JOURNAL OF COMPUTATIONAL PHYSICS **129,** 357–363 (1996)
ARTICLE NO. 0255

Some Progress in Lattice Boltzmann Method. Part I. Nonuniform Mesh Grids

Xiaoyi He,[1] Li-Shi Luo,[2] and Micah Dembo

Los Alamos National Laboratory, Los Alamos, New Mexico 87545

Received January 18, 1996; revised July 19, 1996

A new lattice Boltzmann algorithm is proposed to simulate the Navier–Stokes equation on arbitrary nonuniform mesh grids. The new algorithm retains the advantages of the lattice Boltzmann method: parallel of algorithm, ease of programming, and ability to incorporate microscopic interactions. A simulation of flow in a two-dimensional symmetric channel with sudden expansion is carried out using the new algorithm on a nonuniform mesh. The results of the simulation are in excellent agreement with previous experimental and numerical results. © 1996 Academic Press, Inc.

I. INTRODUCTION

In recent years, the lattice Boltzmann (LB) method [1–5] has attracted much attention in the computational physics and engineering communities. The LB method has demonstrated its ability to simulate single-component hydrodynamics [1–3], multiphase and multicomponent fluids [6] including particulate suspensions [7], magneto-hydrodynamics [8], reaction-diffusion systems [9], flows through porous media [10], and other complex systems [4, 5]. Meanwhile, the LB method has demonstrated a significant potential and broad applicability with numerous computational advantages, including the parallel of algorithm, the simplicity of programming, and the ability to incorporate microscopic interactions. However, the method also has some undesirable shortcomings. One of these shortcomings, which is addressed specifically in this paper, is the uniformity of its mesh grids.

Historically, the LB method evolves from the lattice-gas automata (LGA) method [11]. Consequently, the LB method inherits some features from its precursor, the LGA method. In the lattice-gas automata, the dynamics of particles evolving on a lattice space consists of two steps: (1) particles at the same site collide according to a set of hard-sphere particle collision rules which conserve mass, momentum, and energy (for multispeed models) at each lattice site; (2) after colliding, particles advect to the next lattice sites in the direction of their velocities. The small number of discrete velocities allowed is consistent with the simple lattice structures of LGA models. In other words, the discretization of physical space is coupled with the discretization of momentum space. The first LB model was a floating-point version of its LGA counterpart—each particle in the LGA model (represented by a single-bit Boolean integer) was replaced by a single-particle distribution function, f_α, in the LB model (represented by floating-point number). The lattice structure and the evolution rule remained the same [1]. Two important improvements to enhance the computational efficiency have been made for the LB method: the linearization of the collision operator [2] and the BGK approximation [12] (single relaxation time approximation) [3]. The uniform lattice structure was unchanged.

A nonuniform mesh is certainly desirable in many practical applications. What inhibits the use of a nonuniform grid in previous LB methods is the coupling of the discretization of physical and momentum spaces. The minimal advection distance of the density distributions in a single time step must be equal to the minimal lattice separation. The density distributions have to move from one lattice site to another in a single step to warrant the followed collision process. This feature is inherited from the LGA, but is not necessary for the LB method. Since the prime variable in LB method is the single particle distribution function, the exactness of the Boolean operation in the LGA method is no longer mandatory and various approximations can be employed.

In this paper, we propose an LB algorithm for nonuniform mesh grids. In this new algorithm, the computational mesh is uncoupled from the discretization of momentum space and it can have an arbitrary shape. Collisions still take place on the grid points of the computational mesh. After a collision, the density distributions move according to their velocities. Although the density distributions at the grid sites now may not be exactly determined, they can always be calculated using interpolation. After interpolation, collision and advection steps are repeated.

This paper is organized as follows. Section II briefly reviews the 9-bit lattice BGK model as an example of

[1] E-mail address: xyh@t13.lanl.gov.

[2] E-mail address: luo@t13.lanl.gov.

0021-9991/96 $18.00

implementing the LB algorithm on nonuniform mesh grids. Section III describes the general procedure to implement the LB algorithm on a nonuniform mesh. Section IV presents numerical results of simulations of flow in a 2D sudden expansion channel using the 9-bit lattice BGK model on nonuniform mesh grids. Section V discusses the results and concludes the paper.

II. NINE-BIT INCOMPRESSIBLE LATTICE BGK MODEL

In this paper, we use the 9-bit incompressible lattice Boltzmann model [13] to illustrate the implementation of the lattice Boltzmann method on nonuniform mesh grids. In the incompressible lattice Boltzmann model, the pressure distribution function, p_α, instead of the single-particle distribution function, f_α, is used in the evolution equation of the system. The evolution equation of the system is

$$p_\alpha(x + e_\alpha\delta_t, t + \delta_t) - p_\alpha(x, t) = \frac{1}{\tau}[g_\alpha(x, t) - p_\alpha(x, t)], \quad (1)$$

where τ is the dimensionless relaxation time and g_α is the equilibrium pressure distribution function. The e_α's, $\alpha \in \{0, 1, 2, ..., 8\}$, are the nine discrete velocities in the system defined by

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

In the above equation, $c = \delta_x/\delta_t$, where δ_x and δ_t are the lattice separation and the time step, respectively. The pressure distribution function, p_α, is related to the density distribution function, f_α, by

$$p_\alpha \equiv c_s^2 f_\alpha,$$

where c_s is the sound speed and it is $c/\sqrt{3}$ for the 9-bit LB model. The equilibrium pressure distribution, g_α, for the 9-bit incompressible LB model is chosen to be

$$g_\alpha = w_\alpha[p + \rho_0((e_\alpha \cdot u) + \frac{3}{2}\frac{(e_\alpha \cdot u)^2}{c^2} - \frac{1}{2}u^2)], \quad (3)$$

where

$$w_\alpha = \begin{cases} \frac{4}{9}, & \alpha = 0, \\ \frac{1}{9}, & \alpha = 1,2,3,4, \\ \frac{1}{36}, & \alpha = 5,6,7,8, \end{cases}$$

and ρ_0 is the constant average density in the system. The pressure, p, and the velocity, u, in g_α are calculated by

$$p = \sum_\alpha p_\alpha, \quad (4a)$$

$$u = \frac{1}{\rho_0 c_s^2}\sum_\alpha e_\alpha p_\alpha. \quad (4b)$$

Through the Chapman–Enskog procedure, the above model, Eqs. (1)–(4), recovers to the incompressible Navier–Stokes equations

$$\frac{1}{\rho_0 c_s^2}\frac{\partial p}{\partial t} + \nabla \cdot u = 0, \quad (5a)$$

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\frac{1}{\rho_0}\nabla p + \nu\nabla^2 u, \quad (5b)$$

where the kinetic viscosity is

$$\nu = \frac{(2\tau - 1)}{6}\frac{\delta_x^2}{\delta_t}.$$

In the small Mach number limit (equivalent to the incompressible limit), the first term in the left-hand side of the continuity equation, Eq. (5a), is negligible. Thus, the incompressible Navier–Stokes equations are obtained (see Ref. [13] for details of the incompressible lattice Boltzmann model).

III. LATTICE BOLTZMANN MODEL FOR NONUNIFORM GRID

In existing LB models, the computational mesh is restricted to a uniform one and the lattice separation is equal to the basic advection length, δ_x—the minimal distance that the density distributions are allowed to travel in a given time step, δ_t. This constraint is fundamental and cannot be relaxed for the LGA models. However, for LB models, this constraint becomes unnecessary because the density distributions, p_α or f_α, are continuous functions in both space and time. The value of a function at one location in space can always be approximately interpolated from the values of the function at the neighboring locations. The assumption here is that the function must be sufficiently smooth (differentiable up to the order consistent with the interpolation order).

In what follows, we shall consider a nonuniform rectangular computational mesh. Let $X_{i,j} \equiv (X_i, Y_j)$ denote a grid point on an arbitrary rectangular computational mesh in a Cartesian coordinate system. The dX_i and dY_j denote grid sizes

$$dX_i = X_{i+1} - X_i, \tag{6a}$$

$$dY_j = Y_{j+1} - Y_j. \tag{6b}$$

The computational grid can be sparser or denser than the regular lattice mesh, depending on the required flow resolution. In the following text, we define the ratio of nonuniform to uniform grid sizes as

$$r_x^i = dX_i/\delta_x, \tag{7a}$$

$$r_y^j = dY_j/\delta_y. \tag{7b}$$

For the 9-bit LB model in 2D space, the mesh is a uniform square lattice grid with $\delta_x = \delta_y$. The evolution of the LB system on a nonuniform mesh consists of three steps: collision, advection, and interpolation. The initial values of p_α (or f_α) are specified at each grid point (X_i, Y_j). Then, the system evolves in the following three steps:

1. Since the pre-collision value $p_\alpha(X_{i,j}, t)$ is known on each grid point (X_i, Y_j), the macroscopic quantities, such as p and u, can be computed at each point (X_i, Y_j). Also, the equilibrium distribution function, g_α, can be constructed on each grid point (X_i, Y_j), and the collision process can be completed according to Eq. (1). The postcollision value of $p_\alpha(X_{i,j}, t)$ is henceforth obtained;

2. After the collision, advection takes place, and the $p_\alpha(X_{i,j} + e_\alpha \delta_t, t + \delta_t)$ are obtained;

3. The values of $p_\alpha(\{X_{i,j}\}, t + \delta_t)$ on the mesh grids $\{X_{i,j}\}$ are computed from the values of $p_\alpha(\{X_{i,j} + e_\alpha \delta_t\}, t + \delta_t)$ on the points $\{X_{i,j} + e_\alpha \delta_t\}$ by interpolation. Then the collision and the advection process are repeated.

So far as the evolution process is concerned, the only distinction of the new algorithm to other existing LB algorithms is the addition of the new interpolation step.

IV. NUMERICAL RESULTS

A. *Flow in a 2D Symmetric Channel with Sudden Expansion*

The flow in a symmetric channel with sudden expansion is a hydrodynamic system with an array of interesting phenomena, including symmetry breaking bifurcations, Hopf bifurcations, and a specific route to turbulence [15]. The system has been studied both experimentally [16–18, 15, 19] and numerically [15, 20, 21]. Although the system possesses complicated hydrodynamic phenomena, the geometry of this system is very simple. It is relatively easy to generate a mesh for this system. A 2D channel with an expansion ratio of 1 : 3 and an aspect ratio of 1 : 8 is studied here. It has been shown experimentally that, at a moderate value of Reynolds number, the flow in the channel with such a geometric configuration is two-dimensional [15]. The 9-bit BGK model for incompressible flow is used to simulate the 2D flow in the system. The geometric configuration of the 2D channel with sudden expansion is illustrated in Fig. 1.

The Reynolds number for the system is defined as

$$\mathrm{Re} = \frac{hU_0}{2\nu}, \tag{8}$$

where h is the height of the entry section, U_0 is the maximum inlet velocity, and ν is the kinematic viscosity. The Reynolds number is chosen to be 26.0 for our simulations.

B. *Boundary and Initial Conditions*

At the entrance (upstream), a parabolic profile of the horizontal component of velocity, u_x, with a maximum $U_0 = 0.1c$, is enforced, and the vertical component of velocity, u_y, is set to zero. At the exit (downstream), a constant pressure boundary condition

$$p(x = N_x, t) = 1.0\rho_0 c^2$$

is enforced. At the walls, a nonslip boundary condition is applied [14].

The average density, ρ_0, is set to be 1.0. And the initial value of the velocity field is set to be zero in the interior of the channel.

C. *Meshes*

Three types of meshes are used in the simulations: uniform square mesh grid (Fig. 1a), uniform rectangular mesh grid (Fig. 1b), and nonuniform rectangular mesh grid (Fig. 1c). The system size of the square mesh grid is $N_x \times N_y = 385 \times 49$. For the uniform square mesh (lattice), the simulation is carried out without any interpolation, i.e., $r_x = r_y = 1$. The results obtained on the uniform square mesh grid are used to benchmark the results obtained on the other two meshes involving interpolation schemes.

Three uniform rectangular meshes were used in the simulation. The system sizes of the three meshes are $N_x \times N_y = 193 \times 49$, 97×49, and 49×49, corresponding to $r_x = 2$, 4, and 8, respectively. In all meshes, r_y is fixed to be unity, which is the same as in the previous LB methods. Fig. 1b illustrates the mesh grids with $r_x = 8$, the coarsest one among the rectangular meshes.

For the nonuniform grid system, the system size of the mesh is $N_x \times N_y = 43 \times 49$ (Fig. 1c). Throughout the entire mesh, r_y is always equal to 1, while r_x varies monotonically from 1 to 32 along the flow direction from upstream to downstream

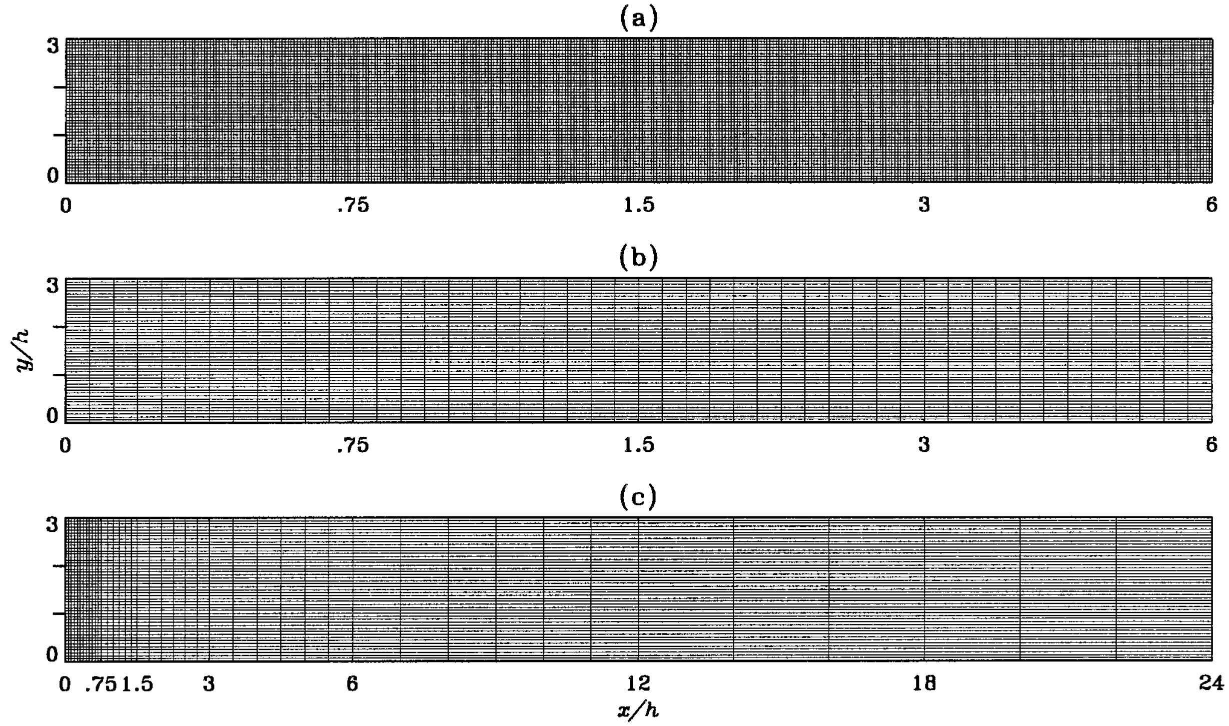

FIG. 1. Three meshes used in the simulations: (a) uniform square lattice mesh for the 9-bit LB model; (b) uniform rectangular mesh; (c) nonuniform rectangular mesh.

$$\begin{aligned} r_x &= 1, && 0.0 \le x/h \le 0.75,\\ r_x &= 2, && 0.75 < x/h \le 1.5,\\ r_x &= 4, && 1.5 < x/h \le 3.0,\\ r_x &= 8, && 3.0 < x/h \le 6.0,\\ r_x &= 16, && 6.0 < x/h \le 12.0,\\ r_x &= 32, && 12.0 < x/h \le 24.0. \end{aligned} \qquad (9)$$

Because the value of r_x monotonically increases streamwise along the channel, the upstream mesh grids are much denser than the downstream mesh grids.

The quadratic interpolation is used in the simulations unless indicated otherwise. The linear interpolation scheme is also tested against the quadratic interpolation to study the effect due to different interpolation schemes.

With the above configuration of the channel, $h = 16\delta_x$, and the Reynolds number is

$$\mathrm{Re} = \frac{4.8}{2\tau - 1}.$$

For Re = 26.0, $\tau \approx 0.5923$.

D. *Results of Simulations*

All our simulations were conducted on a cluster of IBM RISC/6000 560 workstations at Los Alamos National Laboratory. In each run, the following inequality is used as the criterion of convergence,

$$\frac{\sum_{i,j} \|u(X_{i,j}, t + \delta_t) - u(X_{i,j}, t)\|}{\sum_{i,j} \|u(X_{i,j}, t)\|} \le 1.0 \times 10^{-6}, \qquad (10)$$

where $\|\cdot\|$ is the L_2 norm. Typically, it takes about 50,000 time steps of iteration to achieve the above criterion.

Figure 2 shows the profiles of u_x at $x/h = 2.5$, and 5.0, respectively. The profiles obtained with three different meshes are compared against each other. The results are also compared with the experimental data of Ref. [15]. As shown in the figure, the numerical results consistently agree with each other very well. Also, our numerical results are in good agreement with the experimental data in Ref. [15]. The maximum relative error between the numerical results and the experimental results is 8.0%. The error can be attributed to, in part, the asymmetry of the experimental results in Ref. [15].

Figure 3 shows the profiles of u_x along the centerline of the channel obtained using three different meshes. The quadratic interpolation scheme was used for the uniform and nonuniform rectangular meshes. The results are also compared with the experimental data in Ref. [15]. As shown in the figure, the numerical results (with square, uniform, and nonuniform rectangular mesh grids) agree

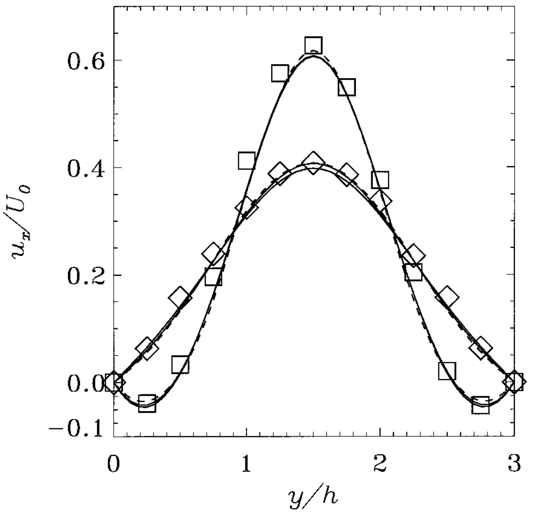

FIG. 2. The velocity profiles of u_x (normalized by U_0) of the flow in the 2D symmetric channel with sudden expansion at Re = 26. The profiles are measured at two cross sections of the channel: $x/h = 2.5$ in the recirculating region, and $x/h = 5.0$ out of the recirculating region, respectively. The thick solid line, the dashed line, and the thin solid line, represent the numerical results of the LB simulations on the square lattice mesh grid, the uniform rectangular mesh grid ($r_x = 8$), and the nonuniform mesh grid, respectively. The symbols (the squares for $x/h = 2.5$, and the diamonds for $x/h = 5.0$) represent the experimental data in Ref. [15].

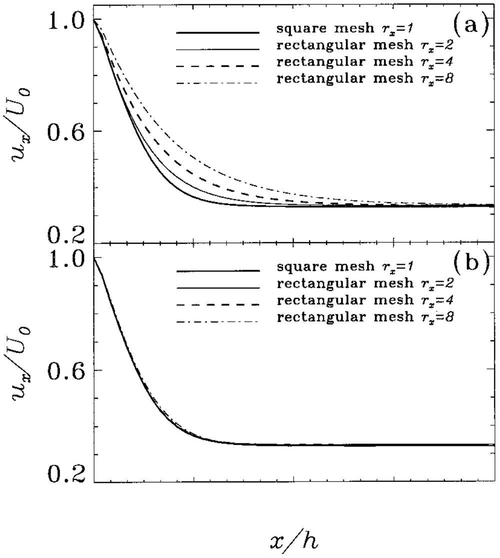

FIG. 4. The velocity u_x (normalized by U_0) at the symmetric axis (the center line) of the channel, the same as Fig. 3, with various grid sizes and interpolation schemes. The grid sizes used are $r_x = 1, 2, 4$, and 8 times of the regular lattice grid size of the normal lattice grid. ($r_x = 1$ is the square lattice mesh, and no interpolation is used in this case.) The interpolation schemes applied in the simulations are (a) linear interpolation and (b) quadratic interpolation, respectively.

with each other very well, and they are in good agreement with the experimental results.

In order to study the effects of grid size and interpolation, a set of calculations with different grid sizes and interpolation schemes were conducted. Figure 4 shows the profiles of u_x along the centerline of the channel on uniform rectangular meshes of different grid sizes with both linear and quadratic interpolations. For the rectangular meshes, r_y is equal to 1, and r_x is equal to 2, 4, and 8, respectively. All the results are compared with the result obtained using the 9-bit LB algorithm on the uniform square mesh grids ($r_x = r_y = 1$), in which no interpolation is used. Figures 4a and 4b show the results of the linear and quadratic interpolation, respectively. It is clear that the error due to linear interpolation is significant, as shown in Fig. 4a. The error grows as the grid size increases. The error introduced by the quadratic interpolation is small and certainly not significant for all grid sizes.

Table I shows the relative global differences of velocity field (for the flow in 2D channel with sudden expansion) due to grid size and interpolation scheme. This difference

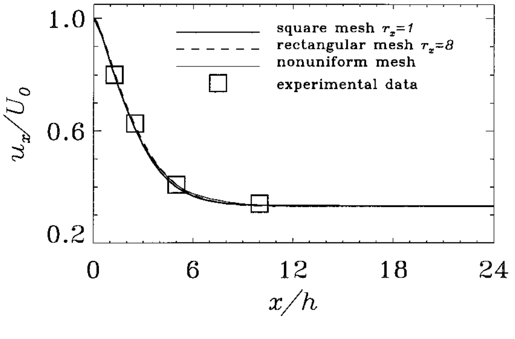

FIG. 3. The velocity profile of u_x (normalized by U_0) along the symmetric axis (the center line) of the channel, at Re = 26.0. The lines represent the results by the LB simulations with various meshes, the same as in the Fig. 2. The squares represent the experimental data in Ref. [15].

TABLE I

The Relative Global Difference of the Velocity Fields Obtained with Various Meshes and Interpolation Schemes, Defined by Eq. (11), for the Simulations of the Flow in 2-D Symmetric Channel with Expansion. The Difference Is Tabulated vs Different Meshes and Interpolation Schemes.

	Interpolation scheme	
Mesh	Linear	Quadratic
$r_x = 2$	3.17%	0.20%
$r_x = 4$	6.90%	0.55%
$r_x = 8$	11.98%	1.68%
Nonuniform	10.74%	0.80%

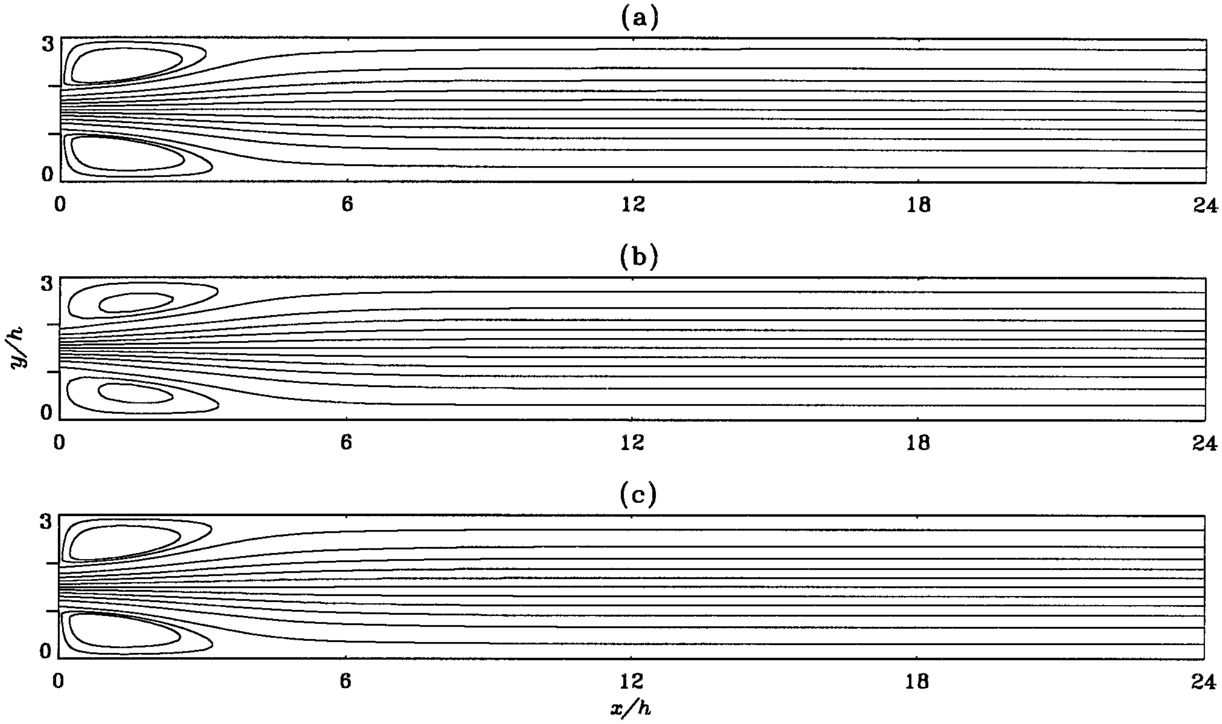

FIG. 5. The contour lines of the stream function for the flow in the 2D symmetric channel with sudden expansion at Reynolds number Re = 26.0. The values of the contours are 0.0, ±0.1, ±0.2, ±0.3, ±0.4, ±0.5, ±0.535, and ±0.55, respectively. The stream function is set to be zero along the symmetric axis of the channel. The simulation is done by (a) the 9-bit LB model with uniform square lattice, system size 385 × 49; (b) the proposed new algorithm with uniform rectangular mesh, 49 × 49; and (c) the proposed new algorithm with nonuniform rectangular mesh, 43 × 49.

is computed in reference to the result of uniform square lattice mesh ($r_x = r_y = 1$) simulation

$$\frac{\sum_{i,j}\|u(X_{i,j},t) - u_0(X_{i,j},t)\|}{\sum_{i,j}\|u_0(X_{i,j},t)\|}, \tag{11}$$

where u_0 and u are the velocity field computed with square lattice mesh and other types of rectangular meshes, respectively, and $\|\cdot\|$ is the L_2 norm. Table I quantitatively shows that the quadratic interpolation is sufficient for the LB algorithm. The difference is less than 1.0% except the case of $r_x = 8$. In contrast, linear interpolation introduces much larger errors.

Finally, Fig. 5 shows the contour lines of the stream function for the flow in the symmetric channel with sudden expansion at Re = 26.0, obtained using the three different meshes. The stream function is set to be zero at the symmetry axis of the channel. In the recirculating region near entrance, there are visible differences between the contour lines for the uniform rectangular mesh with $r_x = 8$ and $r_y = 1$ (Fig. 5b) and those for other two meshes. Obviously the difference is due to the coarseness of the mesh grids. That is, if the mesh is refined in the recirculation area, the difference should disappear. This is verified in the case of nonuniform mesh. In contrast to the case with uniform rectangular mesh, there is no significant difference between the contour lines for the nonuniform rectangular mesh grids and those for the uniform square mesh grids, even though there is almost one order of magnitude difference in their system sizes (43 × 49 vs 385 × 49). It should be pointed out that because the stream function is an integral of the velocity field, the error in the velocity is cumulative and the stream function is more sensitive to numerical error than the velocity field.

V. DISCUSSION AND CONCLUSION

We have proposed a new algorithm for implementing the lattice Boltzmann method on arbitrary nonuniform mesh grids. This has been done by introduction of an additional interpolation step—a type of ‘‘reconstruction step’’ used in many other numerical schemes (e.g., [22]). The new algorithm retains all the advantages of the previous lattice Boltzmann method, such as parallelism of the method, ease of the programming, and the capability to incorporate microscopic interactions. In addition, it overcomes inflexibility of the uniform mesh grids associated with the previous lattice Boltzmann algorithms. Numerical simulations have verified the accuracy and robustness of the new algorithm.

Since the proposed LB algorithm uses interpolation, it is worthwhile to discuss the additional numerical error that the interpolation step may bring. The numerical accuracy of linear interpolation is of the first order in grid size (in terms of dX and dY), which is one order less accurate than the LB method. The accuracy of the LB method is of second order in space (in terms of δ_x or δ_y). Therefore the numerical error introduced by the linear interpolation is significant, and it is the dominant numerical error when $dX, dY \gg \delta_x, \delta_y$. Whereas the accuracy of quadratic interpolation is of the second order in grid size, which is comparable to the accuracy of the LB method itself (second order in space). Therefore, quadratic interpolation is sufficient for this LB algorithm. In addition to that, the interpolation distance may also have an impact on the numerical viscosity. To reduce the numerical error, it will be helpful to vary the interpolation distance to be small where velocity gradients are large. Our numerical studies have verified these facts in simulations of flow in the 2D symmetric channel with sudden expansion, as indicated in Table I.

Although we only used rectangular grids in all the meshes, including the nonuniform one, it should be stressed that the algorithm is not limited to the rectangular mesh grids. It should be valid for arbitrary mesh grids. The rectangular mesh was chosen in our simulations only for the sake of simplicity, since the generation of a complex nonuniform mesh can be a very difficult task by itself. Furthermore, we use Cartesian coordinates in our analysis. Again, there is no reason this should be a restriction. Any coordinate system can be used in the analysis or the implementation of the algorithm.

Finally, our results of numerical simulation of the flow in a symmetric channel with a sudden expansion agree with both experimental and previous numerical results in Ref. [15] very well. Because the results in Ref. [15] are not directly accessible to us, the data points were digitized from the experimental data in Fig. 4 in Ref. [15]. This would inevitably bring some human errors.

REFERENCES

1. G. McNamara and G. Zanetti, *Phys. Rev. Lett.* **61,** 2332 (1988).
2. F. J. Higuera, S. Succi, and R. Benzi, *Europhys. Lett.* **9,** 345 (1989); F. J. Higuera and J. Jeménez, *Europhys. Lett.* **9,** 663 (1989).
3. H. Chen, S. Chen, and W. H. Matthaeus, *Phys. Rev. A* **45,** R5339 (1991); Y. H. Qan, D. d'Humières, and P. Lallemand, *Europhys. Lett.* **17,** 479 (1992).
4. G. D. Doolen (Ed.), *Lattice Gas Methods for Partial Differential Equations* (Addison–Wesley, Redwood City, CA, 1990).
5. R. Benzi, S. Succi, and M. Vergassola, *Phys. Rep.* **222,** 145 (1992).
6. X. Shan and H. Chen, *Phys. Rev. E* **47,** 1815 (1993); **49,** 2941 (1994); M. R. Swift, W. R. Osborn, and J. M. Yeoman, *Phys. Rev. Lett.* **75,** 830 (1995).
7. A. J. C. Ladd, *J. Fluid Mech.* **271,** 285 (1994); **271,** 311 (1994).
8. S. Chen H. Chen, D. Martínez, and W. Matthaeus, *Phys. Rev. Lett.* **67,** 3776 (1991).
9. S. Chen *et al., Comput. Chem. Eng.* **19,** 617 (1995).
10. A. K. Gunstensen, Ph.D. thesis, MIT, June 1992 (unpublished).
11. U. Frisch, B. Hasslacher, and Y. Pomeau, *Phys. Rev. Lett.* **56,** 1505 (1986); S. Wolfram, *J. Stat. Phys.* **45,** 471 (1986).
12. P. L. Bhatnagar, E. P. Gross, and M. Krook, *Phys. Rev.* **94,** 511 (1954).
13. X. He and L.-S. Luo, *J. Stat. Phys.,* submitted.
14. X. He, Q. Zou, and L.-S. Luo, *J. Stat. Phys.,* submitted.
15. R. M. Fearn, T. Mullin, and K. A. Cliffe, *J. Fluid Mech.* **211,** 595 (1990).
16. F. Durst, A. Melling, and J. H. Whitelaw, *J. Fluid Mech.* **64,** 111 (1974).
17. W. Cherdron, F. Durst, and J. H. Whitelaw, *J. Fluid Mech.* **84,** 13 (1978).
18. I. Sobey, *J. Fluid Mech.* **151,** 395 (1985).
19. F. Durst, J. C. F. Pereira, and C. Tropea, *J. Fluid Mech.* **248,** 567 (1993).
20. I. Sobey and P. G. Drazin, *J. Fluid Mech.* **171,** 263 (1986).
21. L. Kaiktsis, preprint, 1992.
22. K. Xu *et al.,* AIAA Paper 96-0525, 1996 (unpublished).