocity conditions on curved walls [30–33]. In these treatments boundary conditions for the particle distribution function had to be handled with given macroscopic quantities. In complicated fluid flows, boundary conditions might include a combination of velocity, density, temperature, and their derivatives. To a certain degree, achieving self-consistent boundary conditions with a given accuracy is as important as developing numerical schemes themselves.

In the present work, we establish a three-dimensional LB model on a cubic lattice. Due to the simple form of the equilibrium distribution, the fourth-order velocity tensors are not involved in the formulation. Unlike traditional lattice Boltzmann models, no special treatment is required for the homogeneity of fourth-order velocity tensors on square lattices. In this formulation, the second-order discretization errors in velocity have been eliminated so that the model is more accurate for viscous flows. The model is capable of simulating both viscous and inviscid compressible flows, and is shock capturing. In the traditional LB models the boundary conditions must be given in terms of the particle distribution function, while the present scheme deals only with the equilibrium distribution that depends only on fluid density, velocity, and internal energy. The boundary condition on a curved wall is easily treated by extrapolating macroscopic variables.

II. THREE-DIMENSIONAL COMPRESSIBLE LB MODEL

A. Basic equations

Conventionally, the LB method solves a discretized BGK type of the Boltzmann equation, where the unknown variable is the particle density distribution function $f(\mathbf{x},\mathbf{c}_j,t)$, where $\mathbf{x}$ is the location of the lattice node, and $\mathbf{c}_j$ is the particle velocity. In the conventional LB models, the particle velocity magnitude is restricted to $c_j=l/\Delta t$, where l is the length of the side of the lattice. The macroscopic velocity obtained from this model can only be less than c_j. On the other hand, the speed of sound, in general, is of the order of $l/\Delta t$. Thus, the Mach number of the solution is severely limited and high-speed compressible flows cannot be solved.

In order to overcome this limitation on the macroscopic velocity, we introduce a larger particle velocity set, $S=\{\mathbf{c}\}$, in the present model, where $\mathbf{c}$ is the migrating velocity of the particles. The migrating velocity $\mathbf{c}$, unlike $\mathbf{c}_j$ of the standard LB models, is unrestricted so that the particles are allowed to travel any number of lattice lengths.

In the final LB model, only a discrete finite subset of the migrating velocity $\mathbf{c}$ is used; this causes errors in the macroscopic solution. In order to minimize the discretization error, we introduce a continuous mass m, momentum $\boldsymbol{\xi}$, and energy ζ, transported by a particle. The migrating velocity $\mathbf{c}$ is only used to calculate the location of the particle.

We have $\boldsymbol{\xi}$ and $\mathbf{c}\in D_1$, where D_1 is a bounded domain in $\mathfrak{R}^3$; m and $\zeta\in D_0$, where D_0 is a bounded domain in $\mathfrak{R}$. In a standard LB model, space, time, and the particle velocity are all discrete. If we let $\boldsymbol{\xi}$ and ζ take the values $m\mathbf{c}$ and $m\mathbf{c}^2/2$, respectively, then the present model will be consistent with the conventional LB model; however, the velocity set $S(\mathbf{c})$ is still larger than that of the conventional model.

With the above definition of velocities, momentum, and energy, we now define $f(\mathbf{x},\mathbf{c},m,\boldsymbol{\xi},\zeta,t)$ as the particle density distribution function for particles located at $\mathbf{x}$, with a continuous mass m, momentum $\boldsymbol{\xi}$, and energy ζ to be transported at a migrating velocity $\mathbf{c}$. In other words, these particles will move to $\mathbf{x}+\mathbf{c}\Delta t$ after Δt, and transporting with them a mass m, momentum $\boldsymbol{\xi}$, and energy ζ.

The macroscopic quantities, i.e., mass ρ, momentum $\rho\mathbf{v}$, and energy ρE, are defined as

$$\mathbf{Y}\equiv\sum_{\mathbf{c}}\int_D \boldsymbol{\eta} f(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t)\,d\boldsymbol{\eta}, \qquad (1)$$

where $D \equiv D_0 \times D_1 \times D_0$, $\mathbf{Y} \equiv (\rho, \rho\mathbf{v}, \rho E)$, and $\boldsymbol{\eta} \equiv (m, \boldsymbol{\xi}, \zeta)$, $d\boldsymbol{\eta} \equiv dm\, d\boldsymbol{\xi}\, d\zeta$, and $f(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t) \equiv f(\mathbf{x},\mathbf{c},m,\boldsymbol{\xi},\zeta,t)$.

In an LB model, the Boltzmann equation is written as

$$f(\mathbf{x}+\mathbf{c}\Delta t,\mathbf{c},\boldsymbol{\eta},t+\Delta t) - f(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t) = \Omega, \tag{2}$$

where

$$\Omega = -\frac{1}{\tau}[f(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t) - f^{eq}(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t)], \tag{3}$$

and $f^{eq}(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t)$ is the equilibrium distribution determined by the macroscopic mass, momentum, and energy, and satisfies

$$\sum_{\mathbf{c}} \int_D \boldsymbol{\eta} f^{eq}(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t)\, d\boldsymbol{\eta} = \mathbf{Y}. \tag{4}$$

The macroscopic conservation equations can be derived by the Chapman-Enskog expansion of Eq. (2) as follows: we choose $\Delta t = \varepsilon T$, where T is a reference time scale and ε is a typical small parameter. We write the solution of Eq. (2) in an asymptotic expansion near the equilibrium distribution of the form [34,35]

$$f = f^{(0)} + \varepsilon f^{(1)} + \varepsilon^2 f^{(2)} + \cdots, \tag{5}$$

$$\frac{\partial \mathbf{Y}}{\partial t} = \mathbf{F}^{(0)} + \varepsilon \mathbf{F}^{(1)} + \varepsilon^2 \mathbf{F}^{(2)} + \cdots, \tag{6}$$

where $f^{(n)}$ and $\mathbf{F}^{(n)}$ depend only on $\mathbf{Y}$ and its successive gradients. $f^{(0)} = f^{eq}$ is completely determined by the macroscopic variables ρ, $\rho\mathbf{v}$, and ρE and satisfies Eq. (4).

Substituting Eq. (5) into Eq. (1), and subtracting Eq. (4) from the resulting equation, we obtain

$$\sum_{\mathbf{c}} \int_D \boldsymbol{\eta} f^{(n)}(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t)\, d\boldsymbol{\eta} = 0, \quad \forall n \geqslant 1. \tag{7}$$

Multiplying Eq. (3) by $\boldsymbol{\eta}$ and integrating, and considering Eqs. (1) and (4), we have

$$\sum_{\mathbf{c}} \int_D \boldsymbol{\eta} \Omega\, d\boldsymbol{\eta} = 0. \tag{8}$$

A function $\phi(\mathbf{c},\boldsymbol{\eta})$ defined on $S \times D$ is called a *collision invariant* if it satisfies the following relation:

$$\sum_{\mathbf{c}} \int_D \phi(\mathbf{c},\boldsymbol{\eta}) \Omega\, d\boldsymbol{\eta} = 0. \tag{9}$$

From Eq. (8) we know that all the components of the vector $\boldsymbol{\eta}$ are collision invariants. We hope that they are the only collision invariants in the LB model. A collision invariant other than the components of the vector $\boldsymbol{\eta}$, if it exists, is called spurious collision invariant that may cause nonphysical phenomena.

Using Taylor expansion in the left-hand side of Eq. (2) and considering Eqs. (7) and (8), we can determine $f^{(1)}$, $\mathbf{F}^{(0)}$, and $\mathbf{F}^{(1)}$. Up to order 1, Eq. (6) is written as [20,22]

$$\begin{aligned} \frac{\partial \mathbf{Y}}{\partial t} = & -\nabla \cdot \sum_{\mathbf{c}} \int_D f^{eq} \mathbf{c} \boldsymbol{\eta}\, d\boldsymbol{\eta} \\ & - \varepsilon T \left(\frac{1}{2} - \tau\right) \nabla \cdot \left[\nabla \cdot \sum_{\mathbf{c}} \int_D f^{eq} \mathbf{cc} \boldsymbol{\eta}\, d\boldsymbol{\eta} \right. \\ & \left. + \sum_{\mathbf{c}} \int_D \frac{\partial f^{eq}}{\partial \mathbf{Y}} \cdot \mathbf{F}^{(0)} \mathbf{c} \boldsymbol{\eta}\, d\boldsymbol{\eta} \right] + O(\varepsilon^2). \end{aligned} \tag{10}$$

This is the macroscopic conservation equation. It depends on the distribution of f^{eq}. When the equilibrium distribution is properly determined, Eq. (10) becomes the continuity equation, the Navier-Stokes equations, and the energy equation. If we neglect the first-order term in ε, the Euler equations can be recovered.

B. Equilibrium distribution

The basic idea of the present LBM for compressible flows is to allow particles to travel a distance that is much larger than the grid size of the lattices, and by doing so the velocity can now be arbitrarily larger (or smaller) than the speed of sound, which is determined by the molecular diffusion velocity. In order to achieve this goal, we symbolically decompose the discrete migrating velocity $\mathbf{c}$ of a particle into two components:

$$\mathbf{c} = \mathbf{v} + \mathbf{c}', \tag{11}$$

where the first component $\mathbf{v}$ is the macroscopic fluid velocity, and the second component $\mathbf{c}'$ is the molecular diffusion velocity.

The macroscopic velocity $\mathbf{v}$ is a continuous quantity, which can be used to evaluate the momentum and kinetic energy carried by a particle, but it cannot be used to determine particle migration since in an LB model a particle must move from one node to another at a time step of a time-marching procedure. To address this issue, we introduce a discrete macroscopic velocity $\mathbf{v}_k$ as an approximation to $\mathbf{v}$. Supposing that the macroscopic velocity $\mathbf{v}$ carries a particle from its originating node into an interior point of a lattice cube, we introduce a set of correction velocity vectors $\mathbf{v}_k'$ $(k = 1,2,\ldots,8)$ that will carry fractions of the particle from the interior point to the eight nodes of the destination lattice cube (see Fig. 2). We then define the modified macroscopic velocity as the sum of the exact macroscopic velocity $\mathbf{v}$ and the correction velocity $\mathbf{v}_k'$:

$$\mathbf{v}_k = \mathbf{v} + \mathbf{v}_k'. \tag{12}$$

For high-speed flows the fluctuating velocities $\mathbf{v}_k'$ are small compared to $\mathbf{v}$.

With this modification to the macroscopic velocity, the molecular diffusion velocity can now easily be defined on a uniform lattice. We consider a symmetric vector set $\{\mathbf{c}_j', j = 1,\ldots,n\}$ connecting a node to its equal distanced neighboring nodes, where n is the number of vectors. If we take $\Delta t = 1$, then these vectors are equivalent to the length of the lattice sides. In the following description, $\Delta t = 1$ is implied,

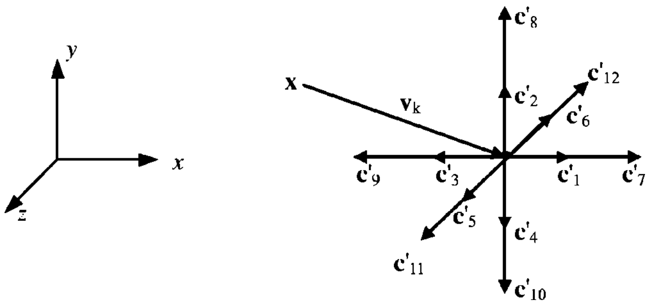

FIG. 1. Definition of symmetric velocity $\mathbf{c}'_j$ for $j=1,...,12$ with two velocity levels. The length of the vectors must be a multiple of the grid length. **x** is a lattice node. $\mathbf{v}_k$ is an approximation of **v** and is defined in Fig. 2.

and velocities are used as distances without further explanation. We choose the number of vectors $n=12$, with two velocity levels and six directions for each level. The modulus of $\mathbf{c}'_j$ is c'_1 for $j=1,...,6$ and c'_2 for $j=7,...,12$ (see Fig. 1). c'_1 and c'_2 must be a multiple of the grid length. The symmetric particle velocity set $\{\mathbf{c}'_j\}$ defined here is similar to a particle velocity set of the traditional lattice Boltzmann or lattice gas methods with multivelocity levels [34].

With the above definitions of macroscopic and microscopic velocities, analogous to Eq. (11), we define a discrete velocity set $S_0(\mathbf{c}_{jk})$ through the following relation:

$$\mathbf{c}_{jk}=\mathbf{v}_k+\mathbf{c}'_j\,. \tag{13}$$

Here $S_0(\mathbf{c}_{jk})$ is a subset of the velocity set $S(\mathbf{c})$ defined in Sec. II A.

Figures 1 and 2 demonstrate how the discrete particle velocities $\mathbf{c}_{jk}$ ensure the particle to move from one node to another during one time step. In summary, the index k varying from 1 to 8 is associated to the discretization of the macroscopic velocity **v** and the index j varying from 1 to 12 represents the 12 vectors $\mathbf{c}'_j$ with six symmetric directions and two levels. Thus, the quantity $\mathbf{c}_{jk}$ represents an array of 96 numbers.

In the following, we will define the equilibrium distribution f^{eq}. Our goal here is to design a model that is as simple as possible under the condition that the correct macroscopic equations (Navier-Stokes equations) can be recovered. We restrict ourselves to the set of particles at $\mathbf{c}=\mathbf{c}_{jk}$, $m=m_{jk}$, $\boldsymbol{\xi}=\boldsymbol{\xi}_{jk}$, and $\zeta=\zeta_{jk}$, where m_{jk}, $\boldsymbol{\xi}_{jk}$, and ζ_{jk} are mass, momentum, and energy carried by a particle with a velocity $\mathbf{c}_{jk}$, and, as is shown later in this section, can be determined by the macroscopic variables ρ, **v**, and E.

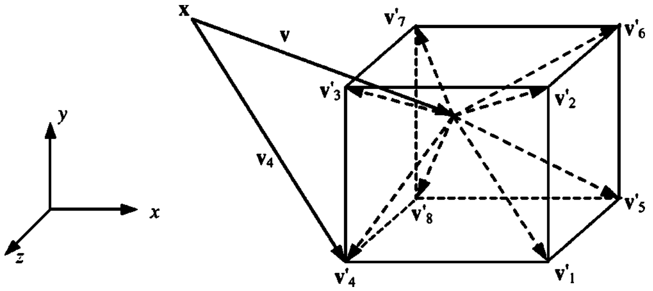

FIG. 2. Definition of $\mathbf{v}_k$ and $\mathbf{v}'_k$, for $k=1,...,8$. $\mathbf{v}_k$ is an approximation of **v** and connects the lattice node **x** to another node on the cubic of 1 grid in size. Only $\mathbf{v}_4$ is drawn in the figure. $\mathbf{v}'_k$ is the difference between $\mathbf{v}_k$ and **v**.

The equilibrium distribution is defined as follows:

$$f^{eq}(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t)\equiv\begin{cases} d_{jk}\delta(m-m_{jk})\,\delta(\boldsymbol{\xi}-\boldsymbol{\xi}_{jk})\,\delta(\zeta-\zeta_{jk}) & \text{for } \mathbf{c}=\mathbf{c}_{jk}\\ 0 & \text{for other } \mathbf{c},\end{cases} \tag{14}$$

where $\delta(\boldsymbol{\xi})$ is a δ function, $\delta(\boldsymbol{\xi})=\mathbf{0}$ for $\boldsymbol{\xi}\neq\mathbf{0}$, and $\int g(\boldsymbol{\xi})\,\delta(\boldsymbol{\xi})d\boldsymbol{\xi}=g(\mathbf{0})$, and d_{jk} is defined as the following: We introduce a density fraction $\alpha_k=\rho_k/\rho$ and suppose d_{jk} to have a factorized form related to the density fraction α_k as

$$d_{jk}=\alpha_k d_1\,,\quad j=1,...,6\ \text{ and }\ d_{jk}=\alpha_k d_2\,,\quad j=7,...,12. \tag{15}$$

The quantity ρ_k in the density fraction can be regarded as the amount of mass carried by the fractional particles that are sent to the eight vertices of a lattice cube and can be determined from the following relations:

$$\begin{aligned}
&\rho_1=\rho|u'_7v'_7w'_7|,\quad \rho_2=\rho|u'_8v'_8w'_8|,\\
&\rho_3=\rho|u'_5v'_5w'_5|,\quad \rho_4=\rho|u'_6v'_6w'_6|,\\
&\rho_5=\rho|u'_3v'_3w'_3|,\quad \rho_6=\rho|u'_4v'_4w'_4|,\\
&\rho_7=\rho|u'_1v'_1w'_1|,\quad \rho_8=\rho|u'_2v'_2w'_2|,
\end{aligned} \tag{16}$$

where u'_k, v'_k, and w'_k are the components of $\mathbf{v}'_k$ in the x, y, and z directions, respectively. The redistribution of mass based on Eq. (16) ensures the conservation of mass.

Internal energy can be defined as

$$e=E-\tfrac{1}{2}\mathbf{v}^2. \tag{17}$$

The coefficient d_1 and d_2 used in Eq. (15) are defined as function of density ρ and internal energy e:

$$d_1=\rho\,\frac{c'^2_2-D(\gamma-1)e}{b(c'^2_2-c'^2_1)},\quad d_2=\rho\,\frac{D(\gamma-1)e-c'^2_1}{b(c'^2_2-c'^2_1)}, \tag{18}$$

where c'_1 and c'_2 are the modulus of the particle velocities, $b=6$ is the number of velocity directions for each level, and γ is the specific heat ratio. In order to ensure the positivity of d_1 and d_2, c'_1 and c'_2 must satisfy $c'^2_1<D(\gamma-1)e<c'^2_2$. However, c'_1 and c'_2 are not completely determined. In practice, c'_1 is set to be the integer part of $\sqrt{D(\gamma-1)e}$ and $c'_2=c'_1+1$. Therefore, c'_1 and c'_2 are adaptive to the internal energy e.

The particle mass, momentum, and energy (m_{jk}, $\boldsymbol{\xi}_{jk}$, and ζ_{jk}) in Eq. (14) are defined as

$$m_{jk}=1-\chi_{jk}\,, \tag{19}$$

$$\boldsymbol{\xi}_{jk}=\mathbf{v}+\mathbf{c}'_j-\chi_{jk}\mathbf{v}, \tag{20}$$

$$\zeta_{jk}=\tfrac{1}{2}(\mathbf{v}^2+2\mathbf{c}'_j\cdot\mathbf{v}+\overline{c'^2})+\Phi-\chi_{jk}[\tfrac{1}{2}(\mathbf{v}^2+\overline{c'^2})+\Phi], \tag{21}$$

where

$$\chi_{jk}=\frac{D}{2\mathbf{c}'^2_j}(\mathbf{c}'_j\cdot\mathbf{v}'_k), \tag{22}$$

$$\overline{c'^2}=\frac{1}{\rho}b(d_1c'^2_1+d_2c'^2_2), \tag{23}$$

$$\Phi=[1-(\gamma-1)D/2]e. \tag{24}$$

Φ is a potential energy introduced to help obtain an arbitrary specific heat ratio. $\overline{c'^2}$, the mean value of $\mathbf{c}'^2$, is used in Eq. (21) in order to recover the correct conduction term in the energy equation. The terms of χ_{jk} in Eqs. (19), (20), and (21) are introduced to compensate the second-order discretization errors $\mathbf{v}'_k\mathbf{v}'_k$ in the final macroscopic conservation equations.

At this point, the equilibrium distribution f^{eq} is completely defined by Eq. (14) and we have following remarks to make.

(1) The equilibrium distribution defined by Eq. (14) is a function of macroscopic variables ρ, $\rho\mathbf{v}$, and ρE only. It is similar to the beam method [36,37] and is much simpler than that of traditional LB models, which is generally a second-order polynomial in fluid velocity $\mathbf{v}$. It can be shown that Eq. (14) is equivalent to the equilibrium distribution of traditional LB models when $\mathbf{v}$ is null.

(2) $f^{eq}(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t)$ is defined for all $\mathbf{c}$, m, $\boldsymbol{\xi}$, ζ in $S\times D_0\times D_1\times D_0$. However, $f^{eq}(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t)$ is nonzero only for $(\mathbf{c},m,\boldsymbol{\xi},\zeta)$ in $\{\mathbf{c}_{jk}\}\times\{m_{jk}\}\times\{\boldsymbol{\xi}_{jk}\}\times\{\zeta_{jk}\}$, which is called the support set of $f^{eq}(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t)$. The support set is discrete and relatively small. The particle velocity $\mathbf{c}_{jk}$ is similar to the adaptive particle velocity in Refs. [17,18].

(3) Now we are in a position to comment on the size of the velocity set $S(=\{\mathbf{c}\})$: By definition $\mathbf{c}_{jk}=\mathbf{v}_k+\mathbf{c}'_j$, where $\mathbf{v}_k$ depends on $\mathbf{v}$ and the modulus of $\mathbf{c}'_j$ depends on internal energy e. Therefore $\mathbf{c}_{jk}$ varies from node to node and from time to time because of the variation in v and e. We define a *nontrivial velocity set* S_0 as the set of all the possible $\mathbf{c}_{jk}$ for all nodes and all time steps in consideration. We suppose that the set S contains S_0 at least. The elements in S but not in S_0 are called *trivial velocity*. The greater the variation in $\mathbf{v}$ and e, the larger the set S_0. For example, S_0 is large in the case of strong shock.

The set S may contain trivial velocities in it, but they do not affect the model because for a trivial velocity $\mathbf{c}$ the equilibrium distribution $f^{eq}(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t)$ always remains zero, and so does $f(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t)$, according to the lattice Boltzmann equation (2), if f is initialized by the equilibrium distribution f^{eq} at $t=0$, which is a common practice for initial condition.

In general, we do not know the exact size of S_0 before simulation. It is a good idea to make a rough estimation of S_0 and to consider a larger set S that surely contains S_0. Otherwise, we have to resize the set S whenever a new $\mathbf{c}_{jk}$ outside S appears during the simulations. This construction of the set S ensures the recovery of the Navier-Stokes equations from the lattice Boltzmann equation with any relaxation time $\tau>1/2$. But S is not practical to handle in simulations. In Sec. III A we discuss the technique to implement this scheme in a simple and efficient way in a special case with $\tau=1$, in which case the size of the velocity set, at any given time step and given node, is restricted to 96.

(4) One will see in the following that with S defined above, the Navier-Stokes equations can be recovered even with such a simple equilibrium distribution as given in Eq. (14) because the effect of the fluid velocity $\mathbf{v}$ is taken into account in the particle velocity $\mathbf{c}_{jk}$. Due to the simple form of the equilibrium distribution, the fourth-order velocity tensors do not appear in the formulation. Unlike the standard lattice Boltzmann model on cubic lattices that usually requires 18 particle velocities to obtain the homogeneity of fourth-order velocity tensors, six directions of particle velocity $\mathbf{c}'_j$ with two levels (see Fig. 1), are symmetric enough to recover the Navier-Stokes equations. In fact, to recover the Navier-Stokes equations in the present LB model, we only have to enforce the following relations for $\mathbf{c}'_j$:

$$\sum_{j=1}^{6}\mathbf{c}'_j=\sum_{j=7}^{12}\mathbf{c}'_j=\mathbf{0},$$

$$\sum_{j=1}^{6}\mathbf{c}'_j\mathbf{c}'_j=\frac{b}{D}c'^2_1\mathbf{I}_d,\quad \sum_{j=7}^{12}\mathbf{c}'_j\mathbf{c}'_j=\frac{b}{D}c'^2_2\mathbf{I}_d, \tag{25}$$

$$\sum_{j=1}^{6}\mathbf{c}'_j\mathbf{c}'_j\mathbf{c}'_j=\sum_{j=7}^{12}\mathbf{c}'_j\mathbf{c}'_j\mathbf{c}'_j=\mathbf{0},$$

where D is the space dimension, $D=3$ for the present model. $\mathbf{I}_d$ is a unit tensor of second order.

The equilibrium distribution has the following properties.

Property 1. ρ_k defined by Eq. (16) satisfies

$$\sum_{k=1}^{8}\rho_k=\rho,\quad \sum_{k=1}^{8}\rho_k\mathbf{v}_k=\rho\mathbf{v}, \tag{26}$$

$$\sum_{k=1}^{8}\alpha_k=1,\quad \sum_{k=1}^{8}\rho_k\mathbf{v}'_k=\mathbf{0}. \tag{27}$$

Proof. The components of the vector $\mathbf{v}'_k$ satisfy the following relations (see Fig. 2):

$$\begin{aligned}&|u'_1|=|u'_2|=|u'_5|=|u'_6|,\\&|u'_3|=|u'_4|=|u'_7|=|u'_8|,\\&|v'_1|=|v'_4|=|v'_5|=|v'_8|,\\&|v'_2|=|v'_3|=|v'_6|=|v'_7|,\\&|w'_1|=|w'_2|=|w'_3|=|w'_4|,\\&|w'_5|=|w'_6|=|w'_7|=|w'_8|,\end{aligned} \tag{28}$$

$$|u_1'|+|u_4'|=1,$$

$$|v_1'|+|v_2'|=1,$$

$$|w_1'|+|w_5'|=1.$$

Equation (26) can be obtained by using the relations above and the definition of ρ_k [Eq. (16)]. Equation (27) is a direct result of Eq. (26).

Equation (27) is important to eliminate the first-order terms of $\mathbf{v}_k'$ in the macroscopic conservation equations. The reason why we assume a factorized form of d_{jk} in Eq. (15) is to make use of Eq. (27) in deriving the macroscopic conservation equations.

Property 2. The following equations hold for χ_{jk} defined by Eq. (22):

$$\sum_{j=1}^{6}\chi_{jk}=0,\quad \sum_{j=7}^{12}\chi_{jk}=0,\quad \sum_{k=1}^{8}\rho_k\chi_{jk}=0. \tag{29}$$

Proof. The first two equations are true due to the symmetry of $\mathbf{c}_j'$, i.e., the first equation of Eq. (25). The last equation holds because of the last relation of Property 1.

Property 3. Equation (4) holds for the equilibrium distribution f^{eq} defined by Eq. (14), i.e.,

$$\sum_{\mathbf{c}}\int_D \boldsymbol{\eta} f^{eq}(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t)\,d\boldsymbol{\eta}=\sum_{k,j}\boldsymbol{\phi}_{jk}d_{jk}=\mathbf{Y},$$

where

$$\boldsymbol{\phi}_{jk}\equiv[m_{jk},\boldsymbol{\xi}_{jk},\zeta_{jk}]. \tag{30}$$

The proof can be achieved by substituting f^{eq}, defined by Eq. (14), into the left-hand side of Eq. (4), and then considering the definitions of $\boldsymbol{\phi}_{jk}$, d_{jk}, d_1, and d_2 successively and using Property 1 and Property 2.

Property 4. The model satisfies the following equation of state for a perfect gas with specific heat ratio γ:

$$p=(\gamma-1)\rho e, \tag{31}$$

where p is the pressure defined by

$$p=\frac{1}{D}b(d_1c_1'^2+d_2c_2'^2). \tag{32}$$

The proof is obvious if d_1 and d_2 defined in Eq. (18) is substituted into Eq. (32).

Now we are ready to derive the macroscopic conservation equations. After the substitution of f^{eq} into Eq. (10) and the calculation of the integrals we obtain

$$\frac{\partial\mathbf{Y}}{\partial t}=-\boldsymbol{\nabla}\cdot\sum_{k,j}\left\{d_{jk}\mathbf{c}_{jk}\boldsymbol{\phi}_{jk}+\varepsilon T\left(\frac{1}{2}-\tau\right)\cdot\left[\boldsymbol{\nabla}\cdot(d_{jk}\mathbf{c}_{jk}\mathbf{c}_{jk}\boldsymbol{\phi}_{jk})+\mathbf{F}^{(0)}\cdot\frac{\partial}{\partial\mathbf{Y}}(d_{jk}\mathbf{c}_{jk}\boldsymbol{\phi}_{jk})\right]\right\}+O(\varepsilon^2), \tag{33}$$

where $O(\varepsilon^2)$ is the error term derived from the Chapman-Enskog expansion and $\boldsymbol{\phi}_{jk}$ is defined by Eq. (30).

Considering Properties 1–4, we have the following continuity, momentum, and energy equations after substituting $\boldsymbol{\phi}_{jk}$ into Eq. (33):

$$\frac{\partial\rho}{\partial t}+\boldsymbol{\nabla}\cdot(\rho\mathbf{v})=O(\varepsilon^2), \tag{34}$$

$$\frac{\partial\rho\mathbf{v}}{\partial t}+\boldsymbol{\nabla}\cdot(\rho\mathbf{v}\mathbf{v})+\boldsymbol{\nabla}p=\boldsymbol{\nabla}\cdot\{\mu[\boldsymbol{\nabla}\mathbf{v}+(\boldsymbol{\nabla}\mathbf{v})^T-(\gamma-1)\boldsymbol{\nabla}\cdot\mathbf{v}\mathbf{I_d}]\}+O(\varepsilon^2), \tag{35}$$

$$\frac{\partial\rho E}{\partial t}+\boldsymbol{\nabla}\cdot(p\mathbf{v}+\rho E\mathbf{v})=\boldsymbol{\nabla}\cdot\{\mu\mathbf{v}\cdot[\boldsymbol{\nabla}\mathbf{v}+(\boldsymbol{\nabla}\mathbf{v})^T-(\gamma-1)\boldsymbol{\nabla}\cdot\mathbf{v}\mathbf{I_d}]\}+\boldsymbol{\nabla}\cdot\{\kappa\boldsymbol{\nabla}e-(\gamma-1)e\boldsymbol{\nabla}\kappa\}+O(\varepsilon^2), \tag{36}$$

where

$$\mu=\kappa=\varepsilon T[\tau-(1/2)]\frac{1}{D}b(d_1c_1'^2+d_2c_2'^2), \tag{37}$$

μ and κ are the viscosity and heat conductivity, respectively; εT is the time step; and $O(\varepsilon^2)$, the error terms derived in the Chapman-Enskog expansion, is of higher order than the viscous term and heat conduction term in Eqs. (35) and (36) [see Eq. (37)].

In Eq. (36), the first term and the second term of the right-hand side corresponds to the dissipation and the heat conduction, respectively.

From the above analysis, one may conclude that a solution of the Boltzmann equation (2) is equivalent to a solution of the Navier-Stokes system, Eqs. (34)–(36). Nevertheless, if the viscous terms are considered as discretization error and a slip wall condition is used, the solution can be compared with compressible Euler solutions. Therefore, this model can simulate both viscous and inviscid flows.

III. SIMULATION SCHEME AND BOUNDARY CONDITIONS

A. The simulation scheme

The Boltzmann equation (2) is defined for all $(\mathbf{c},m,\boldsymbol{\xi},\zeta)$ in $S\times D_0\times D_1\times D_0$, which is a very large set; therefore, Eq. (2) is difficult to solve for the general case. Fortunately, when $\tau=1$ the Boltzmann equation (2) is simplified:

$$f(\mathbf{x}+\mathbf{c}\Delta t,\mathbf{c},\boldsymbol{\eta},t+\Delta t)=\mathbf{f}^{eq}(\mathbf{x},\mathbf{c},\boldsymbol{\eta},t). \tag{38}$$

Since f^{eq} depends only on the macroscopic fluid density, velocity, and internal energy, the particle distribution function f at $t+\Delta t$ is also determined by them. Therefore, f at $t+\Delta t$ is independent of the particle distribution f at time t. In this way, the requirement for computer memory and computation time is considerably reduced. During the numerical simulations, only mass, momentum, and energy are stored

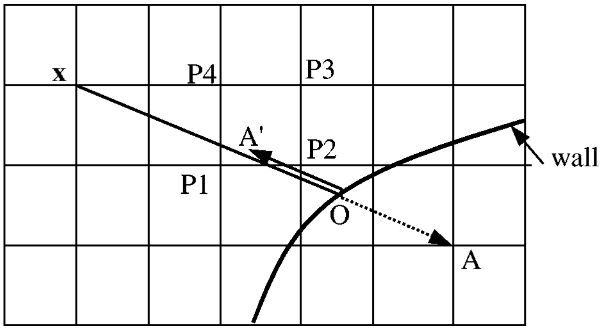

FIG. 3. Bounced-back boundary condition on a curved wall. A particle moving from a node x to a node A in the wall is bounced back precisely from the wall to point A', where $OA=-OA'$. The particle is then redistributed to the neighbor nodes, $P1,P2,\ldots$.

and transported by the particles, and there is no need to store the particle distribution f itself. As soon as we have the mass, momentum, and energy distributions, we know f^{eq} via the definition given by Eq. (14). Then, we can evaluate the new distribution f by Eq. (38) and, in turn, we know mass, momentum, and energy to be transported by the particles for the next time step. Due to the fact that $f^{eq}=0$ for $\mathbf{c}\neq\mathbf{c}_{jk}$, the mass, momentum, and energy transported by the particles from a node $\mathbf{x}$ to $\mathbf{x}+\mathbf{c}_{jk}\Delta t$ are the components of the vector

$$\int_D \boldsymbol{\eta} f^{eq}(\mathbf{x},\mathbf{c}_{jk},\boldsymbol{\eta},t)\,d\boldsymbol{\eta}=\boldsymbol{\phi}_{jk}d_{jk}. \tag{39}$$

In fact, in the simulations we only have to calculate $\boldsymbol{\phi}_{jk}d_{jk}$, for $j=1,\ldots,12$, and $k=1,\ldots,8$ [see Eqs. (30) and (19)–(21) for the definition of $\boldsymbol{\phi}_{jk}$], i.e., 96 particle velocities are treated for each lattice node. The solution obtained in this way is the exact solution of the Boltzmann equation, Eq. (2), when $\tau=1$. And the solution of Eq. (2) is equivalent to a solution of the Navier-Stokes system, Eqs. (34)–(36).

The simulations presented in this paper are all carried out under the conditions $\tau=1$ and $\gamma=1.4$.

B. Boundary conditions

For no-slip walls, the bounce-back boundary condition is second order for walls aligned with the lattice. We successfully simulated viscous flows with flat walls, using the bounce-back boundary condition [21,22,27]. In the case of curved walls, an easy way to deal with curved boundaries is to approximate them by a series of stairs and apply a bounce-back condition. However, this treatment leads to a reduced computational accuracy.

Bounce-back condition. In the present work, we first tested the conventional bounce-back condition at an arbitrary curved solid wall that is not approximated by stairs, as shown in Fig. 3. If a particle is moving from a node x to a node A through the wall, it is bounced back from the wall to point A', where $OA=-OA'$. Usually, point A' is not a lattice node. The particle is then redistributed to the neighbor nodes, $P1,P2,\ldots$, of point A' with portions defined in a similar manner to Eq. (16).

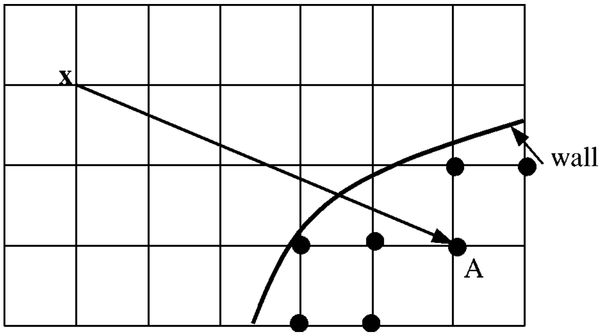

FIG. 4. Extrapolated boundary condition. A particle moving from a node $\mathbf{x}$ to a node A in the wall is disregarded. Its effect is taken into account by the auxiliary nodes inside the wall (marked by ●) that are involved in the computation in the same way an ordinary node would. The macroscopic variables on the auxiliary nodes are extrapolated from the values in the computational domain.

Because the particle is bounced back precisely from the curved wall, the boundary condition is more accurate than that approximated by stairs. However, simulations demonstrate instabilities near the wall, as shown in the following example.

A flow over a NACA0012 airfoil was simulated for Mach number 0.5 and Reynolds number 600 at a zero angle of attack. The pressures are shown in Fig. 5. In Fig. 5(a), we see a significant noise at the wall and an asymmetry distribution. Figure 5(b) shows the pressure distribution along upper wall surface.

This noise is due to the nonuniformity of the lattice at the wall. To resolve this problem we propose the following improved boundary condition for curved wall.

Extrapolated boundary condition. In order to keep the uniformity of the lattice at the wall, we introduce auxiliary nodes inside the wall, as shown in Fig. 4. The macroscopic variables on the auxiliary nodes are extrapolated from the values in the computational domain. The following conditions have to be satisfied at the solid wall during the extrapolation:

$$\mathbf{v}=\mathbf{0},\quad \frac{\partial\rho}{\partial n}=0,\quad \frac{\partial e}{\partial n}=0, \tag{40}$$

where n is the normal direction of the wall surface. Any particle moving from a node in the computational domain to a node inside the wall is disregarded. Its effect is taken into account by the nodes inside the wall (marked by ● in Fig. 4) that are involved in the computation such as ordinary nodes.

Unlike the standard LB models in which boundary conditions must be given in terms of particle distribution function f [30–33], the present scheme deals only with the equilibrium distribution f^{eq} that depends only on the macroscopic fluid density, velocity, and internal energy. The boundary conditions imposed on macroscopic variables, such as those given in Eq. (40), can be directly implemented in the scheme.

The same flow over a NACA0012 airfoil was simulated for Mach number 0.5 and Reynolds number 600 using the

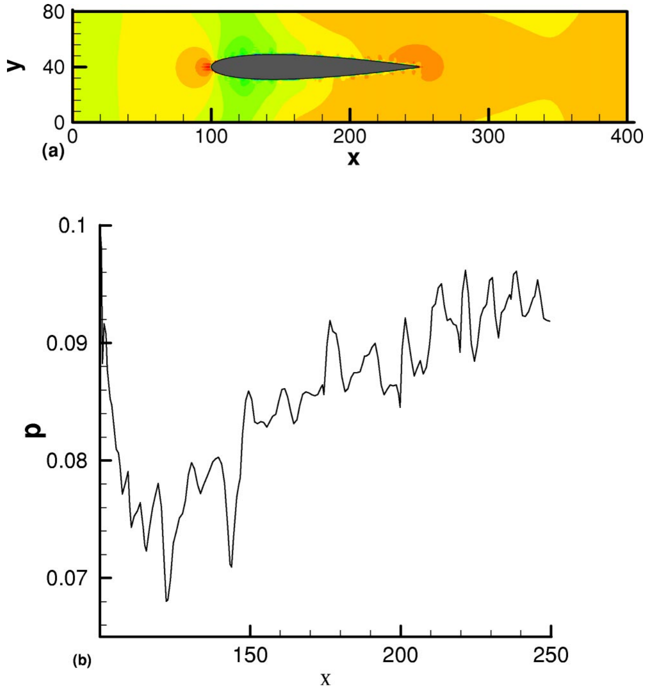

FIG. 5. (Color online) Pressure distribution of a flow over NACA0012 using bounced-back wall condition for Mach number 0.5 and Reynolds number 600. (a) Pressure on a lattice of 400 ×80×3 nodes. Periodical boundary condition is imposed in z direction. (b) Dimensionless pressure on the wall. The significant noise at the wall is due to the nonuniformity of the lattice at the wall.

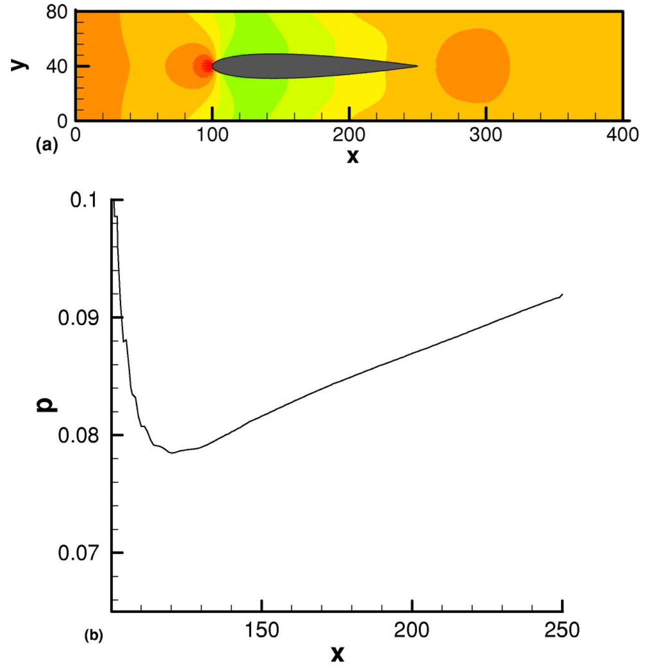

FIG. 6. (Color online) Pressure distribution of a flow over NACA0012 for the same Mach number and Reynolds number as in Fig. 5, but using extrapolated wall condition. (a) Pressure, (b) dimensionless pressure on the wall. The distribution is more symmetric and the noise at the wall is much less than that in Fig. 5.

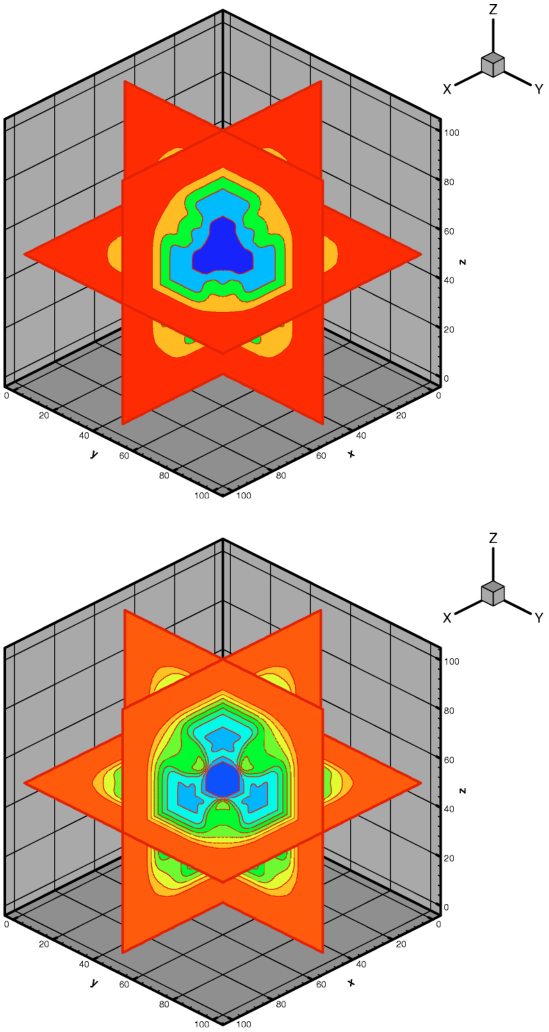

FIG. 7. (Color online) Three-dimensional shock wave propagation on a lattice of 100×100×100 at $t=20$. (a) Density, (b) pressure. At the initial time, the velocity is zero, the pressure and density inside a 50×50×50 box in the center is one-tenth and one-eighth of that outside the box, respectively. The shocks move toward the center from four directions and they interact at the corners.

new boundary condition. The pressure contours are shown in Fig. 6. In Fig. 6(a), we see that the pressure distribution is more symmetric and the noise at the wall is much less than the previous case, and line plots of the pressure distribution along the wall surface presented in Figs. 5(b) and 6(b) further illustrate this point.

The comparison of these two cases indicates that the extrapolated boundary condition is more stable. All the simulations presented in the following section were carried out, using the auxiliary node boundary condition.

IV. NUMERICAL SIMULATIONS

Validation cases for the three-dimensional model are presented in this section. Cases A, B, and C are inviscid flow simulations. When the viscous terms and the diffusion terms of the right-hand sides of Eqs. (35) and (36) are regarded as discretization errors, Eqs. (34)–(36) become the inviscid Eu-

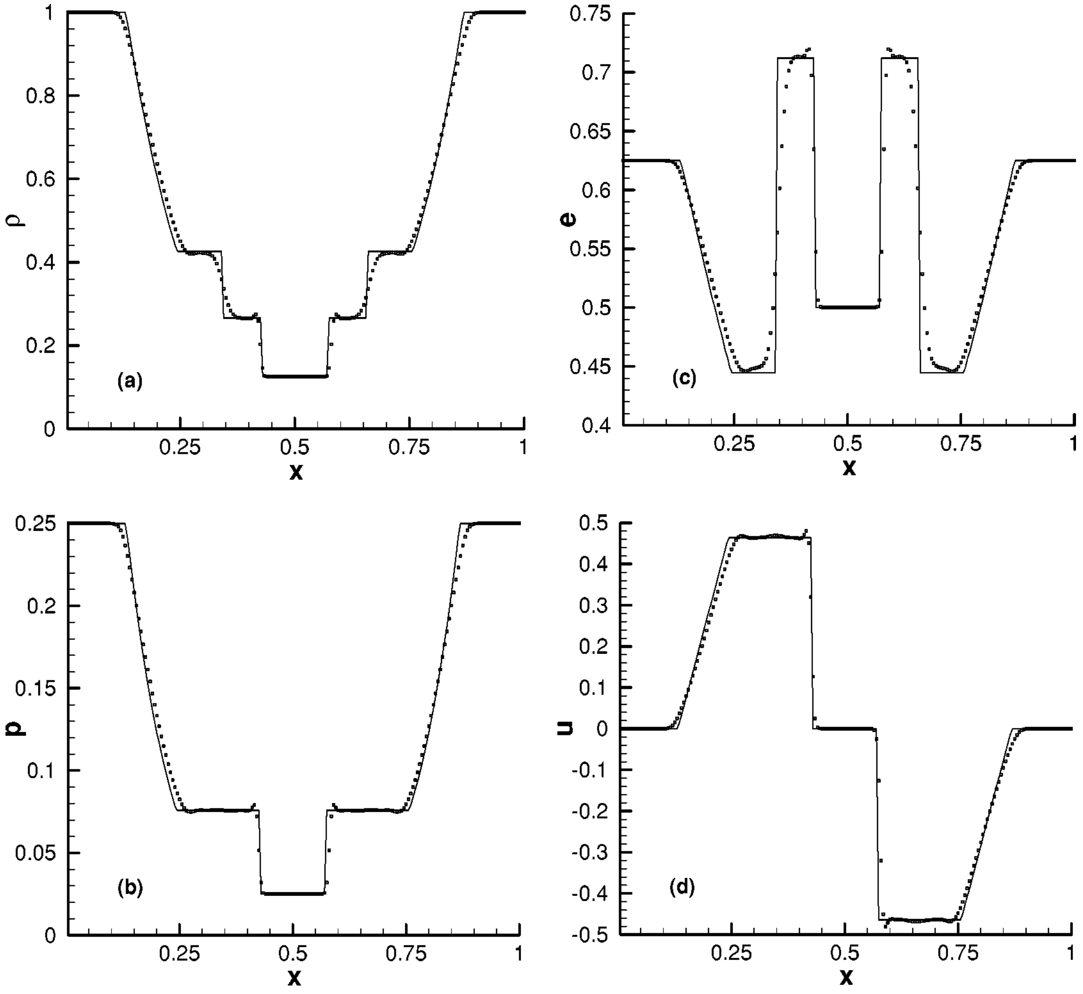

FIG. 8. Three-dimensional shock wave propagation at $t = 20$. (a) Density, (b) pressure, (c) internal energy, (d) velocity distribution vs x at $y = 0.5$ and $z = 0.5$. Solid lines are exact solutions. The variables are dimensionless.

ler system, where the artificial viscosity and diffusivity are of order $(\tau - 1/2)l^2/\Delta t$, with l the lattice length and Δt the time step. The slip wall boundary condition is applied at the solid surface for these inviscid flow simulations.

Cases $D-F$ deal with viscous flows and the no-slip condition proposed in the preceding section is applied at the walls.

Test case A is a genuine three-dimensional flow, while the other cases are two-dimensional flows solved, using a three-dimensional flow solver. These two-dimensional flows are chosen for validation purposes, for lack of better three-dimensional data for comparison. In the two-dimensional flow cases, periodical boundary conditions were used in z direction.

All the variables are dimensionless. The lattice length l and the time step Δt are taken as characteristic length and time scales, respectively.

A. Three-dimensional shock-wave propagation

As a first test case for the three-dimensional model, flow caused by a cubic-shock box is simulated. The simulation was carried out on a lattice of $100\times 100\times 100$ nodes under the conditions $\tau = 1$ and $\gamma = 1.4$. At the initial time, the velocity is zero, the pressure and the density inside a $50\times 50\times 50$ box in the center of the computational domain is one-tenth and one-eighth of that outside the box, respectively, i.e., $(\rho, p, \mathbf{v}) = (0.125, 0.025, \mathbf{0})$ for $25 < x < 75$, $25 < y < 75$, and $25 < z < 75$; $(\rho, p, \mathbf{v}) = (1, 0.25, \mathbf{0})$ for others, where $\mathbf{0}$ represents a zero vector.

Figure 7 shows the density and the pressure contours at $t = 20$. The shocks move toward the center from four directions and they interact at the corners.

At the central line defined by $y = 0.5$ and $z = 0.5$, where the shocks have not yet interacted with each other at time $t = 20$, the solution similar to shock-tube problems [38] can be compared with the exact Riemann solutions. Here, the computational domain $100\times 100\times 100$ is normalized to $1\times 1\times 1$. Figure 8 compares the density, pressure, internal energy, and velocity distributions versus x, at the central line with the exact Riemann solutions. The points and the solid lines represent the numerical and the exact solutions, respectively. The present solution and the exact Riemann solution agree well. The shocks are captured within only three points in the present solution.

The simulation takes 50 s per iteration on a Pentium II 366 computer for the case of $100\times100\times100$ nodes, i.e., 5.0×10^{-5} seconds per node. The computation time is proportional to the number of nodes.

The CPU time reported here is for the three-dimensional Navier-Stokes solver. For a two-dimensional projection [27] of this model, the CPU time required is 1.44×10^{-5} s per node, i.e., 3.47 times faster. Moreover, in the cases where only an Euler solution is of interest, the correction terms χ_{ik} in Eqs. (19)–(21) may be neglected, and m_{jk}, $\boldsymbol{\xi}_{jk}$, and ζ_{jk} become independent of the index *k*; as a result, the computer code is at least two times faster than the Navier-Stokes case. Neglecting terms χ_{jk} introduces additional terms in the form of $\mathbf{v}_k'\mathbf{v}_k'$ in the macroscopic conservation equations, which influence the viscous and thermal conductive terms of Navier-Stokes equation and energy equation but do not affect the accuracy of an Euler solution [20].

B. Double Mach reflection (Ma=10)

This example is a double Mach reflection [39], i.e., a normal shock of Mach number 10 passing a 30° wedge. This example was also simulated in our previous paper [26] where the wall was aligned with the lattice and particles were reflected on the wall in the same way as a beam of light reflected on a mirror.

In the present case, the wall is inclined and a slip boundary condition is imposed for macroscopic variables at the wall, i.e.,

$$v_n=0, \quad \frac{\partial v_t}{\partial n}=0, \quad \frac{\partial \rho}{\partial n}=0, \quad \frac{\partial e}{\partial n}=0, \tag{41}$$

where v_n and v_t are the normal velocity and the tangential velocity of the fluid at the wall, respectively.

Figure 9 shows the pressure, density, and entropy distributions. They agree with the results of other numerical methods [15,39] and our previous results [26].

C. Propagation of a shock wave at a Mach number 1.09

This example is to validate the model for the flow regime with Mach number ~1. This is a one-dimensional shock-wave case. Periodic boundary conditions are imposed in *y* and *z* directions on a lattice of $200\times3\times3$. The following initial condition is imposed: $t=0$: $(\rho_2,p_2,u_2)=(1.152,0.0544,0.03595)$ for $0\leqslant x<25$ and $(\rho_1,p_1,u_1)=(1.0,0.0446,0.0)$ for $25\leqslant x\leqslant200$, corresponding to a shock wave Mach number 1.09. The shock moves at a speed of 0.2725 grids per iteration. Figure 10 shows the density, pressure, internal energy, and velocity distributions from $t=0$ to 500 at an interval of 125. The solid lines are exact solutions. The results agree well with the theoretical values. The pressure ratio and the density ratio are 1.2194 and 1.152, respectively. If we look at the simulation in a coordinate moving with the shock, the Mach number is 1.09 before the shock and 0.9197 after the shock.

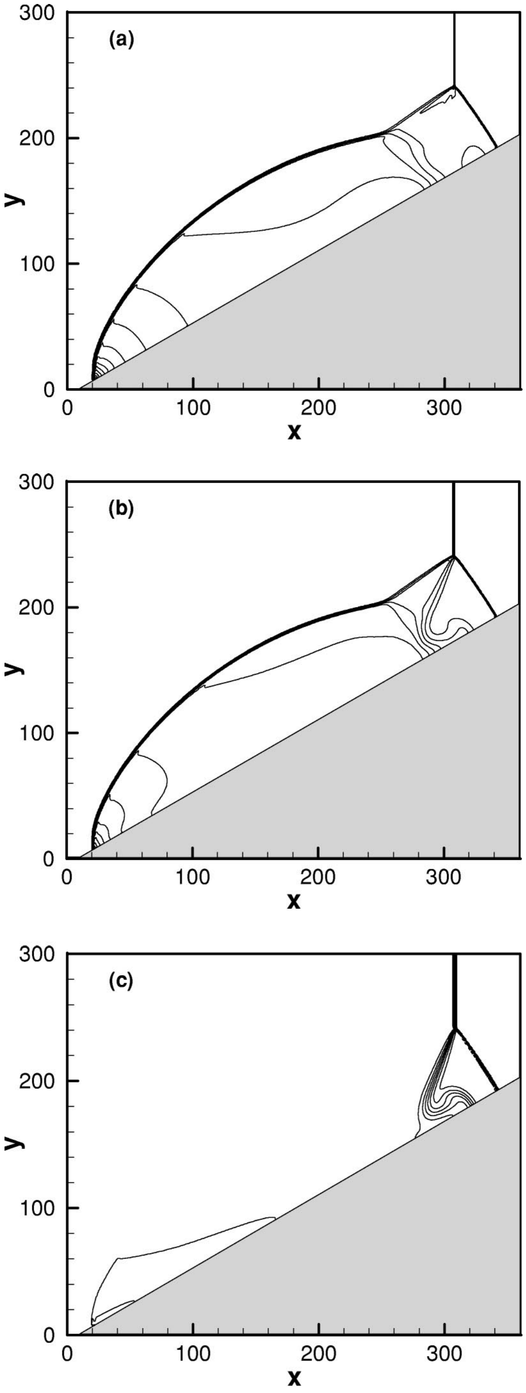

FIG. 9. Double Mach reflection (Ma=10) over a 30° wedge on a lattice $360\times300\times3$. (a) Pressure, (b) density, (c) entropy.

D. Boundary layer flow

A boundary layer flow over a flat plate is simulated using the current scheme. On the wall, the no-slip boundary condition given by Eq. (40) was employed. The plate was placed at an arbitrary angle with respect to the grid, i.e., the boundary does not coincide with any grid line, to test the capability of the scheme of dealing with complex geometry. The contours of the fluid velocity parallel to the plate are plotted in Fig. 11. The dimensionless velocity u/U versus η

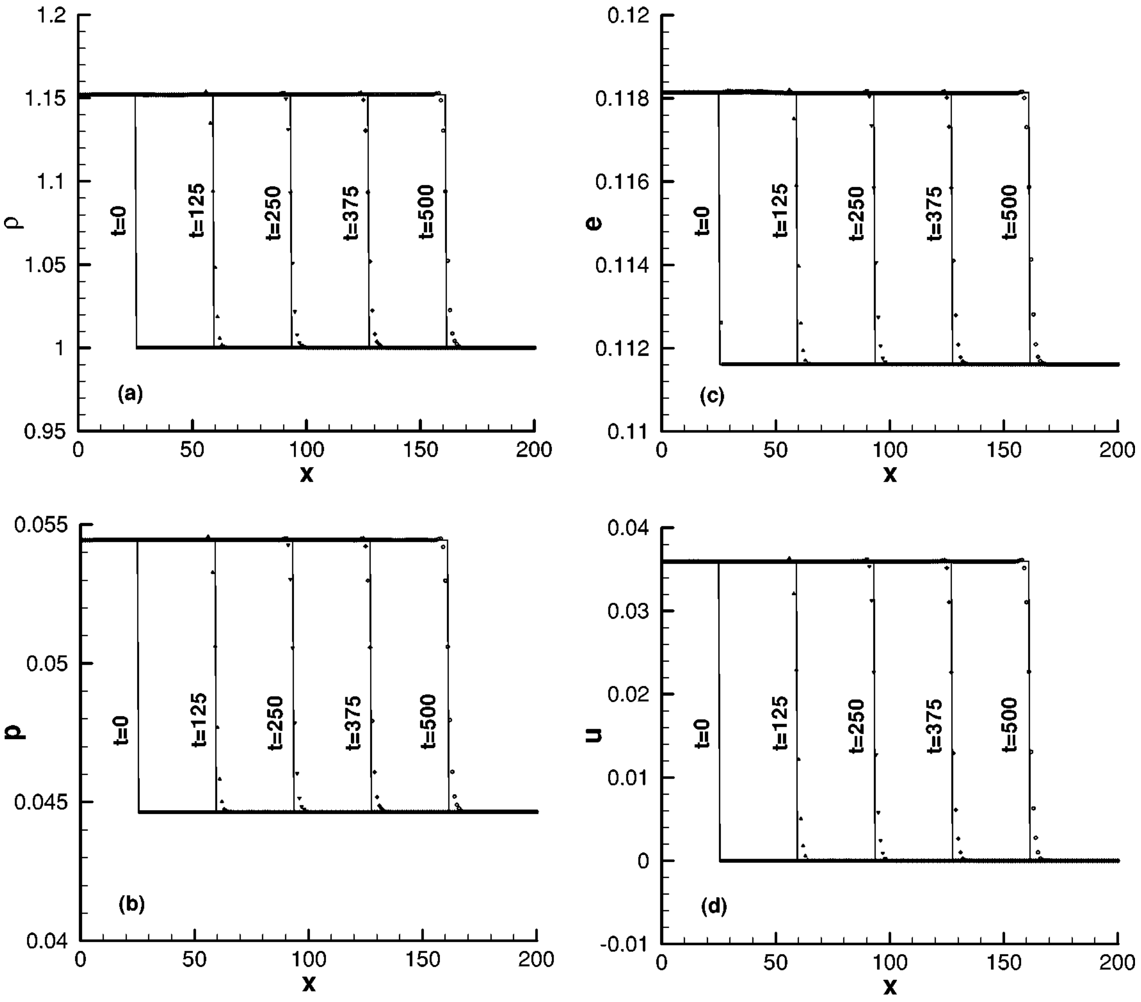

FIG. 10. Propagation of a shock wave at a Mach number 1.09. (a) Density, (b) pressure, (c) internal energy, and (d) velocity distributions from $t=0$ to 500 at an interval of 125. Solid lines are exact solutions. The shock moves at speed of 0.2725. The variables are dimensionless.

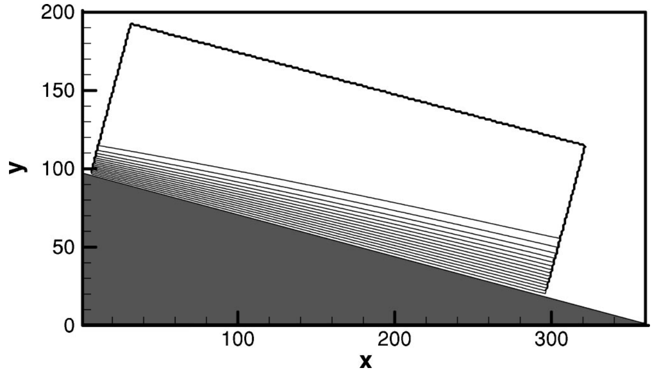

FIG. 11. Boundary layer flow. Velocity contours. The plate was placed at an arbitrary angle to justify the boundary condition that does not coincide with any grid line.

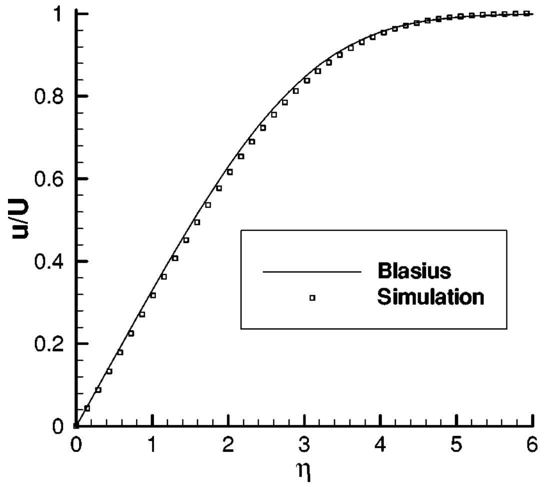

FIG. 12. Boundary layer flow. Comparison to the Blasius solution. The dimensionless velocity u/U vs $\eta(=y\sqrt{U/(\nu x)})$, the dimensionless distance.

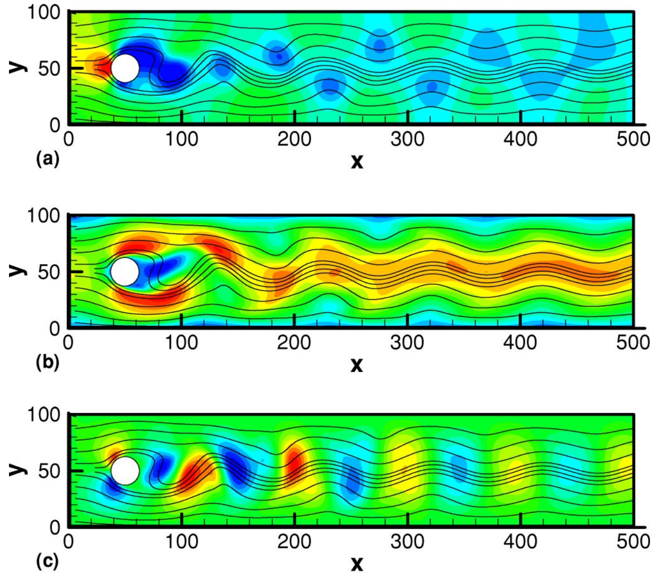

FIG. 13. (Color online) Flow over a cylinder. (a) Pressure, (b) u, (c) v.

$(=y\sqrt{U/(\nu x)}, \nu=\mu/\rho)$, the dimensionless distance from the wall, is shown in Fig. 12, where the square symbols are the numerical results, and the solid line represents the Blasius solution. Good agreement is observed between the present numerical results and the Blasius solution.

E. Flow over a cylinder

This example is a flow over a cylinder inside a channel for which benchmark results were reported in Ref. [40]. The simulation was performed on a lattice of 500×100×3 nodes with a cylinder of a diameter 20 located at $x=50$ and $y=50$. The Reynolds number based on the averaged inlet velocity and the diameter of the cylinder is 100, at which the flow is periodical in time and there is a vortex street downstream of the cylinder. Figure 13 shows an instant distribution of the pressure and velocity components u and v. The normalized lift force coefficient versus time is plotted in Fig.

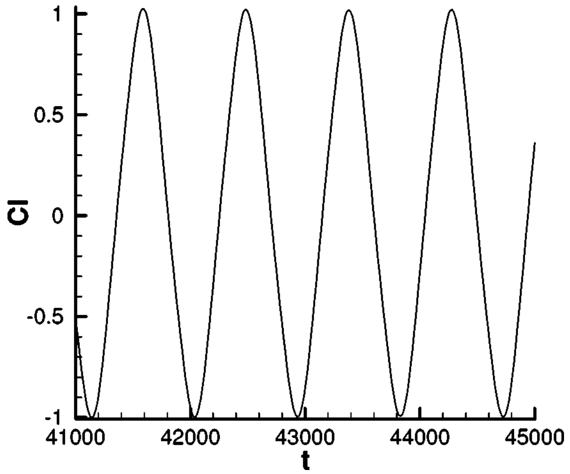

FIG. 14. Flow over a cylinder. Normalized lift coefficient Cl vs t. The period is 900, Strouhal number St=0.314.

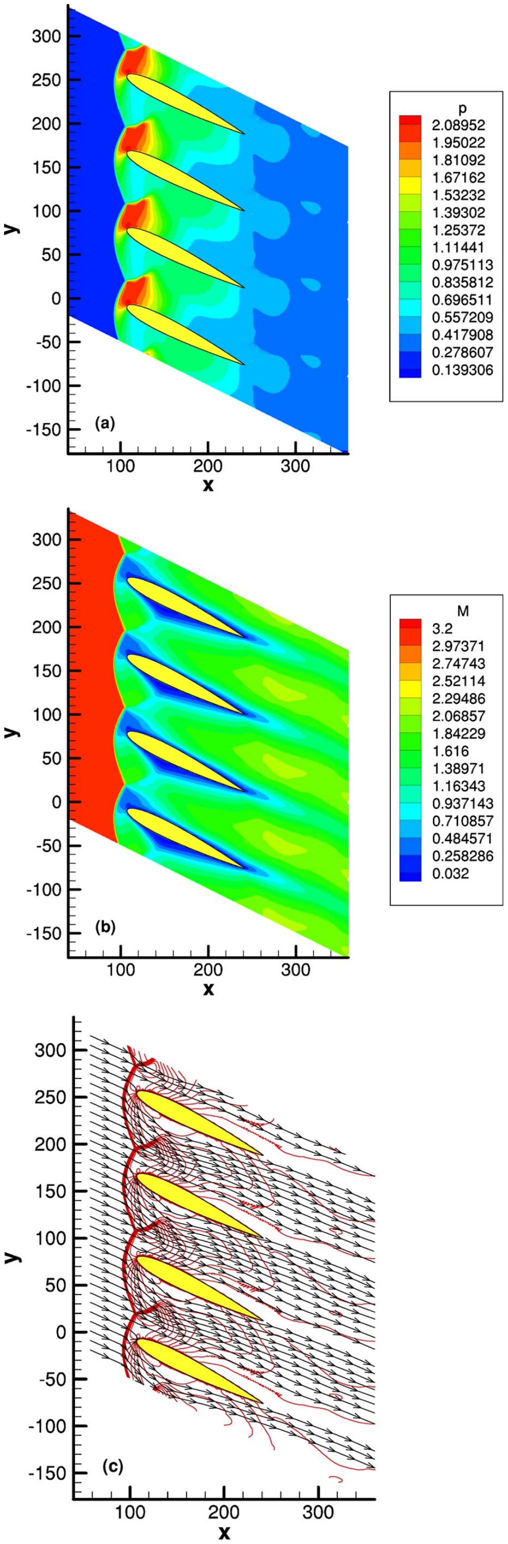

FIG. 15. (Color online) Transonic flow with a inflow Mach number 3.0 over a NACA0012 cascade on a lattice of 400×80×3. (a) Pressure, (b) Mach number, (c) density contours and streamlines. Detached oblique shocks are formed in front of blades. Across these shocks, the pressure increases and the Mach number decreases. The streamlines show the flow deviations crossing the shocks and formation of a slight boundary-layer separation followed by a reattachment at the lower surface of the blade.

14. The periodicity is 900 Hz, which corresponds to a Strouhal number of 0.314. The predicted Strouhal number agrees well with the values of 0.295–0.310 given in Ref. [40].

F. Transonic flow over a NACA0012 cascade

For the cascade simulation, the blade shape is taken from NACA0012 airfoil. At the upstream, constant boundary conditions are imposed for density, pressure, and velocity: $(\rho,p,u)=(1.0,0.25/1.4,1.5)$. The angle of attack is zero, and inflow Mach number is 3. Periodical boundary conditions are imposed on the upper and lower boundaries. Initial condition is set up to be the same as the upstream boundary condition. Figure 15 shows the lattice Boltzmann solution for pressure, Mach number, and streamlines through the cascade. Four blades are plotted. A lattice of $400\times80\times3$ is used for each blade field. Detached oblique shocks are formed in front of the blades. Across these shocks pressure, density, and internal energy increase, while the Mach number decreases. The flow changes direction crossing the shocks and forms a slight boundary-layer separation and reattachment at the lower surface of the blade due to shock-boundary layer interaction, which can be observed from the streamline plots.

V. CONCLUSIONS

A three-dimensional compressible LB model on a square lattice is proposed in the present paper. A large particle-velocity set is used to enable the simulation of high Mach number flows. Meanwhile, in order to make the computation more tractable, a small support set for the equilibrium distribution is employed. This model can handle flows over a wide range of Mach numbers and capture jumps through shock waves. Because of the simple form of the equilibrium distribution, the fourth-order velocity tensors are not involved in the calculations. Unlike the standard lattice Boltzmann model on square lattices, no special treatment is required for the homogeneity of fourth-order velocity tensors. Therefore, the Navier-Stokes equations and energy equation were recovered with only six symmetric particle velocity directions. The second-order discretization errors in velocity have been eliminated to improve the accuracy in viscous flow simulations. The model is valid for both viscous and inviscid compressible flows with or without shocks.

The present scheme requires only the equilibrium distribution that depends on fluid density, velocity, and internal energy. We proposed a boundary condition based on an extrapolation of the macroscopic variables for curved walls. This boundary condition treatment is self-consistent, easy to implement, and suitable for both slip wall and no-slip wall boundary conditions. Moreover, it can be easily extended to complex flows with moving walls, mass injection from the walls, and heat exchange with the walls.

To verify the scheme for inviscid flows, we have successfully simulated a three-dimensional shock-wave propagation in a box, a normal shock of Mach number 10 over a wedge, and a one-dimensional shock of Mach number 1.09. As an application to viscous flows, we have simulated a flat plate boundary layer flow, flow over a cylinder, and a transonic flow over a NACA0012 cascade.

ACKNOWLEDGMENT

This work was funded by NASA Glenn Research Center under Grant No. NAG3-2399.

---

[1] H. Chen, S. Chen, and W. Matthaeus, Phys. Rev. A **45**, R5339 (1992).

[2] Y. H. Qian, D. d'Humières, and P. Lallemand, Europhys. Lett. **17**, 479 (1992).

[3] F. Higuera, S. Succi, and R. Benzi, Europhys. Lett. **9**, 345 (1989); R. Benzi, S. Succi, and M. Vergassola, Phys. Rep. **222**, 145 (1992).

[4] U. Frisch, B. Hasslacher, and Y. Pomeau, Phys. Rev. Lett. **56**, 1505 (1986).

[5] U. Frisch, D. d'Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, and J. P. Rivet, Complex Syst. **1**, 649 (1987).

[6] G. McNamara and G. Zanetti, Phys. Rev. Lett. **61**, 2332 (1988).

[7] F. Higuera and J. Jimenez, Europhys. Lett. **9**, 663 (1989).

[8] P. Bhatnagar, E. Gross, and M. Krook, Phys. Rev. **94**, 511 (1954).

[9] S. Chen and G. D. Doolen, Annu. Rev. Fluid Mech. **30**, 329 (1998).

[10] Y. H. Qian, S. Succi, and S. A. Orszag, *Annual Reviews of Computational Physics III*, edited by D. Stauffer (World Scientific, Singapore, 1995), p. 195.

[11] F. J. Alexander, H. Chen, and G. D. Doolen, Phys. Rev. A **46**, 1967 (1992).

[12] H. Yu and K. Zhao, Phys. Rev. E **61**, 3867 (2000).

[13] B. Palmer and D. Rector, J. Comput. Phys. **161**, 1 (2000).

[14] K. Xu, J. Stat. Phys. **81**, 147 (1995).

[15] K. Xu, L. Martinelli, and A. Jameson, J. Comput. Phys. **120**, 48 (1995).

[16] B. Nadiga and D. Pullin, J. Comput. Phys. **112**, 162 (1994).

[17] B. Nadiga, J. Stat. Phys. **81**, 129 (1995).

[18] J. Huang *et al.*, Int J. Mod. Phys. C **8**, 827 (1997).

[19] X. He and L. Luo, Phys. Rev. E **56**, 6811 (1997).

[20] C. H. Sun, Phys. Rev. E **58**, 7283 (1998).

[21] C. H. Sun, Chin. Phys. Lett. **17**, 209 (2000).

[22] C. H. Sun, Phys. Rev. E **61**, 2645 (2000).

[23] C. H. Sun, J. Comput. Phys. **161**, 70 (2000).

[24] C. H. Sun, Acta Mech. Sin. **16**, 289–300 (2000).

[25] A. Hsu, C. H. Sun, and A. Ecer, in *Parallel Computational Fluid Dynamics*, edited by C. B. Jenssen *et al.* (Elsevier Science, Amsterdam, 2001), p. 375–382.

[26] A. Hsu, C. H. Sun, C. Wang, A. Ecer, and I. Lopezb, in *Parallel Computational Fluid Dynamics*, edited by P. Wilders *et al.* (Elsevier Science, Amsterdam, 2002), p. 175–182.

[27] C. H. Sun, Chin. Phys. Lett. **17**, 743 (2000).

[28] I. Ginzbourg and P. M. Adler, J. Phys. II **4**, 191 (1994).

[29] D. Noble, S. Chen, J. Georgiadis, and R. Buckius, Phys. Fluids **7**, 203 (1995).

[30] S. Chen and D. Martinez, Phys. Fluids **8**, 2527 (1996).
[31] I. Ginzbourg and D. d'Humière, J. Stat. Phys. **84**, 927 (1996).
[32] R. Mei, L. Luo, and W. Shyy, J. Comput. Phys. **155**, 307 (1999).
[33] R. Mei, W. Shyy, D. Yu, and L. Luo, J. Comput. Phys. **161**, 680 (2000).
[34] D. Bernardin, O. Sero-Guillaume, and C. H. Sun, Physica D **47**, 169 (1991).
[35] C. H. Sun, Ph.D. Dissertation, LEMTA, L'Institut National Polytechnique de Lorrain, Nancy, France, 1993 (unpublished).
[36] R. Sanders and K. Prendergast, Astrophys. J. **188**, 489 (1974).
[37] K. Xu and L. Luo, ICASE Report No. 99-10 (unpublished).
[38] G. Sod, J. Comput. Phys. **27**, 1 (1978).
[39] P. Colella, J. Comput. Phys. **87**, 171 (1990).
[40] M. Schäfer and S. Turek, in *Notes in Numerical Fluid Mechanics*, edited by W. Hackbusch and G. Wittum (Vieweg Verlag, Braunschweig, 1996), Vol. 52, pp. 547–566.