Journal of Computational Physics 229 (2010) 6732–6743

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A semi-implicit lattice method for simulating flow

David R. Rector *, Mark L. Stewart

Pacific Northwest National Laboratory, Energy and Environmental Directorate, Richland, WA 99352, United States

ARTICLE INFO

Article history:
Received 20 July 2009
Received in revised form 15 March 2010
Accepted 20 May 2010
Available online 26 May 2010

Keywords:
Lattice-Boltzmann method
Incompressible Navier–Stokes equation
Lid-driven cavity
Unsteady Poiseuille flow
Oscillating plate

ABSTRACT

We propose a new semi-implicit lattice numerical method for modeling fluid flow that depends only on local primitive variable information (density, pressure, velocity) and not on relaxed upstream distribution function values. This method has the potential for reducing parallel processor communication and permitting larger time steps than the lattice-Boltzmann method. Several benchmark problems are solved to demonstrate the accuracy of the method.

1. Introduction

Computational methods based on the lattice-Boltzmann equation have attracted much attention in recent years, especially for applications involving liquid–vapor multiphase flow [1], flow in complex geometries such as porous media [2] and turbulent aerodynamics [3]. Advantages of the lattice-Boltzmann method over other flow simulation methods include the ability to model phase interfaces and efficient scalability on massively parallel computers.

The lattice-Boltzmann equation describes the evolution of the single particle distribution function, f_i, as a function of time and space. The most widely used version is the Bhatnagar–Gross–Krook (BGK) approximation, where the distribution function values are continuously relaxed toward the local equilibrium values while streaming. The relaxation time, τ, is related to the fluid kinematic viscosity.

Unfortunately, certain disadvantages to the lattice-Boltzmann method limit its usefulness as a general purpose flow simulation tool. The lattice-Boltzmann method is explicit in time, and the time step size is limited to a fraction of the Courant limit. Sukop and Thorn recommend a maximum velocity of one tenth the lattice characteristic speed [4]. In addition, values of the distribution function, f_i, must be stored and communicated between computer processors. For a D3Q19 model, values for 18 vector directions must be communicated. If the information to be communicated could be reduced, the efficiency of the algorithm on parallel computers would greatly improve. Finally, the lattice-Boltzmann algorithm uses a relaxation parameter, τ, which is the rate at which the incoming distribution function is relaxed toward the local equilibrium distribution. For very small (near 0.5) or very large values, the accuracy and stability of the lattice-Boltzmann method degrades.

Several attempts have been made to address some of these limitations. For example, an implicit formulation of the lattice-Boltzmann method was developed by Sankaranarayanan et al. [5] to permit larger time steps in the simulation of bubbly suspensions. An expression is derived for each element of the particle velocity distribution function using the trapezoidal rule, so that the rate of relaxation is based on both the old-time and new-time values for f_i and $f_i^{(eq)}$. As a result, the method

* Corresponding author. Tel.: +1 509 372 4530.
E-mail address: david.rector@pnl.gov (D.R. Rector).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.05.020

could require several iterations for each vector direction during each time step, partially offsetting the computational savings resulting from larger time steps.

A lattice kinetics scheme was developed by Inamuro [6] to eliminate the need to store and transmit the elements of the particle velocity distribution function. Instead, the distribution function values are calculated using local primitive variables (velocities and pressure or density). This method has been extended to incompressible viscous flows [7] and liquid–vapor multiphase flows [8] but has not yet gained wide acceptance.

A major advance in recent years has been the development of the multiple-relaxation-time (MRT) lattice-Boltzmann method [9,10]. With the simpler BGK model, a single parameter, τ, is used to relax the particle distribution functions in all lattice directions (f_i) toward their equilibrium values, $f_i^{(eq)}$. It has been observed that the single relaxation time can cause shifts in the effective boundary locations as viscosity changes. This can lead to significant errors in flow and pressure field predictions in cases where wall interactions are dominant, as in flow through porous media [11]. The MRT method, by contrast, includes a matrix of relaxation parameters which may be individually selected in order to reduce un-physical viscosity dependencies [12]. It has also been demonstrated that he MRT method can be more numerically stable than the lattice BGK approach [9]. It has been estimated that equivalent models solved with the MRT method take approximately 10–20% more processor time to solve than lattice BGK [12].

Since the accuracy of the lattice-Boltzmann method with the commonly used BGK collision operator has been shown to depend upon the relaxation parameter, many investigators have limited their work with the lattice-Boltzmann method to models with $\tau = 1$, which fixes the relationship between viscosity, time step, and lattice spacing. The lattice-Boltzmann method is also limited to velocities that are small compared to the lattice speed of sound, which sets a maximum ratio between lattice spacing and time step. These two constraints placed on three key system parameters sometimes make it difficult to apply the lattice-Boltzmann method to real-world problems, where lattice spacing is set by system geometry and relevant length scale, and time steps must be chosen to adequately resolve system dynamics or allow reasonably quick steady-state solutions.

These limitations help explain why a large fraction of investigations using the lattice-Boltzmann method have resorted to dimensional analysis in order to relate simulations to reality. This requires knowledge of the key dimensionless groups for a given problem and introduces occasions for error and misinterpretation. Clearly, a simulation tool which allows dimensional simulations over a wide range of conditions is a much more desirable solution. As described above, multiple relaxation techniques are a mature approach for addressing some of these limits. The implicit lattice kinetic method described in this paper offers another alternative, which may yield another set of advantages, such as increased parallelization, simple boundary conditions, and ease of simulation state archival and retrieval.

A primary advantage of lattice-based computational fluid dynamics (CFD) methods is their inherent parallelism. For this reason, these methods hold enormous promise for simulation of flow and transport problems of unprecedented complexity by efficiently using massively parallel computational resources. In addition to the steady march in the performance of high-end supercomputers, the decreasing cost of clusters incorporating commodity multi-core processors is making it practical to run programs with hundreds or even thousands of parallel processes for an ever wider range of applications. Rather than attempting to solve a global pressure field, as in many other approaches to CFD, lattice-based techniques evolve local pressure solely as the result of interactions with neighboring lattice sites. Therefore, subdivided domains only require communication across block boundaries among a limited number of close neighbors. In the case of the lattice-Boltzmann method, the single particle distribution functions for all mobile lattice directions (18 in the case of the D3Q19 lattice) must be streamed to neighboring lattice sites. In the lattice kinetic approach, however, only the primitive variables of pressure and velocity are stored and communicated. This substantial decrease in communications overhead should translate directly into enhanced performance in parallel applications.

The next section provides a brief description of the lattice-Boltzmann method followed by a presentation of the proposed implicit lattice kinetics method. We then demonstrate the recovery of the Navier–Stokes equations from the lattice kinetics equations. The method is then applied to a lid-driven cavity, an infinite oscillating plate, and transient 2D channel flow to assess the accuracy and stability of the new method.

2. Method

The implicit lattice kinetics method was developed as an alternative to the lattice-Boltzmann equation method, which is described here briefly for comparison.

2.1. Lattice-Boltzmann

The single particle distribution function $f(\mathbf{x},\mathbf{v},t)$ is the probability of finding particles at position $\mathbf{x}$ with velocity $\mathbf{v}$ at time t. The Boltzmann equation (shown here with the simple BGK collision operator) describes the evolution of the distribution function

$$\frac{\partial f}{\partial t} + \mathbf{v}\cdot\nabla f = -\frac{f - f^{eq}}{\tau} \tag{1}$$

where τ is the relaxation time and f^{eq} is the equilibrium Maxwell distribution. To solve this equation on a computer, the spatial domain is discretized into a finite number of lattice sites, where each lattice site has values for density, pressure, flow, etc. The single particle distribution function, which describes the probability of a particle traveling along a particular direction and speed, is discretized to form a finite set of displacement vectors connecting each lattice site to adjacent sites.

The lattice-Boltzmann equation describes the evolution of the discretized particle distribution function, $f_i(\mathbf{x},t)$, along direction vector $\mathbf{i}$ as a function of time. Fig. 1 shows the lattice directions for the commonly used two-dimensional D2Q9 lattice, while Fig. 2 shows the lattice directions used for the three-dimensional D3Q19 lattice.

The new-time distribution function is given by the equation

$$f_i(\mathbf{x}+\mathbf{e_i}, t+\Delta t) - f_i(\mathbf{x},t) = -\frac{1}{\tau}\left[f_i(\mathbf{x},t) - f_i^{eq}(\mathbf{x},t)\right] \tag{2}$$

where τ is a linear relaxation parameter and f^{eq} is the local equilibrium distribution. The relaxation parameter is a function of the local kinematic viscosity, v

$$\tau = \frac{1}{2} + \frac{3v}{c^2\Delta t} \tag{3}$$

where c is the reference lattice speed, $c = \Delta x/\Delta t$. The local equilibrium is expressed in the form of a quadratic expansion of the Maxwellian distribution [13]

$$f_i^{eq} = w_i\left[\rho + \rho\left(\frac{3\mathbf{e}_i\cdot\mathbf{u}}{c^2} + \frac{9(\mathbf{e}_i\cdot\mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}\cdot\mathbf{u}}{2c^2}\right)\right] \tag{4}$$

where ρ is the density, $\mathbf{e}_i$ is the lattice direction vector, $\mathbf{u}$ is the fluid velocity vector and the weight coefficients, w_i for a D3Q19 system are $w_0 = 1/3$, $w_{1-6} = 1/18$ and $w_{7-18} = 1/36$ [14]. The expression for an incompressible flow field is similar except that the first term is stated in terms of pressure

$$f_i^{eq} = w_i\left[\frac{p}{c_s^2} + \rho\left(\frac{3\mathbf{e}_i\cdot\mathbf{u}}{c^2} + \frac{9(\mathbf{e}_i\cdot\mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}\cdot\mathbf{u}}{2c^2}\right)\right] \tag{5}$$

where p is the pressure and $c_s^2 = c^2/3$ is the lattice sound speed [15].

Density (or pressure) and momentum are determined by summing the incoming distribution function values

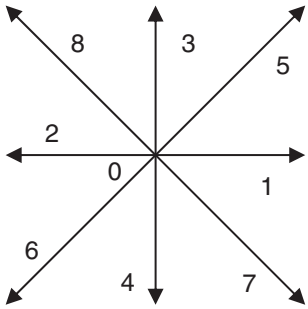

Fig. 1. D2Q9 lattice.

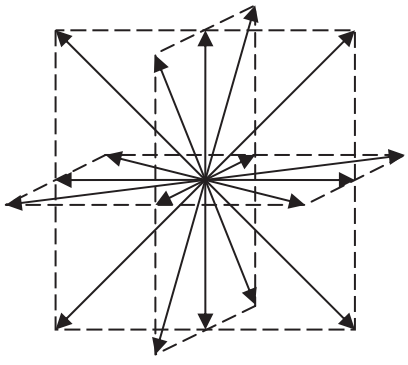

Fig. 2. D3Q19 lattice.

$$\rho = \sum_i f_i \quad p = c_s^2 \sum_i f_i \quad \rho \mathbf{u} = \sum_i \mathbf{e_i} f_i \tag{6}$$

Note that density and momentum are strictly conserved.

2.2. Implicit lattice kinetics

We wish to create a lattice method that depends solely on the local primitive variables (density, or pressure, and momentum). We accomplish this by calculating the incoming distribution function using a modified form of the equilibrium distribution function expression. Different distribution functions expressions are used for the density/pressure and momentum/velocity equations

$$f_{i,k} = w_i \left[\rho + A_k \rho \frac{3\mathbf{e}_i \cdot \mathbf{u}}{c^2} + \rho \left(\frac{9(\mathbf{e}_i \cdot \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u} \cdot \mathbf{u}}{2c^2} \right) \right] \tag{7}$$

$$A_\rho = 1 \quad A_u = \frac{6\nu}{c^2 \Delta t}$$

where $k = \rho$, u indicates density/pressure or momentum/velocity, respectively. The incompressible (pseudo-compressible) form of the expression is

$$f_{i,k} = w_i \left[\frac{p}{c_s^2} + A_k \rho \frac{3\mathbf{e}_i \cdot \mathbf{u}}{c^2} + \rho \left(\frac{9(\mathbf{e}_i \cdot \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u} \cdot \mathbf{u}}{2c^2} \right) \right] \tag{8}$$

The individual terms may be summed separately, so the distribution function values do not have to be explicitly calculated, streamed or stored. This will dramatically reduce the communication costs for parallel simulations. For the case of $\tau = 1$, the coefficient A_u equals one and both the lattice-Boltzmann and lattice kinetics methods produce the same distribution function values for calculating momentum.

The elimination of the need to store and stream distribution functions has several other practical advantages. One example is the simplification of checkpoint restarts. To save the state of a lattice-Boltzmann simulation and restart without introducing any false transients, the particle distribution functions for all lattice directions must be written to file and read back into the program. This is because momentum is transferred in a chain from distant lattice sites through the distribution functions. An exception to this is when the relaxation parameter τ has a value of exactly one. In this case, the chain has a minimum length because momentum effects of upstream sites are obliterated by each collision. Distribution functions are set equal to the equilibrium values, which are calculated at each lattice site from the current values of the primitive variables. Therefore, when $\tau = 1$, the state of a lattice-Boltzmann simulation is completely specified by the primitive variable values. With the implicit lattice kinetic method described here, this advantage is maintained regardless of the system viscosity. Instead of momentum effects being transferred via the decay of probability distributions during the collision step, they are accounted for in the expressions for the equilibrium distributions themselves [by Eqs. (7) and (8)].

In lattice-Boltzmann models, single particle distributions are streamed out from a given lattice site and replaced by distributions streaming in from the site's immediate neighbors during each time step. The distributions are then summed according to Eq. (6) to find the new density (or pressure) and velocity values. Changes in the primitive variables are therefore calculated according to the differences of the sums of the distributions entering and leaving the site over the course of the previous time step. The lattice-Boltzmann method is fully explicit because the new-time values of the primitive variables are calculated only from old-time information. In finite difference approaches to CFD, implicit numerical schemes are routinely used to improve stability and enable larger time steps [16]. Similarly, in the method described here, primitive variables are calculated using the new-time distribution functions entering and leaving a given lattice site.

The new-time pressure and momentum for incompressible flows are calculated using the following expressions:

$$\frac{p^{n+1}}{\kappa} = \frac{p^n}{\kappa} + \left(\sum_{i=1}^{N} f_{i,\rho} \right)_{in}^{n+1} - \left(\sum_{i=1}^{N} f_{i,\rho} \right)_{out}^{n+1} = \frac{p^n}{\kappa} + \left(\sum_{i=1}^{N} f_{i,\rho}^{\mathbf{x}-\mathbf{e_i}} \right)^{n+1} - (1 - w_0) \frac{p^{n+1}}{c_s^2} - \frac{3}{2} w_0 \frac{\rho \mathbf{u} \cdot \mathbf{u}}{c^2} \tag{9}$$

$$\begin{aligned} (\rho \mathbf{u})^{n+1} &= (\rho \mathbf{u})^n + \left(\sum_{i=1}^{N} \mathbf{e}_i f_{i,u} \right)_{in}^{n+1} - \left(\sum_{i=1}^{N} \mathbf{e}_i f_{i,u} \right)_{out}^{n+1} + \mathbf{S_0} - S_1 \mathbf{u}^{n+1} \\ &= (\rho \mathbf{u})^n + \left(\sum_{i=1}^{N} \mathbf{e}_i f_{i,u}^{\mathbf{x}-\mathbf{e}_i} \right)^{n+1} - (\rho \mathbf{u})^{n+1} \frac{6\nu}{c^2 \Delta t} + \mathbf{S_0} - S_1 \mathbf{u}^{n+1} \end{aligned} \tag{10}$$

where κ is the derivative of pressure with respect to density. The superscript ^{n} indicates values at the beginning of the current time step, while the superscript $^{n+1}$ indicates estimated values at the end of the current time step. N is the total number of lattice directions, not counting the stationary 0 direction. For example, $N = 8$ for the D2Q9 lattice, and $N = 18$ for the D3Q19 lattice. When the probability distributions are summed in the lattice-Boltzmann method to update the primitive variables via Eq. (6), the stationary 0 direction does not contribute to velocity or pressure changes.

Eqs. (9) and (10) can be rearranged to give

$$p^{n+1} = \frac{p^n + \kappa\left(\sum_{i=1}^{N} f_{i,\rho}^{\mathbf{x}-\mathbf{e_i}}\right)^{n+1} - \kappa\frac{3}{2}w_0\frac{\rho\mathbf{u}\cdot\mathbf{u}}{c^2}}{1+\frac{\kappa}{c_s^2}(1-w_0)} \tag{11}$$

$$\mathbf{u}^{n+1} = \frac{(\rho\mathbf{u})^n + \left(\sum_{i=1}^{N}\mathbf{e}_i f_{i,u}^{\mathbf{x}-\mathbf{e_i}}\right)^{n+1} + \mathbf{S_0}}{\rho + \rho\frac{6\nu}{c^2\Delta t} + S_1} \tag{12}$$

Here $f_{i,k}^{\mathbf{x}-\mathbf{e_i}}$ represents the single particle probability distribution arriving from the neighboring lattice site in the $-\mathbf{e}_i$ direction. With the lattice kinetic approach described here, these values are not streamed from the neighboring sites, but are calculated where needed from the primitive variables at the neighbor location using Eq. (8). The third terms on the right hand sides of Eqs. (9) and (10) are calculated by summing the terms for the probability functions leaving the site in all lattice directions according to Eq. (8). Their relative simplicity is the result of the fact that many of the terms from the various lattice directions cancel each other during the summation.

The variable κ is the derivative of pressure with respect to density, which is set equal to c_s^2 in the lattice-Boltzmann method. κ has been varied over a wide range in several simulations using the implicit lattice kinetic method, with little apparent affect on the converged velocity and pressure fields. It is normally set to a value 1–40 times c_s^2. Since κ determines the stiffness of the pseudo-compressible method, a given simulation would likely become unstable if the value were set to high, and excessive iterations might be required if it were set too low. This parameter space has not yet been fully explored, but the stability criteria are expected to depend upon the specific problem being solved.

$\mathbf{S}_0$ and S_1 are momentum source terms. $\mathbf{S}_0$ and S_1 can include such momentum sources as external body forces ($\mathbf{S}_0$) or velocity dependent terms such as Darcy resistance in a porous medium (S_1). Body forces, such as gravity, may be included in the momentum source term as a function of local density

$$\mathbf{S_0} = \rho\mathbf{g} \tag{13}$$

where $\mathbf{g}$ is the acceleration due to gravity. Source terms also include certain necessary corrections for some minor second-order terms arising from the Taylor expansion described in the following section.

This implicit scheme has the advantage of being stable at larger time steps but the disadvantage of requiring an iterative solution. The number of iterations required may be reduced by using a predictor–corrector scheme. Initial guesses for the primitive variables are calculated using a time derivative from the previous time step

$$p^{n+1} = p^n + \left(\frac{\partial p}{\partial t}\right)^n \Delta t \tag{14}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \left(\frac{\partial \mathbf{u}}{\partial t}\right)^n \Delta t$$

The solution procedure at each time step is as follows:

(1) Predict the new-time pressure and velocity values using time derivatives as shown in Eq. (14).
(2) Calculate the corrected new-time values using Eqs. (11) and (12) with the most recent pressure and velocity values to calculate the neighbor distribution functions.
(3) Compare the changes in pressure and velocities to specified convergence limits.
(4) Repeat steps 2 and 3 until convergence is satisfied.

The convergence limit for velocities is typically three or four orders of magnitude below the expected maximum velocity of the system but varies with the application. The number of iterations per time step depends on the transient behavior of the system. For transients with a relatively linear change in velocity, the procedure may converge with one iteration per time step. Regions with significant acceleration or deceleration will require several iterations to reach convergence.

2.3. Recovery of Navier–Stokes equations

We can demonstrate that this method results in a solution to the continuity and Navier–Stokes equations for a two-dimensional system by performing a Taylor expansion of the pressure (or density) and velocities to second-order. For example, the expansion for pressure is

$$p(x_1+\Delta x_1, x_2+\Delta x_2) = p(x_1,x_2) + \Delta x_1\frac{\partial p}{\partial x_1} + \Delta x_2\frac{\partial p}{\partial x_2} + \frac{1}{2}\left[(\Delta x_1)^2\frac{\partial^2 p}{\partial x_1^2} + 2\Delta x_1\Delta x_2\frac{\partial^2 p}{\partial x_1\partial x_2} + (\Delta x_2)^2\frac{\partial^2 p}{\partial x_2^2}\right] \tag{15}$$

The pressure and momentum evolution Eqs. (11) and (12) require the single particle probability distributions, $f_{i,k}^{\mathbf{x}-\mathbf{e_i}}$, arriving from the neighboring lattice sites. These distributions, in turn, depend upon the pressure and velocity values at each of the neighboring sites according to Eq. (8). Separating the RHS of Eq. (8) into three terms for convenience and substituting in the

Taylor expanded pressure and velocity fields with $\Delta x_1 = \Delta x_2 = \Delta x$, we find the following sums for use in the pressure evolution Eq. (11)

$$\sum_{i=1}^{N} w_i p^{\mathbf{x}-\mathbf{e_i}} = (1-w_0)p(x_1,x_2) + \frac{1}{6}\left(\frac{\partial^2 p}{\partial x_1^2} + \frac{\partial^2 p}{\partial x_2^2}\right)\Delta x^2 = (1-w_0)p(x_1,x_2) + \frac{1}{6}(\nabla^2 p)\Delta x^2 \tag{16}$$

$$\sum_{i=1}^{N} w_i (3\mathbf{e}_i \cdot \mathbf{u}^{\mathbf{x}-\mathbf{e_i}}) = -\left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2}\right)\Delta x = -(\nabla \cdot \mathbf{u})\Delta x \tag{17}$$

$$\sum_{i=1}^{N} w_i [4.5(\mathbf{e}_i \cdot \mathbf{u}^{\mathbf{x}-\mathbf{e_i}})^2 - 1.5\mathbf{u}^{\mathbf{x}-\mathbf{e_i}} \cdot \mathbf{u}^{\mathbf{x}-\mathbf{e_i}}] = \frac{3}{2}(1-(1-w_0))\mathbf{u}(x_1,x_2)\cdot\mathbf{u}(x_1,x_2) + \left(\frac{\partial u_1}{\partial x_1}\right)^2 + \left(\frac{\partial u_2}{\partial x_2}\right)^2 + \frac{\partial u_1}{\partial x_1}\frac{\partial u_2}{\partial x_2} + \frac{\partial u_1}{\partial x_2}\frac{\partial u_2}{\partial x_1} + \frac{1}{2}\left(u_1\frac{\partial^2 u_1}{\partial x_1^2} + u_2\frac{\partial^2 u_2}{\partial x_2^2}\right) + O\left(\frac{\partial^2 \mathbf{u}}{\partial x_k^2}\right)^2 \tag{18}$$

Likewise, we find the following sums for use in the momentum evolution Eq. (12) for the first coordinate direction $\hat{\mathbf{x}}_1$

$$\sum_{i=1}^{N} w_i (\mathbf{e}_i \cdot \hat{\mathbf{x}}_1) p^{\mathbf{x}-\mathbf{e_i}} = -\frac{1}{3}\frac{\partial p}{\partial x_1}\Delta x \tag{19}$$

$$\sum_{i=1}^{N} w_i (\mathbf{e}_i \cdot \hat{\mathbf{x}}_1)(3\mathbf{e}_i \cdot \mathbf{u}^{\mathbf{x}-\mathbf{e_i}}) = u_1(x_1,x_2) + \frac{1}{6}\left(\frac{\partial^2 u_1}{\partial x_1^2} + \frac{\partial^2 u_1}{\partial x_2^2} + \frac{\partial^2 u_1}{\partial x_1 \partial x_2}\right)\Delta x^2 + \frac{1}{3}\frac{\partial^2 u_1}{\partial x_1^2}\Delta x^2 \tag{20}$$

$$\sum_{i=1}^{N} w_i (\mathbf{e}_i \cdot \hat{\mathbf{x}}_1)\left[(4.5\mathbf{e}_i \cdot \mathbf{u}^{\mathbf{x}-\mathbf{e_i}})^2 - 3\mathbf{u}^{\mathbf{x}-\mathbf{e_i}} \cdot \mathbf{u}^{\mathbf{x}-\mathbf{e_i}}\right] = -\left(u_1\frac{\partial u_1}{\partial x_1} + u_2\frac{\partial u_1}{\partial y_2}\right)\Delta x - u_1\left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2}\right)\Delta x + O\left(\frac{\partial u}{\partial x_k}\frac{\partial^2 \mathbf{u}}{\partial x_k^2}\right) \tag{21}$$

where $\hat{\mathbf{x}}_1$ represents the unit vector in the first coordinate direction, so that $\mathbf{e}_i \cdot \hat{\mathbf{x}}_1$ is the component of the lattice direction vector $\mathbf{e}_i$ in coordinate direction 1. Equations analogous to (19) and (20) can be written for all coordinate directions. The first term on the right hand side of Eq. (21) pertains to momentum advection and can be expressed as $\mathbf{u}\cdot\nabla\mathbf{u}$ when all coordinate directions are included. The second term on the right hand side of Eq. (21) can be expressed as $\mathbf{u}(\nabla\cdot\mathbf{u})$ with all of the coordinate directions and represents an un-physical change in momentum arising from the pseudo-compressibility of the system. Momentum conservation can be enforced by including a corresponding correction in term S_1 of Eq. (12). The final term in Eq. (18) represents error terms on the order the squares and cross-products of various second derivatives. Likewise, the third term in Eq. (21) indicates error terms on the order of the products of various first and second velocity derivatives.

Note that the pressure gradient calculated in Eq. (19) is located at the lattice node centers as compared to finite volume methods where it is defined between nodes. Additional pressure information is required to accurately calculate the mass flux between lattice nodes. The pressure Laplacian in Eq. (16) provides the correction term required for a second-order accurate solution to the momentum equation. The same is true for the corrective terms derived in Eq. (18) applied to the lattice node centered convective velocity gradients calculated in Eq. (21).

Combining these results and neglecting higher order terms, the resulting continuity and momentum equations take the form

$$\frac{\partial p}{\kappa \partial t} = -\rho \nabla \cdot \mathbf{u} \tag{22}$$

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \rho\mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \mu\nabla^2\mathbf{u} + \mu_b\nabla(\nabla\cdot\mathbf{u}) + (\mathbf{S_0} - S_1\mathbf{u}) \tag{23}$$

which are the continuity and Navier–Stokes equations for a pseudo-compressible method. The bulk viscosity, μ_b, is obtained from the final term in Eq. (20). A similar procedure could be followed for the compressible form of the distribution function expression (7).

The implicit lattice kinetics method is closely related to the discreet velocity model (DVM) approach to solving the kinetic theory equations. In the DVM approach, the fluid velocities are discretized into a finite set and the evolution is solved based on a specified collision operator. The lattice-Boltzmann method may be considered a special case of DVM where the lattice grid spacing and time step discretization are coupled with a limited discrete velocity set. Alternative versions of the DVM have also been developed for solving the Navier–Stokes equation. The method developed by Klar [17] solves a set of coupled moment equations on a staggered grid as in the Marker-and-Cell method. The method developed by Junk and Rao [18] derives a distribution function expression, using a Chapman–Enskog analysis, that can be summed directly to determine the new-time velocities and pressure. The primary difference between the Junk and Rao method and the method described here is how the viscosity is specified and the implicit nature of the current method. All of the methods discussed, including the incompressible lattice-Boltzmann, may be considered pseudo-compressible, as the change in pressure is coupled to the continuity equation using an artificial compressibility.

2.4. Boundary conditions

Three types of flow boundary conditions have been implemented in the lattice kinetics method. In each case, the velocity is calculated using the standard velocity expression (12). The only difference is that there are missing values for the incoming distribution function sum. The following is a description of how to determine the missing values for each boundary type:

- For *specified flow boundary* conditions (including no-slip boundaries), we use the standard mid-plane bounceback boundary conditions commonly used with the lattice-Boltzmann method [4]. Particle probability distributions entering a lattice site from the direction of a no-slip boundary are set equal to the value leaving in the opposite direction. For example, at point (i,j) in Fig. 3 the value of distribution f_6 entering would be set to the value of f_5 leaving (using the vector directions shown in Fig. 1). For a specified flow boundary, the appropriate velocity term is added to the wall distribution function values.
- The *slip boundary* is very straightforward. Primitive variables and properties are copied across the line of symmetry, reversing the direction of velocity components normal to the boundary, and the distribution function terms are calculated as if the boundary nodes were standard fluid nodes. For example, in Fig. 3, pressure and velocity values for site (i,j) are copied across the boundary to site $(i+1,j)$, reversing the sign on the horizontal velocity component. These values would then be referenced to calculate single particle distribution functions at site (i,j) just as if site $(i+1,j)$ were a standard fluid node.
- The *pressure boundary* is similar to that developed by Zou and He [19] where the sum of incoming fluid node distribution functions and the specified pressure or density are used to calculate the sum of missing distribution function values. For the two-dimensional case shown in Fig. 3, the distribution values entering site (i,j) will be known for all directions except those coming from the boundary sites, f_2, f_6, and f_8 (see Fig. 1). The pressure gradient in the normal direction is assumed to be linear and the velocity squared terms are assumed to be uniform. In other words, the flow is assumed to be normal to the boundary surface. As a consequence of this assumption, distributions with components in directions perpendicular to the normal must cancel each other. The sum of incoming distribution functions then becomes:

$$\sum_{\mathbf{e_i}\cdot\mathbf{n}=1} f_i = (1-w_0)\frac{p}{\kappa} - \sum_{\mathbf{e_i}\cdot\mathbf{n}\leqslant 0} f_i \tag{24}$$

where $\mathbf{n}$ is the unit vector normal to the boundary.

3. Numerical examples

In this section, the incompressible form of the implicit lattice kinetic method is applied to several simple example problems with known solutions. In all cases, the stiffness parameter κ in Eq. (8) is set to a value of 10 times c_s^2. Kinematic viscosity (ν) was set to 0.01 cm^2/s and density (ρ) to 1.0 g/cm^3.

3.1. Lid-driven cavity

The two-dimensional lid-driven cavity is a commonly applied validation case for methods used to approximate the Navier–Stokes equations. The domain consists of a square cavity with three stationary walls and a fourth wall sliding length-

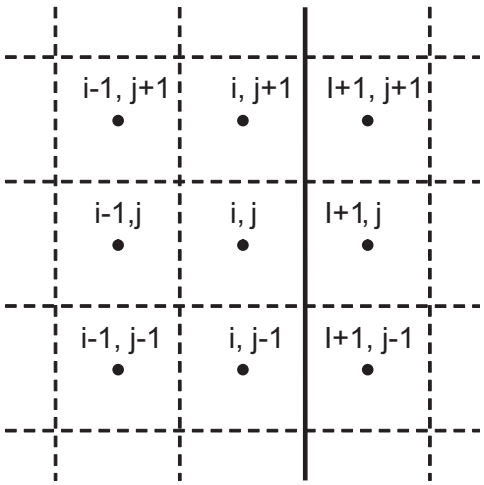

Fig. 3. Lattice sites near a boundary surface in two dimensions.

wise at a constant velocity. Solutions for the resulting steady-state velocity fields have been refined over the years using many different computational approaches and provide reasonably precise standards for comparison to solutions by new methods.

The implicit lattice kinetic approach described herein was applied to a lid-driven cavity with Reynolds numbers of 100 and 1000, based on the lid velocity and the cavity width. Lattice spacing (Δx) and time step (Δt) were set to 0.1 cm and 0.01 s, respectively. The computational grid measured 132 lattice spaces squared. Each of the four wall boundaries was two lattice spaces thick, giving cavity dimensions of 128 lattice spaces squared. Although this is a two-dimensional problem, a three-dimensional grid was used in order to validate the 3D implementation of the method. The domain had a depth of 3 lattice spaces, and periodic boundaries were used in the depth direction.

Fig. 4 shows comparisons of horizontal velocities on the vertical mid-line found by the implicit lattice kinetic method to those presented by Ghia et al. [20]. Fig. 5 shows a similar comparison of vertical velocities on the horizontal mid-line. Velocities shown are scaled by the lid velocity. Both figures show good agreement between the velocity fields calculated by the implicit lattice kinetic method and the baseline solution. As expected, no significant difference in the velocity field was observed in the depth direction.

The lid-driven cavity problem with $N_{Re} = 1000$ was also used to test the stability of the implicit lattice-Kinetic method relative to lattice-Boltzmann with BGK relaxation. In each case, Δt was increased from the baseline value of 0.01 s until the simulations became unstable. The largest time step which could be achieved with lattice-Boltzmann was 0.06 s, which corresponds to a lid velocity of roughly 0.47 times the Courant limit = $\Delta x/\Delta t$. At $\Delta t = 0.07$ s, the lattice-Boltzmann simulation rapidly became unstable and had produced infinite values for primitive variables within a few hundred time steps. Much larger time steps could be achieved with the implicit lattice kinetic method. Some oscillatory behavior was observed as the time step increased, but for a time step as large as 0.3 s the program still converged to a solution which agrees with that obtained with the baseline Δt. This corresponds to a lid velocity of over 2.3 times the Courant limit. This demonstrates that the implicit lattice kinetic approach has clear stability advantages over lattice BGK.

It should also be noted that at the baseline time step of 0.01 s on the grid employed, the relaxation parameter, τ, for the lattice-Boltzmann simulation was equal to a value of 0.53. The lattice BGK method is known to become unstable near $\tau = 0.5$, which may have been an issue if the time step had been made smaller. Increasing the time step to $\Delta t = 0.07$ s where the lattice BGK simulation had became unstable due to the Courant limit, the relaxation parameter $\tau = 0.71$ is still far from the value of $\tau = 1$ which is considered the 'safest' parametric value for lattice BGK [4]. In order to reach $\tau = 1$ and still have a maximum velocity far enough below the Courant limit for a stable solution, the only option with the lattice-Boltzmann method would be to make the spatial resolution for this problem significantly greater, at a considerable computational cost. This illustrates the difficulty in applying lattice-Boltzmann to a given problem with real dimensions, while staying within the recommended parametric ranges.

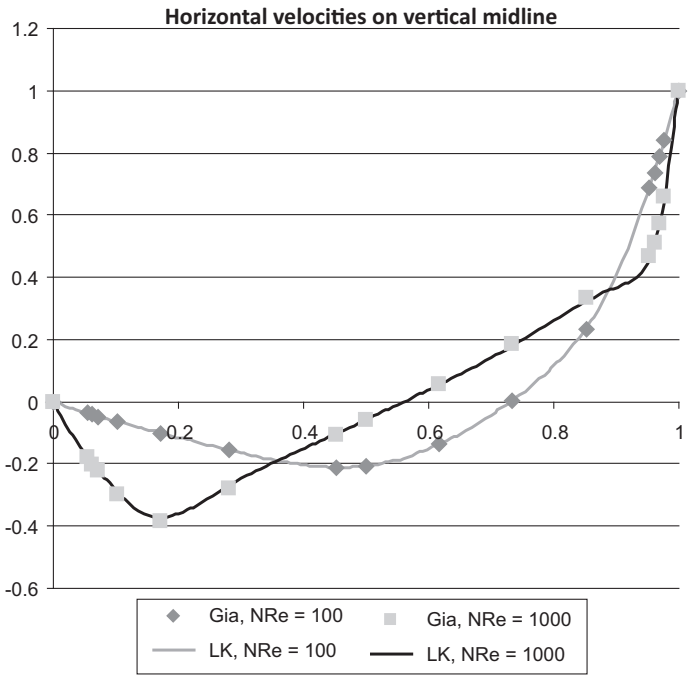

Fig. 4. Horizontal velocities on the vertical mid-line in a square lid driven cavity.

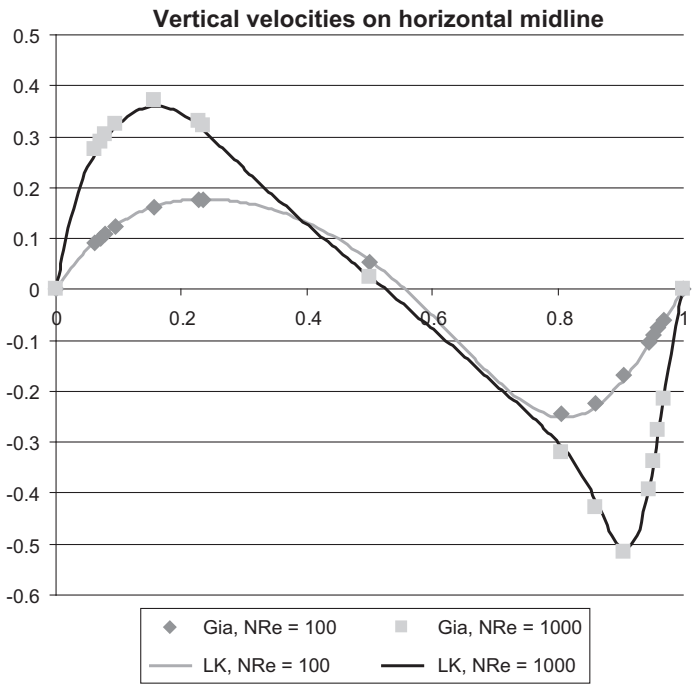

Fig. 5. Vertical velocities on the horizontal mid-line in a square lid driven cavity.

3.2. Oscillating plate

The transient velocity field over an infinite plate which is oscillating laterally is known as a Stokes boundary layer. When the velocity of the plate is given by

$$u_{plate} = u_0 \cos(\omega t) \tag{25}$$

the velocity at time t and normal distance y from the plate is [21]

$$u(y,t) = u_0 e^{-\sqrt{\frac{\omega}{2\nu}}y} \cos\left(\omega t - \sqrt{\frac{\omega}{2\nu}} \cdot y\right) \tag{26}$$

where ω is the angular frequency of oscillation in radians/time.

This regime was simulated on a computational grid 3 lattice spaces deep, 8 lattice spaces along the direction of oscillation, and 128 lattice spaces in the direction normal to the plate surface. Periodic boundaries were used on the four sides. The bottom wall thickness was two lattice spaces. Lattice spacing (Δx) and time step (Δt) were set to 5.0E−3 cm and 4.0E−4 s, respectively. The frequency of oscillation was 2π radians/s. The maximum plate velocity, u_0, was set to 1 cm/s.

The simulation was run for 20 s, or 20 full cycles until there was little variation in the velocity field from one cycle to the next. Fig. 6 shows a comparison between velocities predicted by the simulation and the exact solution at four different times during the last cycle. The plot shows good agreement between the simulation and the analytical solution.

3.3. Transient 2D channel

Unsteady Poiseuille flow is a convenient benchmark problem for transient flow driven by a pressure gradient or body force. Here a viscous fluid is confined between two infinite stationary walls. The fluid begins at rest, when a body force field is applied in a direction parallel to the wall surfaces. The fluid is accelerated toward the well-known parabolic steady-state velocity profile. An analytical solution to the problem for a gap width of 2b is given in [22]

$$u(y,t) = \frac{gb^2}{2\nu}\left[1 - \frac{y^2}{b^2} - \frac{32}{\pi^3}\sum_{n=0}^{N}\left[\frac{(-1)^n}{(2n+1)^3}\cos\left(\frac{(2n+1)\pi y}{2b}\right)e^{\frac{-(2n+1)^2\pi^2\nu t}{4b^2}}\right]\right] \tag{27}$$

where y is measured from a plane half-way between the walls and g is the rate of acceleration due to body forces. $N = 30$ terms were included in the series to calculate velocities for comparison to the simulation results. With this number of terms in the series, all of the predicted velocity values had converged to well within the precision reported.

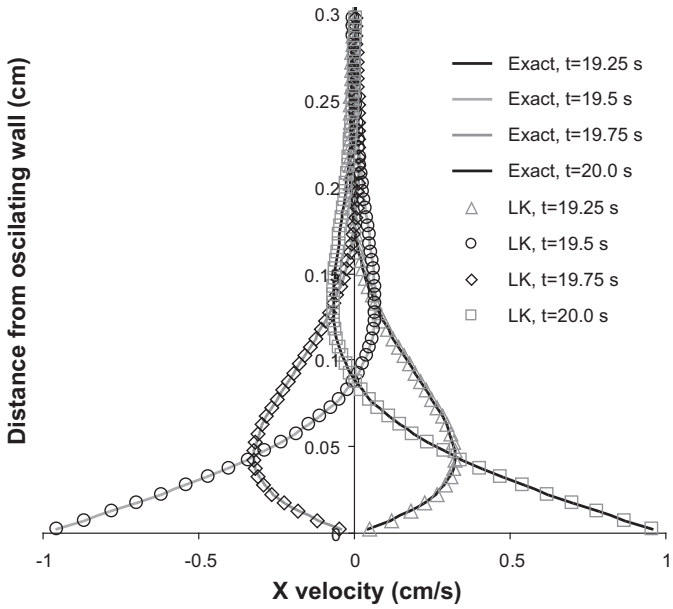

Fig. 6. Horizontal velocities at various distances from an infinite oscillating plate.

The simulation was run on a computational grid measuring 8 lattice spaces in the direction of flow, 3 lattice spaces in the depth direction, and 13 lattice spaces in the direction normal to the wall surfaces. Stationary wall thicknesses of 1 lattice space were used, giving a net gap width of 11 lattice spaces. The lattice spacing (Δx) was 0.25 cm in all cases. The baseline timestep (Δt) was 0.10417 s. The body force acceleration (g) was set to 5.0777E−3 cm/s^2. Fig. 7 shows a comparison between the analytical solution and implicit lattice kinetics simulation at several different points in time. The simulation is able to predict the correct velocity profiles and rate of evolution.

The same problem was solved using the implicit lattice kinetics method and the conventional lattice-Boltzmann method with several different values for the timestep. The evolution of the centerline velocity predicted in each case is shown in

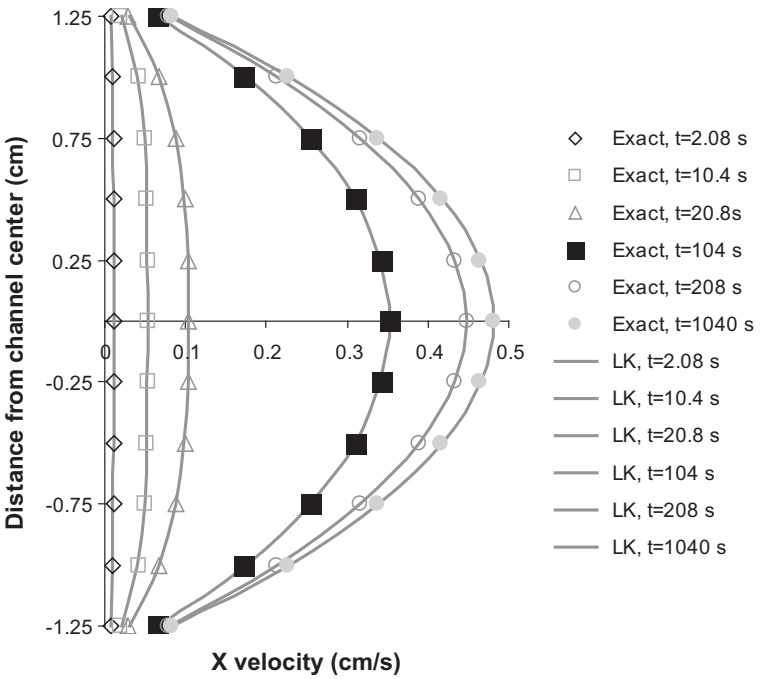

Fig. 7. Velocity profiles in unsteady Poiseuille flow.

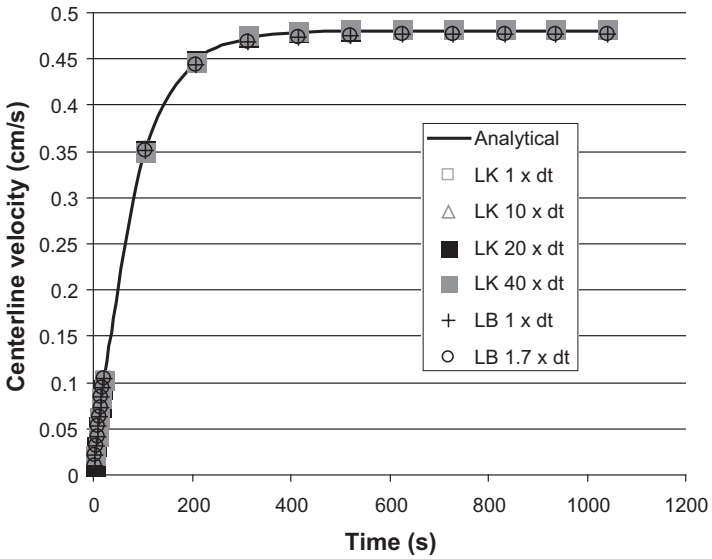

Fig. 8. Evolution of centerline velocity in unsteady Poiseuille flow.

Fig. 8. Both methods produced a reasonable prediction of the centerline velocity evolution. In general, the maximum relative error of both the LB and LK predictions increased with longer timesteps. The implicit LK method tends to yield smaller errors initially because the internal iterations allow faster acceleration of the flow from rest. The maximum absolute error with the LK method remains below 2.5% even with the largest timestep examined.

At 1.7 times the baseline timestep, the maximum velocity reached at the channel centerline corresponds to a Courant number of 0.34. At 1.8 times the baseline timestep, the LB simulation became unstable and produced spurious velocities in the y and z direction. The LK simulation eventually converged to the steady-state solution with a timestep of 1024 times the baseline value, but the transient behavior could obviously not be accurately predicted with such low temporal resolution.

Table 1 shows the wall-clock time needed to solve the flow field up to a simulated time of 1040 s with the various timesteps used in the two methods. The simulations were run using two AMD Opteron 2354 2.2 GHz processors in parallel. It can be seen that at the baseline timestep, the LK method takes 46% longer to solve the problem than conventional LB. Several factors could contribute to the slower performance of the LK method. Compared to explicit LB, some additional floating point operations are needed to make the LK calculation implicit and to separate the probability function terms such that viscosity effects can be added according to Eqs. (7) and (8). Iteration of the semi-implicit method could also contribute, but this particular test problem generally does not require more than one iteration per time step. Another significant factor in the current LK implementation seems to be extensive use of conditional masking statements (to indicate active cells and boundary conditions) which trigger a different level of optimization. It is likely that the performance gap between LB and LK at a given timestep for this particular problem could be closed somewhat by reorganization of the code and optimization for speed.

In any case, the improved stability and temporal accuracy of the LK method allow much larger timesteps. With a timestep of 40 times the baseline value, the LK method yields a reasonably accurate transient solution in roughly one fifth the wall-clock time required for the fastest LB simulation. At the longest timestep examined here, the LK method seems to be approaching a point of diminishing returns in terms of minimizing wall-clock time for an accurate transient solution to this particular problem. In general, the added flexibility of the implicit LK method can be expected to result in faster solutions than LB in cases where the LB timestep is limited by the Courant condition. This limitation might be especially important in simulations with very high local velocities, as with fluid jets or enclosed flows. The LB method might be expected to have the advantage in problems where the Courant limit is not an issue and in problems where rapidly changing velocities over

Table 1
Required wall-clock time for simulation of 1040 s in the unsteady Poiseuille example problem.

Case	Wall-clock time (s)
LK, $\Delta t = 1 \cdot$ baseline	5.01
LK, $\Delta t = 10 \cdot$ baseline	0.58
LK, $\Delta t = 20 \cdot$ baseline	0.44
LK, $\Delta t = 40 \cdot$ baseline	0.41
LB, $\Delta t = 1 \cdot$ baseline	3.43
LB, $\Delta t = 1.7 \cdot$ baseline	2.06

the entire course of the simulation would require a large number of iterations per timestep when using the implicit LK approach.

4. Conclusions

A technique has been developed for the simulation of fluid flow that includes many of the advantages of the lattice-Boltzmann method and overcomes some of the limitations that hinder its application to a wide range of practical problems. The proposed implicit lattice kinetic method uses two single particle probability functions, one to solve the pressure or density field, and one to enforce a momentum balance. Instead of streaming probability functions between neighboring lattice sites like the lattice-Boltzmann method, the implicit lattice kinetics method calculates distribution function values using primitive variables at neighboring sites, along with a term which allows the momentum transfer to be adjusted for viscosity. It can be shown that the proposed system leads to a recovery of the classical Navier–Stokes equations. The application of commonly used boundary conditions is straightforward, as are the application of momentum sources and sinks such as body forces or Darcy resistance to flow through a porous medium. It has been shown that the method gives accurate solutions to the 2D lid-driven cavity problem with Reynolds numbers of 100 and 1000. The method also gives accurate transient solutions to the Stokes boundary layer and unsteady Poiseuille problems. The implicit lattice kinetic method has also been shown to exhibit considerably greater stability than the lattice BGK method at larger time steps, which can result in accurate transient solutions in less wall-clock time.

Acknowledgment

This work was performed under a US Department of Energy LDRD program.

References

[1] P. Yuan, L. Schaefer, Equations of state in a lattice Boltzmann model, Phys. Fluids 18 (2006).
[2] H.N. Li, C.X. Pan, C.T. Miller, Pore-scale investigation of viscous coupling effects for two-phase flow in porous media, Phys. Rev. E 72 (2005).
[3] H.D. Chen, S. Kandasamy, S. Orszag, R. Shock, S. Succi, V. Yakhot, Extended Boltzmann kinetic equation for turbulent flows, Science 301 (2003) 633–636.
[4] M.C. Sukop, D.T. Thorne, Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers, Springer, Berlin, NY, 2006.
[5] K. Sankaranarayanan, X. Shan, I.G. Kevrekidis, S. Sundaresan, Analysis of drag and virtual mass forces in bubbly suspensions using an implicit formulation of the lattice Boltzmann method, J. Fluid Mech. 452 (2002) 61–96.
[6] T. Inamuro, A lattice kinetic scheme for incompressible viscous flows with heat transfer, Philos. T. Roy. Soc. A 360 (2002) 477–484.
[7] Y. Peng, C. Shu, Y.T. Chew, H.W. Zheng, New lattice kinetic schemes for incompressible viscous flows, Int. J. Mod. Phys. C 15 (2004) 1197–1213.
[8] T. Inamuro, T. Ogata, A lattice kinetic scheme for bubble flows, Philos. T. Roy. Soc. A 362 (2004) 1735–1743.
[9] D. d'Humieres, I. Ginzburg, M. Krafczyk, P. Lallemand, L.S. Luo, Multiple-relaxation-time lattice Boltzmann models in three dimensions, Philos. T. Roy. Soc. A 360 (2002) 437–451.
[10] I. Ginzburg, D. d'Humieres, Multireflection boundary conditions for lattice Boltzmann models, Phys. Rev. E 68 (2003).
[11] X.Y. He, Q.S. Zou, L.S. Luo, M. Dembo, Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice Boltzmann BGK model, J. Stat. Phys. 87 (1997) 115–136.
[12] C. Pan, L.-S. Luo, C.T. Miller, An evaluation of lattice Boltzmann schemes for porous medium flow simulation, Comput. Fluids 35 (2006) 898.
[13] X. He, L. Luo, A priori derivation of the lattice Boltzmann equation, Phys. Rev. E 55 (1997) R6333–R6336.
[14] D.A. Wolf-Gladrow, Lattice-Gas Cellular Automata and Lattice Boltzmann Models: An Introduction, Springer, 2000.
[15] X. He, L. Luo, Lattice Boltzmann model for the incompressible Navier–Stokes equation, J. Stat. Phys. 88 (1997) 927–944.
[16] J.H.P.M. Ferziger, Computational Methods for Fluid Dynamics, Springer, New York, 2002.
[17] A. Klar, Relaxation scheme for a lattice-Boltzmann-type discrete velocity model and numerical Navier-Stokes limit, J. Comput. Phys. 148 (1999) 416–432.
[18] M. Junk, S.V.R. Rao, A new discrete velocity method for Navier-Stokes equations, J. Comput. Phys. 155 (1999) 178–198.
[19] Q. Zou, X. He, On pressure and velocity boundary conditions for the lattice Boltzmann BGK model, Phys. Fluids 9 (1997) 1591–1598.
[20] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.
[21] A. Shah, L. Yuan, A. Khan, Upwind compact finite difference scheme for time-accurate solution of the incompressible Navier–Stokes equations, Appl. Math. Comput. 215 (2010) 3201–3213.
[22] M. Emin Erdogan, On the unsteady unidirectional flows generated by impulsive motion of a boundary or sudden application of a pressure gradient, Int. J. Non Linear Mech. 37 (2002) 1091–1106.