

Lattice Boltzmann method on quadtree grids

Yu Chen,^{1,2,3} Qinjun Kang,³ Qingdong Cai,^{1,*} and Dongxiao Zhang^{2,4}

¹*The State Key Laboratory for Turbulence and Complex Systems (LTCS), Center for Applied Physics and Technology (CAPT), and Department of Mechanics and Aerospace Engineering, Peking University, Beijing, People's Republic of China*

²*Sonny Astani Department of Civil and Environmental Engineering, University of Southern California, Los Angeles, California 90089, USA*

³*Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

⁴*Department of Energy and Resources Engineering, Peking University, Beijing, People's Republic of China*

(Received 27 January 2010; revised manuscript received 17 October 2010; published 28 February 2011)

A lattice Boltzmann method on nonuniform quadtree grids is proposed. Our method employs the interpolation-supplemented lattice Boltzmann model. The advantages of the quadtree grid are preserved by using linear interpolation instead of quadratic interpolation to complete the streaming step in the lattice Boltzmann method. The back-and-forth error compensation and correction (BFEC) method is used to improve the accuracy, so that the second-order accuracy of the conventional lattice Boltzmann method is maintained. Several numerical cases, including a BFEC streaming test, lid-driven cavity flow, and flow over an asymmetrically placed cylinder in a channel, are carried out to demonstrate the accuracy and efficiency of our method.

DOI: [10.1103/PhysRevE.83.026707](https://doi.org/10.1103/PhysRevE.83.026707)

PACS number(s): 47.11.Qr

I. INTRODUCTION

The lattice Boltzmann (LB) method, which originated in the late 1980's, has become a powerful numerical tool to model complex flow and multiphysicochemical transport processes [1,2], such as multiphase and multicomponent flows [3–5], flow in porous media [6–9], turbulence [10–12], fluid-particle suspensions [13–17], and reactive transport [18,19]. The success of the LB method is mainly owing to the fact that it is based on microscopic models and mesoscopic kinetic equations, which gives the LB method the advantage of studying complex flows, especially in fluid flow applications involving interfacial dynamics and complex boundaries [1].

One drawback to the conventional LB method is that it is constrained on a special class of uniform and regular lattices [1], limiting its numerical efficiency when there is a need for a high-resolution grid in high gradient flow regions, near a curved solid body, or when there is a far-field boundary condition. To increase numerical efficiency and accuracy, nonuniform grid methods have been developed [20–29], which makes the LB method suitable for many practical applications [12,30–32].

He *et al.* [20] proposed an interpolation-supplemented LB model for nonuniform grids, in which the collision step still takes place on the grid points, while the streaming step is supplemented by a quadratic interpolation procedure in order to maintain the second-order accuracy. The lattice time step is defined by the fine lattice and fixed in the whole computation domain.

Filippova *et al.* [21] proposed a LB local grid refinement method. Each level of the grid has its own lattice spacing and lattice time increment. Nonequilibrium parts of the particle distribution functions (PDFs) between coarse and fine grids are scaled in order to keep constant viscosity in the whole computation domain. The LB equation is first solved on the coarse grid of the entire computation domain, and then the boundary conditions of the fine grid are interpolated from

the postscaled PDFs on the coarse grid using a second-order interpolation scheme.

Based on the idea of Filippova [21], Yu *et al.* [22] proposed a multiblock method, in which the grid blocks are not overlapped with each other, but are connected only through the interface. The information exchanged on the interfaces requires special treatment to ensure mass and momentum conservation between the blocks. A cubic spline is adopted to interpolate information from coarse grid to fine grid in order to eliminate the possible spatial asymmetry.

Quadtree [quadtree in two dimensions (2D) and octree in three dimensions (3D)] is a hierarchical data structure, based on the principle of recursive decomposition [33,34]. Quadtree has been used to generate finite-element meshes [35–37]. The advantage of the quadtree grid is that the grid can be generated automatically and dynamically based on certain criteria without complicated algorithms [35–38]. Recently, a LB method was proposed using quadtree grids [39,40]. Crouse *et al.* [39] adopted the method of Filippova [21] and used linear interpolation in the interface treatment, thus the second-order accuracy may not be maintained. Geier *et al.* [41] and Tölke *et al.* [42] proposed to use linear interpolation of second-order moments to achieve second-order accuracy. In this paper, we present a LB method on nonuniform quadtree grids based on the interpolation-supplemented LB model [20]. While only linear interpolation is used, the second-order accuracy is obtained by using the back-and-forth error compensation and correction (BFEC) method [43–45]. The basic idea has been briefly presented in Ref. [40]. In this study, we further enhance the method and include new results with detailed descriptions.

The rest of the paper is organized as follows. Section II describes the numerical method, including generation of quadtree grids, and a brief review of the LB method and the BFEC method. Section III provides some numerical results to validate our method. Section IV concludes the paper.

II. NUMERICAL METHOD

A. Quadtree grid

The primary units of quadtree grids are quadtree cells. Each quadtree cell is composed of four grid nodes. Geometric

*caiqd@pku.edu.cn

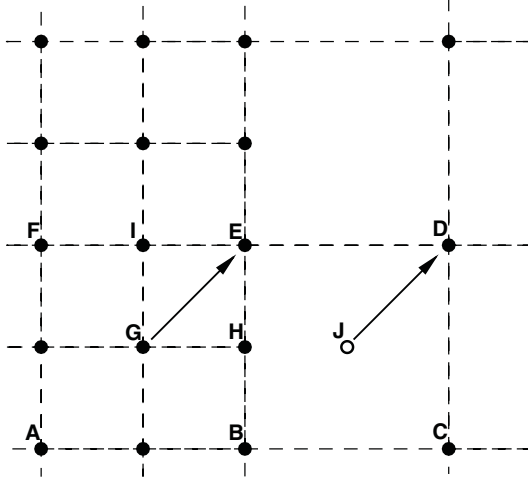


FIG. 1. Cells of a quadtree mesh: ABEF is a parent cell and GHEI is a leaf cell.

information and physical quantities are stored on those grid nodes. Also, the calculation of all physical quantities is performed on these grid nodes. There are two different types of quadtree cells—the parent and leaf cell (see Fig. 1). A parent cell has four children and is equally divided into four subcells. Thus the cell stores the pointers to its four subcells. A leaf cell has no children. Thus the cell stores the pointers to its four grid nodes. The octree grid in 3D is similar to the quadtree grid, but with the parent cell equally divided into eight subcells instead of four. In this paper, we mainly focus on the quadtree grid, however, there is no major technical obstacle to extend our method to the octree grid.

In the quadtree grid, a scale function is often used to control grid density. As an example, we construct a nonuniform quadtree grid on a 1×1 square domain based on the scale function values, as shown in Fig. 2. Here, for simplicity, the scale function is predefined and fixed, while in other cases it can be assigned by certain criterion such as local velocity gradient and can be dynamically changed with time. The procedure is as follows:

Step 1. Set the four corners of the square domain (point 1, 2, 3, 4) as the first four grid nodes that compose the root quadtree cell.

Step 2. Compare the scale function value ($S1$) of the center point (and other characteristic points inside the cell, if necessary) of the current cell with the length ($S2$) of the current cell. If $S1 < S2$, then equally divide the current cell into four subcells.

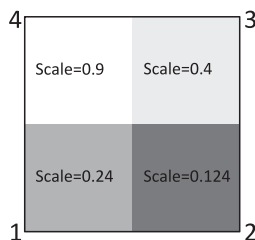


FIG. 2. A unit square domain with a predefined scale function value.

Step 3. Repeat step 2 until the length of each leaf cell is smaller than corresponding scale function value.

Figure 3 illustrates the complete procedure for generating a quadtree grid from the unit square domain shown in Fig. 2. In Fig. 3, divide 1, the scale function at the center point (0.5, 0.5) of cell 1234, is not well defined or has multiple values. In this case, we also check the scale function value at other characteristic points inside cell 1234, such as point (0.25, 0.25). If the scale function value at any of these points is smaller than $S2$, the division will be performed. Because $S2 = 1$ for cell 1234 and $S1 = 0.24$ at point (0.25, 0.25), $S1 < S2$. Then divide 1 takes place. Similarly, divides 2–4 occur subsequently.

If the flow domain is not square, one can just cover the flow domain with a square and delete those grid nodes outside of the flow domain. With a quadtree data structure, refining or coarsening a cell requires a trivial change of the pointers in the tree structures [38].

Figure 3 is for demonstration only. In practical use, in order to simplify the calculations required at the cell boundaries and to ensure smooth transition of the macroscopic variables across the interface of different grid levels, only one tree level difference is adopted between quadtree cells and their neighbors [37,46].

B. Lattice Boltzmann method

Unlike the conventional computational fluid dynamics (CFD) methods that solve the Navier-Stokes equation to obtain the macroscopic quantities of flow field, the LB method is based on microscopic models and mesoscopic kinetic equations [1]. The main variables in the LB method are the PDFs.

It has been shown that the LB equation with a single relaxation time approximation, the so-called lattice Bhatnagar-Gross-Krook (LBGK) model,

$$f_i(\mathbf{x} + \mathbf{e}_i \delta t, t + \delta t) = f_i(\mathbf{x}, t) - \frac{f_i(\mathbf{x}, t) - f_i^{\text{eq}}(\mathbf{x}, t)}{\tau}, \quad (1)$$

can be derived from the following continuous Boltzmann equation with the BGK collision [1,47],

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla f = -\frac{1}{\tau}(f - f^{\text{eq}}). \quad (2)$$

In Eq. (1) f_i is the PDF along the i th direction, f_i^{eq} is the corresponding equilibrium distribution function, \mathbf{e}_i is the i th discrete velocity, δt is the time increment, and τ is the relaxation time, which relates to the kinematic viscosity by

$$\nu = (\tau - 1/2)c_s^2 \delta t, \quad (3)$$

where c_s is the speed of sound.

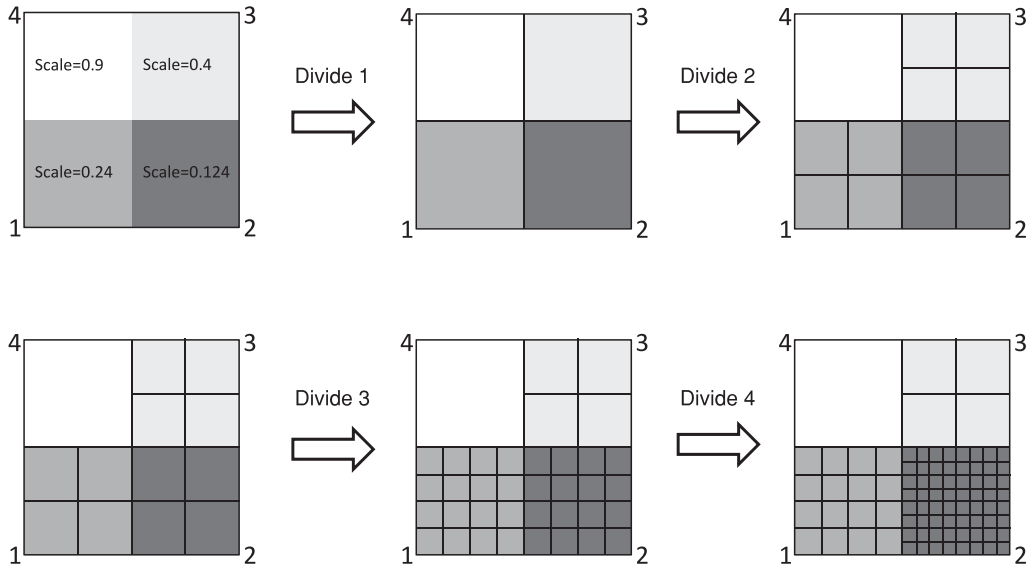


FIG. 3. Procedure for generating a quadtree grid.

For the 2D, nine-speed LB model (D2Q9) [47] (as shown in Fig. 4), we have

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

where $c = \delta x / \delta t$ and δx is the lattice spacing. The equilibrium distribution is

$$f_i^{\text{eq}}(\rho, \mathbf{u}) = \rho w_i \left[1 + \frac{\mathbf{e}_i \cdot \mathbf{u}}{3c^2} + \frac{9(\mathbf{e}_i \cdot \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u} \cdot \mathbf{u}}{2c^2} \right], \quad (5)$$

where $w_0 = 4/9$; $w_i = 1/9$, for $i = 1,3,5,7$; $w_i = 1/36$, for $i = 2,4,6,8$.

The macroscopic density and momentum density are defined as

$$\rho = \sum_i f_i, \quad (6)$$

$$\rho \mathbf{u} = \sum_i f_i \mathbf{e}_i. \quad (7)$$

The pressure is calculated through an equation of state:

$$p = c_s^2 \rho. \quad (8)$$

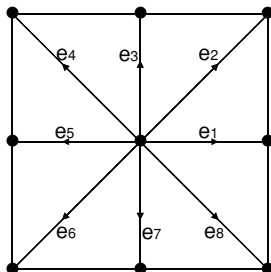


FIG. 4. D2Q9 model.

Equation (8) shows that the LB method is not a strictly incompressible model. The incompressible Navier-Stokes equations can be obtained in the nearly incompressible limit [1]. When used for incompressible flows, the LB method must be viewed as an artificial compressibility method [48]. The density variation and Mach number of the fluid must be very small for an accurate simulation of incompressible flows. These requirements may limit the application of conventional LB models for practical problems, such as high Reynolds number flow or flow through porous media [49].

In order to minimize the compressibility effect of the LB method, we employ the incompressible LBGK model

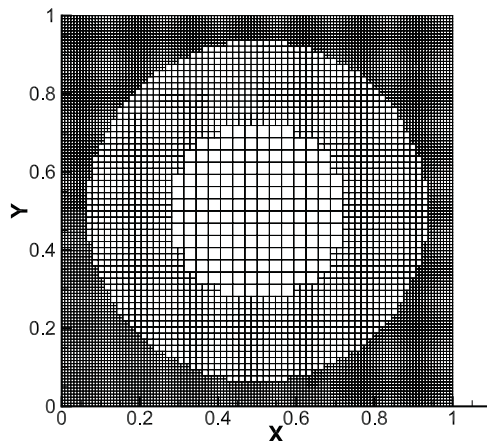


FIG. 5. Quadtree grid in a square domain with high grid density near the edge.

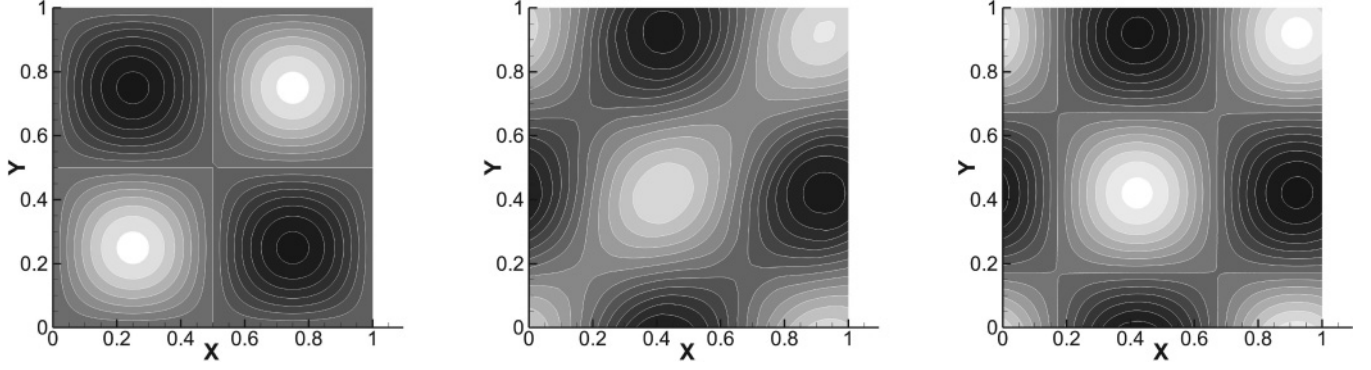


FIG. 6. From left to right, the contours of the initial testing values, and the contours of the testing values on Q257 grid after 300 time steps streaming in direction 2 without BFECC and with BFECC, respectively.

proposed by Guo *et al.* [48]. This incompressible model is capable of simulating unsteady flow.

In Guo's model, the equilibrium distribution function is defined as

$$f_i^{(\text{eq})} = \begin{cases} -4\sigma \frac{p}{c^2} + s_i(\mathbf{u}), & i = 0, \\ \lambda \frac{p}{c^2} + s_i(\mathbf{u}), & i = 1, 3, 5, 7, \\ \gamma \frac{p}{c^2} + s_i(\mathbf{u}), & i = 2, 4, 6, 8, \end{cases} \quad (9)$$

where σ , λ , and γ are parameters satisfying

$$\lambda + \gamma = \sigma, \quad (10)$$

$$\lambda + 2\gamma = \frac{1}{2}, \quad (11)$$

and $s_i(\mathbf{u})$ is defined by

$$s_i(\mathbf{u}) = w_i \left[\frac{\mathbf{e}_i \cdot \mathbf{u}}{3c^2} + \frac{9(\mathbf{e}_i \cdot \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u} \cdot \mathbf{u}}{2c^2} \right]. \quad (12)$$

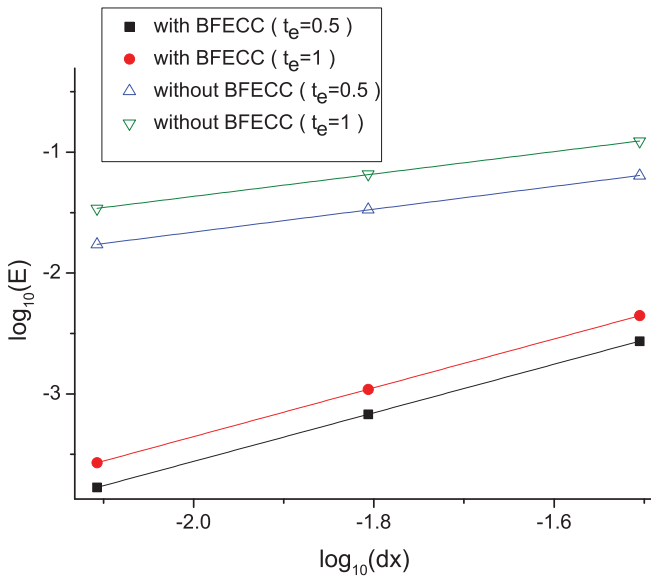


FIG. 7. (Color online) Relative error for the streaming test in direction 1 at $t_e = 0.5$ and $t_e = 1$. The slopes of the lines are 1.98 (with BFECC, $t_e = 0.5$), 1.99 (with BFECC, $t_e = 1$), 0.89 (without BFECC, $t_e = 0.5$), and 0.84 (without BFECC, $t_e = 1$).

Finally, the macroscopic velocity and pressure are defined as

$$\mathbf{u} = \sum_{i=1}^8 f_i \mathbf{e}_i, \quad (13)$$

$$p = \frac{c^2}{4\sigma} \left[\sum_{i=1}^8 f_i + s_0(\mathbf{u}) \right]. \quad (14)$$

Readers may refer to Ref. [48] for details of the incompressible model.

Equation (1) can be broken up into the following two steps:

(1) Collision step:

$$f_i(\mathbf{x}, t + \delta_t) = f_i(\mathbf{x}, t) - \frac{f_i(\mathbf{x}, t) - f_i^{\text{eq}}(\mathbf{x}, t)}{\tau}. \quad (15)$$

(2) Streaming step:

$$f_i(\mathbf{x} + \mathbf{e}_i \delta_t, t + \delta_t) = f_i(\mathbf{x}, t + \delta_t). \quad (16)$$

The collision step only involves the PDFs of the local lattice node, while the streaming step involves the PDFs of

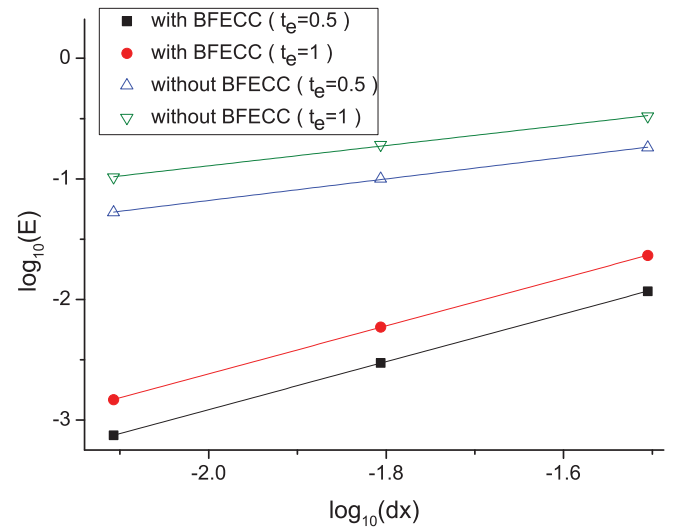


FIG. 8. (Color online) Relative error for the streaming test in direction 2 at $t_e = 0.5$ and $t_e = 1$. The slopes of the lines are 2.01 (with BFECC, $t_e = 0.5$), 2.02 (with BFECC, $t_e = 1$), 0.95 (without BFECC, $t_e = 0.5$), and 0.93 (without BFECC, $t_e = 1$).

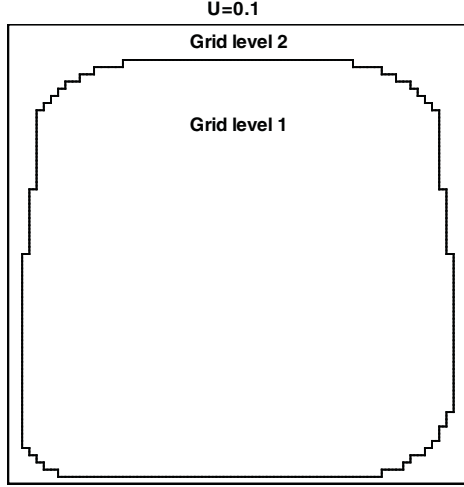


FIG. 9. Grid distribution of the lid-driven cavity flow case.

neighboring nodes. Thus, for a nonuniform quadtree grid, the streaming step requires additional treatment.

Unlike the lattice gas cellular automata model, where the particle occupation is defined by a set of Boolean variables, PDFs of the LB model are real variables. Hence we can calculate the PDFs at one location from its neighboring locations through interpolation [20]. In Fig. 1, both cell GHEI and cell BCDE are leaf cells (cells without child cells) of a quadtree grid. Assume cell GHEI is the smallest cell, then the length of its edge is equal to the lattice spacing. In a streaming step, node E will get information from node G, while node D will get information from point J, which is not a grid node. Thus, we need to obtain the information at point J from neighboring nodes. Considering the data structure of a quadtree grid, it is more convenient and efficient to use linear interpolation of nodes B and D, which are in the same quadtree cell than quadratic interpolation that requires information from other quadtree cells. This, however, will lead to first-order accuracy only. In order to maintain the second-order accuracy of the LB method [1], we adopt the BFECC method.

C. BFECC method

BFECC stands for back-and-forth error compensation and correction, which can be used to reduce dissipation and diffusion encountered in a variety of advection steps, such as velocity, smoke density, and image advection on uniform and adaptive grids and on a triangulated surface. The BFECC method can be implemented easily as a small modification of the first-order upwind or semi-Lagrangian integration of advection equations and is proved to be second-order accurate in both space and time [43–45].

Let function ϕ satisfy the following convection equation:

$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0, \quad (17)$$

where \mathbf{u} represents a velocity field. The procedure of BFECC is

$$\tilde{\phi}^{n+1} = L(\phi^n, \mathbf{u}), \quad (18)$$

$$\phi_1^n = L(\tilde{\phi}^{n+1}, -\mathbf{u}), \quad (19)$$

$$\phi_2^n = \phi^n + \frac{1}{2}(\phi^n - \phi_1^n), \quad (20)$$

$$\phi^{n+1} = L(\phi_2^n, \mathbf{u}), \quad (21)$$

where L is the first-order upwind or semi-Lagrangian integration operator to integrate Eq. (19), and ϕ^n is the value of ϕ at time n . Comparing Eq. (2) with Eq. (19), it should be straightforward to use the BFECC method to reduce the error caused by linear interpolation during the streaming step of LB method on a nonuniform quadtree grid.

Denote the postcollision PDF as \tilde{f}_i . Below is the implementation of the BFECC method in the LB streaming step:

Step 1. Carry out a normal LB streaming step for \tilde{f}_i . Denote current PDF as f_i^* .

Step 2. Carry out an inverse LB streaming step for f_i^* , i.e., a normal LB streaming but with opposite lattice velocity, as indicated in Eq. (19). Denote current PDF as f_i^{**} .

Step 3. Let $f_i^{***} = 1.5\tilde{f}_i - 0.5f_i^{**}$.

Step 4. Carry out a normal LB streaming step for f_i^{***} . Denote current PDF as f_i . Then f_i is the poststreaming PDF.

The BFECC procedure is only performed at a coarse grid where interpolation is required to complete the streaming step.

III. NUMERICAL EXAMPLES

A. BFECC method in streaming step

To validate that the BFECC method reduces the error caused by linear interpolation during the streaming step of the LB method on nonuniform quadtree grids, we designed the following numerical example:

(1) Generate a nonuniform quadtree mesh on a 1×1 square domain with high grid density near the edge of the square, as shown in Fig. 5, of which the smallest grid spacing is equal to 0.007 812 5. We denote this mesh as Q129.

(2) Define a periodic test function:

$$F(x, y, t) = 10 \sin[2\pi(x - \mathbf{e}_{ix}t)] \sin[2\pi(y - \mathbf{e}_{iy}t)], \quad (22)$$

where \mathbf{e}_{ix} and \mathbf{e}_{iy} are the x and y components of lattice velocity \mathbf{e}_i . $\mathbf{e}_{ix} = 1$ and $\mathbf{e}_{iy} = 0$ for lattice direction 1, and $\mathbf{e}_{ix} = 1$ and $\mathbf{e}_{iy} = 1$ for lattice direction 2, as shown in Fig. 4. When $t = 0$, assign each quadtree grid node with the value of function F as the initial value.

(3) Carry out a single direction (such as lattice direction 1 or 2, see Fig. 4) streaming step of LB method for each time step. Apply periodic boundary conditions at the square edge.

TABLE I. The position of the primary vortex center.

	Present work (Q129)	Present work (Q257)	Ghia <i>et al.</i> [50]	Hou <i>et al.</i> [51]	Yu <i>et al.</i> [22]
x	0.6154	0.6155	0.6172	0.6196	0.6172
y	0.7382	0.7375	0.7344	0.7373	0.7390

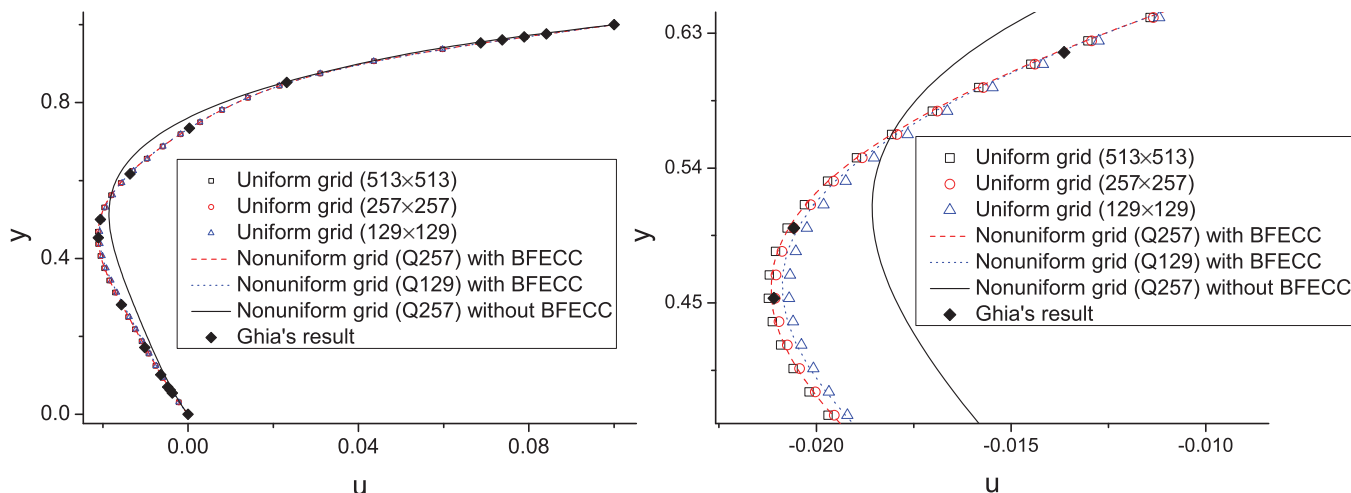


FIG. 10. (Color online) The x-direction velocity profile at line $x = 0.5$.

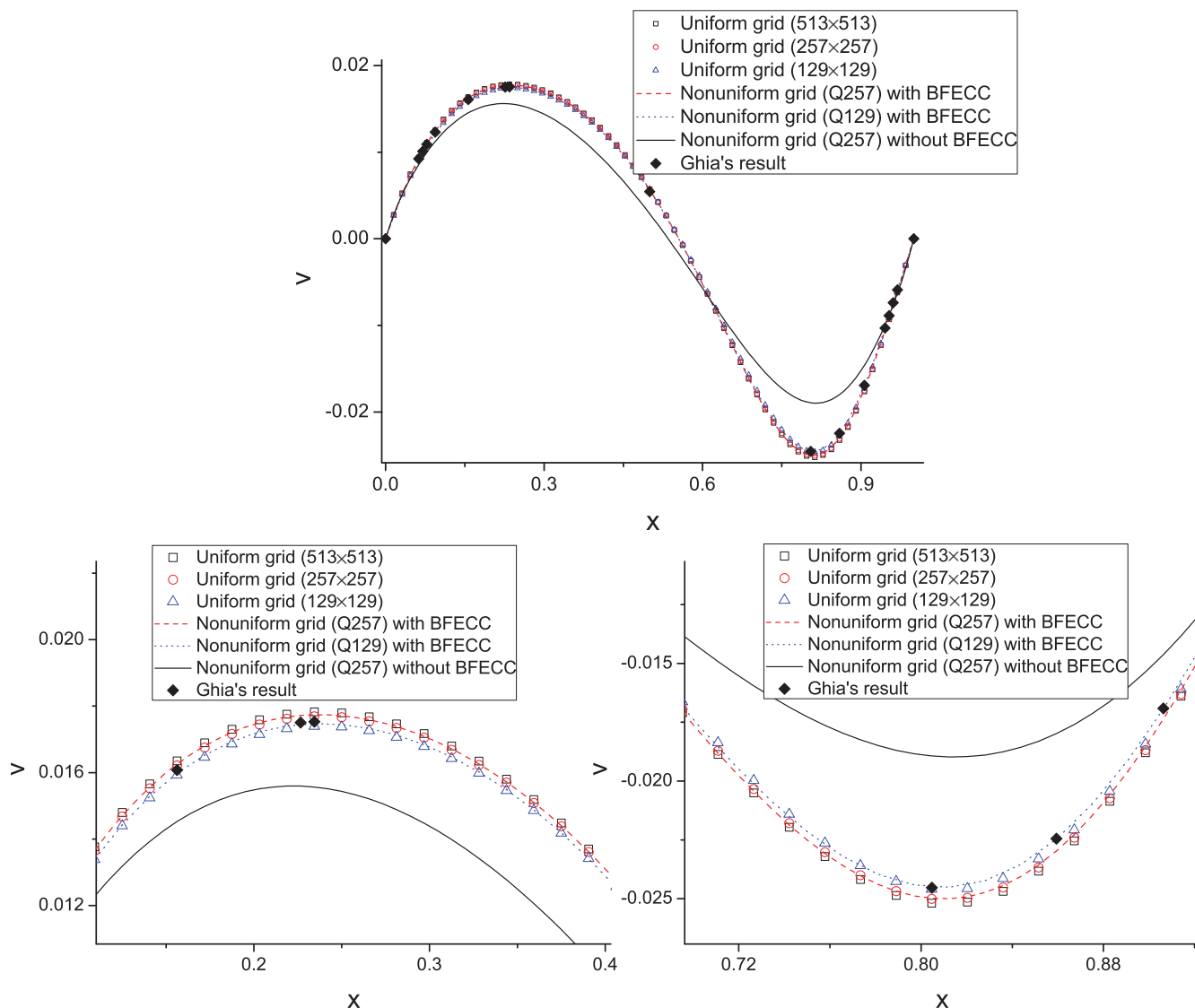


FIG. 11. (Color online) The y-direction velocity profile at line $y = 0.5$.

TABLE II. Global relative errors for the lid-driven cavity flow based on the result obtained from a 513×513 uniform grid.

	Q129	129×129	Q257	257×257
Global relative error	4.563×10^{-2}	4.555×10^{-2}	1.531×10^{-2}	1.454×10^{-2}

(4) Repeat step 3 until $t = t_e$.

Then refine the mesh by dividing each leaf cell of current quadtree mesh into four sub-cells, and we get the nonuniform meshes Q257 and Q513. Repeat the above steps 2, 3, and 4 for Q257 and Q513 meshes. Compare the numerical value at each node with the exact solution F . Let E denote the global relative error at $t = t_e$. E is defined as

$$E = \frac{\sum_{i \in \{\text{leaf cell}\}}^n \sum_{k=1}^4 |f_{ik} - F_{ik}| S_i}{\sum_{i \in \{\text{leaf cell}\}}^n \sum_{k=1}^4 |F_{ik}| S_i}, \quad (23)$$

where n is the total number of quadtree cells, subscript ik denotes the k th child node of leaf cell i , and S_i is the area of cell i . The results are shown in Figs. 6–8. We find the streaming step of the LB method on quadtree grids with BFECC is indeed second-order accurate.

B. 2D lid-driven cavity flow

The lid-driven cavity flow is a classic benchmark case to test numerical schemes for fluid flows [50,51]. The top lid of the 2D unit square cavity moves from left to right with constant speed $U_0 = 0.1$, as shown in Fig. 9. The fluid physics is governed by the Reynolds number given by $\text{Re} = U_0 L / \nu$, where L ($L = 1$) is the length of the top boundary and ν is the kinematic viscosity of the fluid. In order to eliminate numerical errors caused by different relaxation times, we keep $\tau = 0.8$ for different grids.

1. Nonuniform quadtree grid

In order to accurately treat the moving wall and capture vortices near the corners of the walls, we generate a nonuniform quadtree grid on the unit square domain, with the grid distribution shown in Fig. 9. Two grid levels are used here. The grid spacing of level 2 is equal to 0.078 125, the lattice spacing of the 129×129 uniform grid, and the grid spacing of level 1 is twice of level 2. We denote this quadtree grid as Q129. Then a finer grid (denoted as Q257) is obtained by dividing each leaf cell of Q129 into four subcells, of which the level 2 grid spacing is equal to the lattice spacing of the 257×257 uniform grid. There are a total of 6946 grid nodes and 27 033 grid nodes for the Q129 and Q257 grids, respectively, and a total of 16 641 grid nodes and 66 049 grid nodes for the 129×129 and 257×257 uniform grids, respectively.

2. Boundary conditions

The popular bounce-back scheme is only first-order accurate and cannot handle moving boundaries accurately. Thus, we employ the nonequilibrium extrapolation boundary scheme of Guo *et al.* [52,53] for the nonslip velocity boundary conditions at the walls. The basic idea is to decompose the distribution function at the boundary node into its equilibrium and nonequilibrium parts, and then to approximate the

nonequilibrium part with a first-order extrapolation of its counterpart at the neighboring fluid nodes [52]. Two corners of the top moving wall are assigned with the velocity of the moving wall.

3. Results

At $\text{Re} = 100$, the center of the primary vortex is at (0.6154, 0.7382) for the Q129 grid and (0.6155, 0.7375) for the Q257 grid, respectively, which are in good agreement with the results from the finite-difference method and other LB simulations, as shown in Table I.

Figures 10 and 11 show the u -component and v -component velocity profiles at the vertical and horizontal lines that cross the cavity center, respectively. We can see that the results on quadtree grids with BFECC match well with those obtained from the LB method on uniform grids. Without BFECC, the results of the LB method on quadtree grids significantly deviate from other results, which indicates that linear interpolation without BFECC in the streaming step results in significant numerical error.

Being absent an analytical solution, we also perform the simulation on a 513×513 uniform grid and take the result as a reference. Then the global relative error for Q129 and Q257 is defined as

$$E = \frac{\sum_{i \in \{\text{leaf cell}\}}^n \sum_{k=1}^4 |U_{ik} - U_{eik}| S_i}{\sum_{i \in \{\text{leaf cell}\}}^n \sum_{k=1}^4 |U_{eik}| S_i} + \frac{\sum_{i \in \{\text{leaf cell}\}}^n \sum_{k=1}^4 |V_{ik} - V_{eik}| S_i}{\sum_{i \in \{\text{leaf cell}\}}^n \sum_{k=1}^4 |V_{eik}| S_i}, \quad (24)$$

where n is the total number of quadtree cells, subscript ik denotes the k th child node of leaf cell i , S_i is the area of cell i , and U_e and V_e are the x -direction velocity and y -direction velocities obtained from the 513×513 uniform grid, respectively. Table II shows that the relative errors of LBM on nonuniform quadtree grids remain at the same level of corresponding uniform grids.

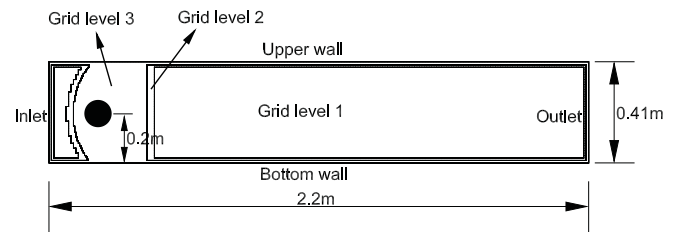


FIG. 12. Grid distribution for the case of flow over an asymmetrically placed cylinder in a channel.

TABLE III. Results for $Re = 20$.

	c_{Dmax}	c_{Lmax}	St
With BFEC	5.5864	0.0104	0
Without BFEC	5.6969	0.0075	0
Results from Ref. [54]			
Upper bound	5.5700	0.0104	0
Lower bound	5.5900	0.0110	0

C. Flow over an asymmetrically placed cylinder in a channel

Flow over an asymmetrically placed cylinder in a channel is a widely tested benchmark case [21,22,54], as shown in Fig. 12. According to Ref. [54], constant parabolic velocity is specified on the inlet boundary and constant pressure is specified on the outlet boundary. The Reynolds number is defined as $Re = \bar{U}D/\nu$, where \bar{U} is the mean velocity at the channel inlet, D is the cylinder diameter, and ν is the kinematic viscosity of the fluid. In this case, the maximum velocity at the inlet is set to 0.1. The drag (c_D) and lift (c_L) coefficients of the cylinder, as well as the Strouhal (St) number, are computed from flow field and compared with the results of Ref. [54]. The definitions of c_D , c_L , and St are as follows:

$$c_D = \frac{2F_x}{\rho\bar{U}^2 D}, \quad (25)$$

$$c_L = \frac{2F_y}{\rho\bar{U}^2 D}, \quad (26)$$

$$St = \frac{Df}{\bar{U}}, \quad (27)$$

where F_x and F_y are the drag force and lift force on the cylinder, respectively, and f is the frequency of separation. Readers can refer to Ref. [54] for a detailed description.

1. Nonuniform quadtree grid

Flow region around the cylinder is vital to calculate the drag or lift coefficient, while downstream flow far from the cylinder has little effect on the flow field near the cylinder. Based on these considerations, we generate a quadtree grid with three grid levels, as shown in Fig. 12. The level 1 grid is the coarsest grid and is far from the cylinder and boundaries. The level 3 grid is the finest grid that encloses the cylinder and the channel boundaries. The lattice spacing is equal to level 3 grid spacing. There are 38 714 grid nodes for the nonuniform Quadtree grid, and there are 146 246 grid nodes for the normal uniform grid. The ratio of grid nodes between nonuniform and uniform grids is 1 : 3.8.

TABLE IV. Results for $Re = 100$.

	c_{Dmax}	c_{Lmax}	St
With BFEC	3.2396	1.0032	0.2993
Without BFEC	3.1212	-0.0136	0
Results from Ref. [54]			
Upper bound	3.2400	1.0100	0.3050
Lower bound	3.2200	0.9900	0.2950

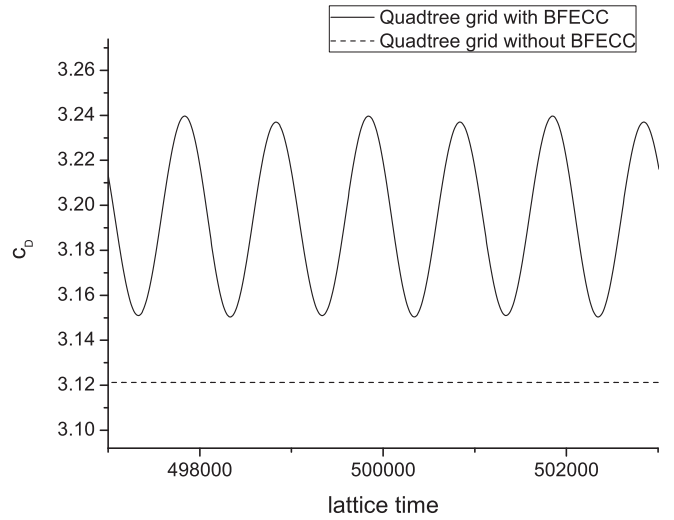


FIG. 13. Time evolution of drag coefficient for flow over an asymmetrically placed cylinder at $Re = 100$ obtained from LB method on quadtree grid with and without BFEC.

2. Boundary conditions and force evaluation

We use the halfway bounce-back scheme for the nonslip velocity condition at the upper and bottom walls, and employ the nonequilibrium extrapolation boundary scheme of Guo [52,53] for the velocity condition at the inlet and the pressure condition at the outlet. In order to accurately treat the curved surface of the cylinder, we adopt the curved boundary condition proposed by Mei *et al.* [55]. The momentum exchange scheme [2,14,15] is used to evaluate the fluid force on the cylinder.

3. Results

Both the $Re = 20$ steady case and $Re = 100$ unsteady case are tested using a nonuniform quadtree grid. As shown in Tables III and IV, our BFEC results match well with those from Ref. [54], while non-BFEC results deviate from

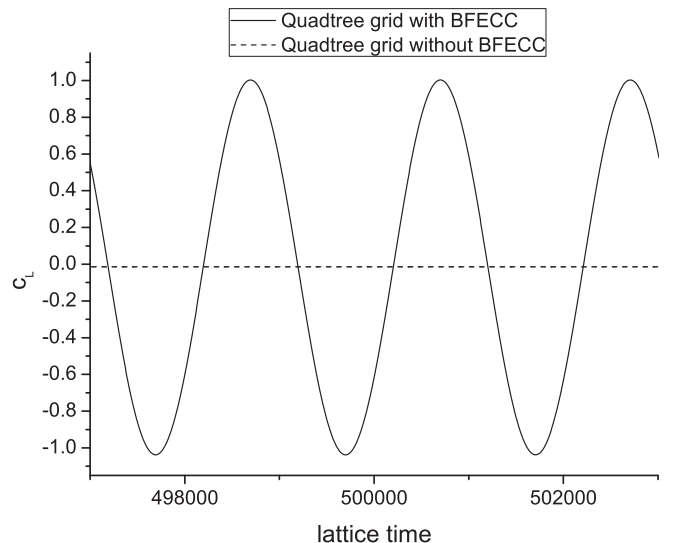


FIG. 14. Time evolution of lift coefficient for flow over an asymmetrically placed cylinder at $Re = 100$ obtained from the LB method on a quadtree grid with and without BFEC.

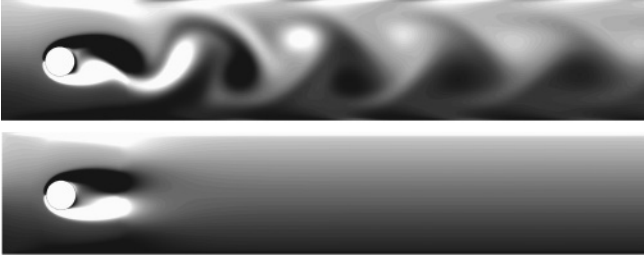


FIG. 15. Vorticity contours of flow over an asymmetrically placed cylinder at $Re = 100$ with (top) and without BFECC (bottom), respectively, after 600 000 time steps.

other results. Without BFECC the results of $Re = 20$ deviate less from reference results than those of the $Re = 100$ non-BFECC case. This is mainly because that for $Re = 100$, the flow is unsteady and vortices keep shedding from the cylinder. More accurate schemes must be used to capture these vortices. Otherwise, the numerical diffusion caused by linear interpolation without BFECC will lead to a steady-state solution without vortex shedding ($St = 0$). Figures 13 and 14 show the c_D curve and c_L curve of the $Re = 100$ case using the quadtree grid with and without BFECC. Figure 15 shows the vorticity contours after 600 000 time steps. The von Kármán vortex street disappears in the case without BFECC. Figure 16 shows the contours of the shear strain rate for the $Re = 100$ unsteady case, which are smooth across the interface of different grid levels.

The computation time for 2000 streaming-collision cycles of the $Re = 20$ case is shown in Table V. The computation is performed on a server configured with an Intel Xeon E5450 central processing unit (CPU). The ratio of computation time between nonuniform quadtree grid and normal uniform grid is 1 : 1.6. This ratio is greater than that of the grid nodes (1 : 3.8). One reason is that the linear interpolation and BFECC procedures take some time. The other reason is that the quadtree data structure cannot be as efficient as the structured squared grid used in the standard LBM code. To separate these two factors, we compare the computation time between the normal uniform and uniform quadtree grids. The ratio is 1 : 1.5. This ratio could be decreased by better optimizing the code. As a result, the calculation on the nonuniform quadtree grid may run faster.

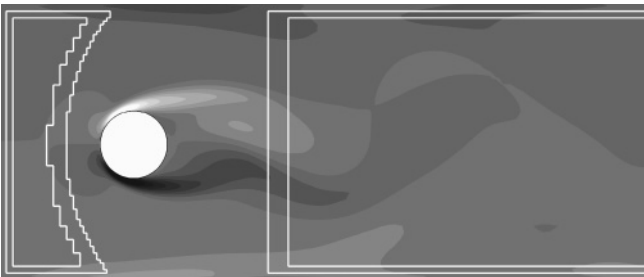


FIG. 16. Contours of shear strain rate for flow over an asymmetrically placed cylinder at $Re = 100$ obtained from LB simulation on a quadtree grid with BFECC.

TABLE V. Computation time for 2000 streaming-collision cycles using different grids for $Re = 20$.

	Nonuniform Quadtree grid	Normal uniform grid	Uniform Quadtree grid
Computation time (s)	30	48	72

IV. CONCLUSIONS AND DISCUSSIONS

A LB method on nonuniform quadtree grids was developed and proved to be accurate and efficient in the benchmark numerical simulations. Our method takes advantage of the quadtree grid and He's interpolation-supplemented LB model [20], but without the need for quadratic interpolation (which will be more complicated in unstructured grids than in structured grids). Instead, the second-order accuracy is achieved by using the BFECC method. Unlike the methods of Filippova [21] and Yu [22], there is no interface treatment between different grid levels. The relaxation time, lattice spacing, and time increment are uniform in the whole domain in our method, hence the method is easier to implement on dynamic adaptive grids. Moreover, although a square lattice is used in our current work, our method can be readily extended to rectangular or even triangular quadtree grids.

On the other hand, as He's method, interpolation is carried out for all leaf cells except for the smallest ones whose edge length is equal to lattice spacing. Although we use linear interpolation to improve efficiency and BFECC to improve accuracy, the numerical diffusion cannot be completely eliminated, especially in very coarse grid regions. In multiblock or multiply nested grids methods, the streaming step in each grid level is exact, and interpolation only takes place on the interface of different grids, although there are some additional steps to treat the interface and to synchronize quantities on different grid levels. In the case of flow over an asymmetrically placed cylinder in a channel, the total number of grid nodes in our method cannot be as low as Yu's [22], otherwise the numerical diffusion will become significant.

However, our method has advantages for cases that have complex geometry or flow pattern and require dynamic adaptive mesh. The work combining our method with the unified LB model proposed by Kang *et al.* [7] to simulate flows in complex multiscale porous media is ongoing and will be presented in future publications.

ACKNOWLEDGMENTS

The authors would like to thank Zhenhua Xia and Dr. Kevin Connington for helpful discussions and for providing the LB code to treat curved boundaries. We would also like to thank Dr. Mark Porter for reviewing this paper and providing useful comments. This work is supported by the National Natural Science Foundation of China (No. 10872005) and LANL's LDRD program (20100025DR). Yu's academic visit at USC is supported by the China Scholarship Council (No. 2008601145).

- [1] S. Chen and G. Doolen, *Annu. Rev. Fluid Mech.* **30**, 329 (1998).
- [2] D. Yu, R. Mei, L. Luo, and W. Shyy, *Prog. Aerosp. Sci.* **39**, 329 (2003).
- [3] A. Gunstensen, D. Rothman, S. Zaleski, and G. Zanetti, *Phys. Rev. A* **43**, 4320 (1991).
- [4] X. Shan and H. Chen, *Phys. Rev. E* **47**, 1815 (1993).
- [5] M. Swift, W. Osborn, and J. Yeomans, *Phys. Rev. Lett.* **75**, 830 (1995).
- [6] A. Gunstensen and D. Rothman, *J. Geophys. Res.* **98**, 6431 (1993).
- [7] Q. Kang, D. Zhang, and S. Chen, *Phys. Rev. E* **66**, 056307 (2002).
- [8] D. Zhang, R. Zhang, S. Chen, and W. Soll, *Geophys. Res. Lett.* **27**, 1195 (2000).
- [9] Z. Guo and T. S. Zhao, *Phys. Rev. E* **66**, 036304 (2002).
- [10] D. Martínez, W. Matthaeus, S. Chen, and D. Montgomery, *Phys. Fluids* **6**, 1285 (1994).
- [11] H. Chen, S. Kandasamy, S. Orszag, R. Shock, S. Succi, and V. Yakhot, *Science* **301**, 633 (2003).
- [12] Z. Lu, Y. Liao, D. Qian, J. McLaughlin, J. Derksen, and K. Kontomaris, *J. Comput. Phys.* **181**, 675 (2002).
- [13] A. Ladd and R. Verberg, *J. Stat. Phys.* **104**, 1191 (2001).
- [14] A. Ladd, *J. Fluid Mech.* **271**, 285 (1994).
- [15] A. Ladd, *J. Fluid Mech.* **271**, 311 (1994).
- [16] Z. Xia, K. Connington, S. Rapaka, P. Yue, J. Feng, and S. Chen, *J. Fluid Mech.* **625**, 249 (2009).
- [17] K. Connington, Q. Kang, H. Viswanathan, A. Abdel-Fattah, and S. Chen, *Phys. Fluids* **21**, 053301 (2009).
- [18] Q. Kang, P. Lichtner, and D. Zhang, *J. Geophys. Res.* **111**, B05203 (2006).
- [19] Q. Kang, P. Lichtner, and D. Zhang, *Water Resour. Res.* **43**, W12S14 (2007).
- [20] X. He, L. Luo, and M. Dembo, *J. Comput. Phys.* **129**, 357 (1996).
- [21] O. Filippova and D. Hänel, *J. Comput. Phys.* **147**, 219 (1998).
- [22] D. Yu, R. Mei, and W. Shyy, *Int. J. Numer. Methods Fluids* **39**, 99 (2002).
- [23] F. Nannelli and S. Succi, *J. Stat. Phys.* **68**, 401 (1992).
- [24] N. Cao, S. Chen, and D. Martinez, *Phys. Rev. E* **55**, 21 (1997).
- [25] T. Lee and C. Lin, *J. Comput. Phys.* **171**, 336 (2001).
- [26] S. Ubertini and S. Succi, *Prog. Comput. Fluid Dyn.* **5**, 85 (2005).
- [27] N. Rossi, S. Ubertini, G. Bella *et al.*, *Int. J. Numer. Methods Fluids* **49**, 619 (2005).
- [28] Y. S. Li, E. J. LeBoeuf, and P. K. Basu, *Phys. Rev. E* **72**, 046711 (2005).
- [29] M. Stiebler, J. Tolke, and M. Krafczyk, *Comput. Fluids* **35**, 814 (2006).
- [30] H. Dixit and V. Babu, *Int. J. Heat Mass Transf.* **49**, 727 (2006).
- [31] Y. Sui, Y. Chew, P. Roy, and H. Low, *Int. J. Numer. Methods Fluids* **53**, 1727 (2007).
- [32] D. Yu and S. Girimaji, *Physica A (Amsterdam)* **362**, 118 (2006).
- [33] H. Samet, *Comput. Surv.* **16**, 187 (1984).
- [34] A. Klinger and C. Dyer, *Comput. Graphics Image Process.* **5**, 68 (1976).
- [35] M. Yerry and M. Shephard, *IEEE Comput. Graphics Applic.* **3**, 39 (1991).
- [36] M. Shephard and M. Georges, *Int. J. Numer. Methods Eng.* **32**, 709 (1983).
- [37] P. Baehmann, S. Wittchen, M. Shephard, K. Grice, and M. Yerry, *Int. J. Numer. Methods Eng.* **24**, 1043 (1987).
- [38] Z. Wang, *Comput. Fluids* **27**, 529 (1998).
- [39] B. Crouse, E. Rank, M. Krafczyk, and J. Tölke, *Int. J. Numer. Mod. Phys. B* **17**, 109 (2003).
- [40] Y. Chen and Q. Cai, *Mod. Phys. Lett. B* **23**, 289 (2009).
- [41] M. Geier, A. Greiner, and J. Korvink, *Eur. Phys. J. Spec. Top.* **171**, 173 (2009).
- [42] J. Tölke and M. Krafczyk, *Comput. Math. Appl.* **58**, 898 (2009).
- [43] T. Dupont and Y. Liu, *J. Comput. Phys.* **190**, 311 (2003).
- [44] T. Dupont and Y. Liu, *Math. Comp.* **76**, 647 (2007).
- [45] B. Kim, Y. Liu, I. Llamas, and J. Rossignac, *IEEE Trans. Visualization Comput. Graphics* **13**, 135 (2007).
- [46] S. Popinet, *J. Comput. Phys.* **190**, 572 (2003).
- [47] Y. Qian, D. Humieres, and P. Lallemand, *Europhys. Lett.* **17**, 479 (1992).
- [48] Z. Guo, B. Shi, and N. Wang, *J. Comput. Phys.* **165**, 288 (2000).
- [49] Q. Kang, P. Lichtner, H. Viswanathan, and A. Abdel-Fattah, *Transp. Porous Media* **82**, 197 (2009).
- [50] U. Ghia, K. Ghia, and C. Shin, *J. Comput. Phys.* **48**, 387 (1982).
- [51] S. Hou, Q. Zou, S. Chen, G. Doolen, and A. Cogley, *J. Comput. Phys.* **118**, 329 (1995).
- [52] Z. Guo, C. Zheng, and B. Shi, *Chin. Phys.* **11**, 366 (2002).
- [53] Z. Guo, C. Zheng, and B. Shi, *Phys. Fluids* **14**, 2007 (2002).
- [54] M. Schäfer and S. Turek, *Numer. Fluid Mech.* **52**, 547 (1996).
- [55] R. Mei, L. Luo, and W. Shyy, *J. Comput. Phys.* **155**, 307 (1999).