Improved treatments for general boundary conditions in the lattice Boltzmann method for convection-diffusion and heat transfer processes

Qing Chen,[1,2] Xiaobing Zhang,[1,*] and Junfeng Zhang[2,†]

[1]*School of Energy and Power Engineering, Nanjing University of Science and Technology, Jiangsu 210094, China*

[2]*Bharti School of Engineering, Laurentian University, 935 Ramsey Lake Road, Sudbury, Ontario P3E 2C6, Canada*

(Received 11 April 2013; revised manuscript received 9 August 2013; published 26 September 2013)

In spite of the increasing applications of the lattice Boltzmann method (LBM) in simulating various flow and transport systems in recent years, complex boundary conditions for the convection-diffusion and heat transfer processes in LBM have not been well addressed. In this paper, we propose an improved bounce-back method by using the midpoint concentration value to modify the bounced-back density distribution for LBM simulations of the concentration field. An accurate finite-difference scheme in the normal boundary direction has also been introduced for gradient boundary conditions. Compared with existing boundary methods, our method has a simple algorithm and can easily deal with boundaries with general geometries, motions, and surface conditions (the Dirichlet, Neumann, and mixed conditions). Carefully designed simulations are performed to examine the capacity and accuracy of this proposed boundary method. Simulation results are compared with those from theory and a representative boundary method, and an improved performance is observed. We have also simulated the effect of reference velocity on global accuracy to examine the performance of our model in preserving the fundamental Galilean invariance. These boundary treatments for concentration boundary conditions can be readily applied to other processes such as heat transfer systems.

DOI: 10.1103/PhysRevE.88.033304 PACS number(s): 47.11.−j, 47.10.ab, 44.05.+e

I. INTRODUCTION

The lattice Boltzmann method (LBM) has been gradually accepted as a useful simulation method for fluid flows and other associated phenomena in the past two decades [1–3]. Unlike conventional computational methods, which are based on the numerical discretization of macroscopic governing equations, LBM originates from the classical kinetic theory. Due to such kinetic particulate nature, LBM possesses several attractive features, such as the simple algorithm formulation, relative easiness in dealing with complex geometries, and potential efficiency with parallel computation. Applications of LBM simulations can be found in various flow situations, for example, multiphase flows, porous flows, compressible flows, particulate flows, biological flows, and microfluidics [1–4].

It have been shown that, via the Chapman-Enskog expansion, the macroscopic continuity and momentum (Navier-Stokes) equations can be obtained from the collision-propagation processes of particle density distributions in LBM [2]. Therefore, from a pure mathematical point of view, the LBM algorithm can be considered as a numerical solver of partial differential equations. Following this train of thought, the original LBM algorithm for fluid flows has been tuned to solve the governing differential equations for various processes and phenomena, including the electric fields, magnetic fields, porous flows, and axisymmetric flows [3]. In addition, convection-diffusion and heat transfer processes are commonly encountered in various natural and industrial situations. Early efforts in LBM simulations of such systems can be traced back to the pioneering work by Dawson *et al.* [5]. Two sets of particle distribution functions (PDFs) were employed there, with one set for the solute density and another for the solvent density. He *et al.* [6] have improved this double-PDF model by assigning one set of PDFs for the solute concentration, and another set for the flow dynamics of the total fluid (including the solute and solvent). This approach has become the standard for LBM convection-diffusion simulations. In addition, models for the multiple-relaxation-time (MRT) LBM algorithm [7], irregular lattice structures [8], and anisotropic diffusion [9,10] have also been reported. Shi *et al.* [11] have also proposed a more accurate treatment for the source term. Similar double-PDF thermal LBM models for heat transfer have also been developed and utilized in various applications [12–15]. In these models, temperature is actually considered a scalar property, and its evolution is governed by a convection-diffusion-type differential equation. For this reason, in the following sections in this paper, we will focus on the convection-diffusion processes, and all analysis and discussions are readily applicable to heat transfer systems.

As with any other numerical methods, correct and accurate boundary treatments play a crucial role in LBM simulations. However, unlike the extensive efforts in the boundary methods for LBM flow simulations, LBM boundary conditions for convection-diffusion systems have not been addressed adequately. A review on these existing boundary conditions can be found in a recent publication [16]. As one can see there, most of these existing boundary methods are limited to flat surfaces [6], stationary surfaces [17,18], or certain types of boundary conditions (i.e., the boundary-value Dirichlet or boundary-gradient Neumman boundary conditions) [19,20]. There are also several boundary methods proposed for general curved boundary geometries [21–23]; however, the complicated algorithms make them difficult to be utilized for systems with complex boundaries, such as transport processes in particulate and porous flows. In addition, these methods typically do not consider the effect of boundary movement

[*] zhangxb680504@163.com

[†] jzhang@laurentian.ca

on the concentration field. By extending the classical bounce-back scheme in LBM for fluid dynamics [24] to convection-diffusion simulations, Zhang *et al.* [16] recently have proposed a boundary method capable of dealing with general boundary situations, including the Dirichlet, Neumann, and mixed Robin boundary conditions, boundary velocity, and nonflat surfaces. Chen *et al.* [25] have argued that this bounce-back scheme is only suitable for static boundaries; however, more rigorous analysis and validations might be necessary to examine this concern in depth and details.

The half-way bounce-back scheme in LBM suffers a low resolution in representing the actual smooth boundary geometry [24]. In this paper, we improve the spatial resolution of the bounce-back scheme by Zhang *et al.* [16] with a recently developed midpoint bounce-back technique for flow [26] and an accurate finite-difference scheme for electric field [27] simulations. The key idea is that the bounced-back PDFs should be adjusted according to the velocity and concentration values at the midpoint of a boundary lattice link, instead of those on the physical boundary. Such midpoint values can be obtained via an interpolation or extrapolation along the boundary lattice direction. In addition, for the Neumann and Robin boundary conditions, when evaluating the boundary concentration value from the boundary gradient, the finite-difference scheme employed in Ref. [16] has neglected the possible surface heterogeneity (i.e., boundary condition variation over the surface). Such possible boundary heterogeneity has also been considered in our improved finite-difference scheme along the boundary normal direction. Several carefully designed systems have been simulated, and the results have been compared with those from theoretical solutions and the original method by Zhang *et al.* [16]. The Galilean invariance from our LBM model has also been examined by studying the effect of reference velocity on global error. These simulations and comparisons demonstrate that our boundary treatments have improved numerical accuracy and can be employed in convention-diffusion systems with various boundary conditions. The method described here can also be readily extended to MRT LBM models and heat transfer processes.

II. MODELS AND METHODS

For simplicity and convenience, in this paper, the model description and demonstration simulations are presented using the classical lattice Bhatnagar-Gross-Krook (LBGK) model over a D2Q9 (two-dimensional, 9-velocity) lattice structure. Extending this present work to three-dimensional systems, other lattice structures, and/or MRT schemes is straightforward.

A. The lattice Boltzmann method for convection-diffusion processes

To simulate the convection-diffusion process in an incompressible flow, two sets of PDFs are employed: one to solve the flow dynamics and another one for the concentration field. Their evolution is governed by the following well-known lattice Boltzmann equations (LBEs) with the single-relaxation-time approximation for the collision operator [1–3],

$$f_i(\mathbf{x}+\mathbf{e}_i\delta t,t+\delta t)-f_i(\mathbf{x},t)=-\frac{1}{\tau_f}\left[f_i(\mathbf{x},t)-f_i^{\mathrm{eq}}(\mathbf{x},t)\right], \tag{1}$$

$$g_i(\mathbf{x}+\mathbf{e}_i\delta t,t+\delta t)-g_i(\mathbf{x},t)=-\frac{1}{\tau_g}\left[g_i(\mathbf{x},t)-g_i^{\mathrm{eq}}(\mathbf{x},t)\right], \tag{2}$$

where $f_i(\mathbf{x},t)$ and $g_i(\mathbf{x},t)$ are the PDFs for, respectively, the flow and concentration fields at location $\mathbf{x}$ and time t and the subscript i indicates the lattice direction. The lattice velocities $\mathbf{e}_i$ for the D2Q9 lattice structure adopted in this study are given as

$$\mathbf{e}_0=(0,0); \tag{3}$$

$$\mathbf{e}_i=\left[\cos\frac{(i-1)\pi}{2},\sin\frac{(i-1)\pi}{2}\right]\frac{\delta x}{\delta t},\quad i=1-4; \tag{4}$$

$$\mathbf{e}_i=\left[\cos\frac{(2i-1)\pi}{4},\sin\frac{(2i-1)\pi}{4}\right]\frac{\sqrt{2}\delta x}{\delta t},\quad i=5-8. \tag{5}$$

Here δx is the lattice grid unit and δt is the time step. The relaxation parameters τ_f and τ_g are related to, respectively, the macroscopic fluid kinematic viscosity ν and the diffusivity D in a similar fashion as

$$\nu=\frac{(2\tau_f-1)\delta x^2}{6\delta t}, \tag{6}$$

$$D=\frac{(2\tau_g-1)\delta x^2}{6\delta t}. \tag{7}$$

The equilibrium distributions f_i^{eq} and g_i^{eq} are typically expressed as

$$f_i^{\mathrm{eq}}=\omega_i\rho\left[1+\frac{\mathbf{e}_i\cdot\mathbf{u}}{c_s^2}+\frac{(\mathbf{e}_i\cdot\mathbf{u})^2}{2c_s^4}-\frac{\mathbf{u}\cdot\mathbf{u}}{2c_s^2}\right], \tag{8}$$

$$g_i^{\mathrm{eq}}=\omega_i C\left[1+\frac{\mathbf{e}_i\cdot\mathbf{u}}{c_s^2}+\frac{(\mathbf{e}_i\cdot\mathbf{u})^2}{2c_s^4}-\frac{\mathbf{u}\cdot\mathbf{u}}{2c_s^2}\right]. \tag{9}$$

For the D2Q9 lattice structure, the lattice sound speed $c_s=\sqrt{3}\delta x/\delta t$, and the lattice weight factors are $\omega_0=4/9$, $\omega_{1-4}=1/9$, and $\omega_{5-8}=1/36$. The macroscopic properties, including the fluid density ρ, flow velocity $\mathbf{u}$, flow pressure P, and concentration C, are readily available from the PDFs as

$$\rho=\sum_i f_i,\quad \mathbf{u}=\sum_i f_i\mathbf{u}_i/\rho, \\ P=c_s^2\rho,\quad C=\sum_i g_i. \tag{10}$$

It can be shown that macroscopic flow and transport equations

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{u})=0, \tag{11}$$

$$\frac{\partial\mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\mathbf{u}=-\frac{1}{\rho}\nabla P+\frac{\mu}{\rho}\nabla^2\mathbf{u}, \tag{12}$$

$$\frac{\partial C}{\partial t}+\nabla\cdot(C\mathbf{u})=\nabla\cdot(D\nabla C), \tag{13}$$

can be derived from the above formulations via the Chapman-Enskog analysis [2].

B. The midpoint bounce-back scheme for boundary velocity and boundary concentration

Different from other numerical methods, in LBM the boundary requirements for macroscopic properties are implemented by specifying the PDFs entering the simulation region across the boundary. As we typically have more unknown incoming PDFs to determine and less macroscopic constraints available, assumptions are required to make the equation system complete. Next we briefly describe the midpoint bounce-back boundary method for the velocity boundary condition recently developed by Yin and Zhang [26] and then extend this scheme to the Dirichlet concentration boundary condition.

For the convenience of our description, we consider the solid surface Γ in the D2Q9 lattice space as illustrated in Fig. 1(a). A boundary lattice link, with two end nodes $\mathbf{x}_s$ in the solid domain and $\mathbf{x}_f$ in the fluid domain, intersects with the boundary surface Γ at the boundary node $\mathbf{x}_b$. The midpoint of this lattice link is denoted by $\mathbf{x}_m$. After the collision step at the fluid node $\mathbf{x}_f$, the PDF f_i^* [$i = 7$ in Fig. 1(a)] leaves $\mathbf{x}_f$, and is then assumed to be bounced-back at the midpoint $\mathbf{x}_m$ in the reversed direction and with a modified magnitude as $f_{\bar{i}}$ [$\bar{i} = 5$ in Fig. 1(a)]:

$$f_{\bar{i}} = f_i^* - \frac{2\omega_i \rho}{c_s^2} \mathbf{u}_m \cdot \mathbf{c}_i, \tag{14}$$

where $\mathbf{u}_m$ is the midpoint velocity at $\mathbf{x}_m$ to be determined. Different from the classical Ladd bounce-back method [24], here we are using the midpoint velocity $\mathbf{u}_m$ instead of the boundary velocity $\mathbf{u}_b$.

To find out the midpoint velocity $\mathbf{u}_m$, we denote the fraction of the boundary lattice link in the solid domain as Δ, i.e.,

$$\Delta = \left| \frac{\mathbf{x}_s - \mathbf{x}_b}{\mathbf{x}_s - \mathbf{x}_f} \right|, \tag{15}$$

and $0 \leqslant \Delta \leqslant 1$. For $\Delta \leqslant 1/2$, the midpoint $\mathbf{x}_m$ locates between $\mathbf{x}_b$ and $\mathbf{x}_f$, and the midpoint velocity $\mathbf{u}_m$ can be readily obtained with a linear interpolation:

$$\mathbf{u}_m = \frac{\frac{1}{2}\mathbf{u}_b + \left(\frac{1}{2} - \Delta\right)\mathbf{u}_f}{1 - \Delta}, \tag{16}$$

where $\mathbf{u}_b$ is the imposed boundary velocity at the intersection point $\mathbf{x}_b$, and $\mathbf{u}_f$ is the flow velocity calculated at the fluid node $\mathbf{x}_f$. For $\Delta > 1/2$, the midpoint $\mathbf{x}_m$ is in the solid domain and therefore an extrapolation is needed to obtain velocity $\mathbf{u}_m$. For a better numerical stability as in previous studies [28,29], we use $\mathbf{u}_{ff} = \mathbf{u}(\mathbf{x}_{ff})$, the velocity at the second inner fluid node $\mathbf{x}_{ff}$ (instead of the first fluid node $\mathbf{x}_f$) from the boundary Γ [Fig. 1(a)],

$$\mathbf{u}_m = \frac{\frac{3}{2}\mathbf{u}_b - \left(\Delta - \frac{1}{2}\right)\mathbf{u}_{ff}}{2 - \Delta}. \tag{17}$$

For the particular case with $\Delta = 1/2$, either Eq. (16) or Eq. (17) yields $\mathbf{u}_m = \mathbf{u}_b$, and our treatment simply reverts back to the original Ladd method.

From the above description, we see that, like in the original Ladd method [24], here the physical boundary surface Γ actually is represented by a virtual stairwise surface Γ′ [dot-dashed lines in Fig. 1(a)] in the LBM calculation. However, we use the estimated midpoint velocity $\mathbf{u}_m$ on virtual surface Γ′, instead of the boundary velocity $\mathbf{u}_b$ on the physical surface Γ, to modify the bounced-back PDF magnitude in Eq. (14); and, hence, the discrepancy between the bounce-back process at the midpoint and the PDF modification using the boundary velocity in the original Ladd method has been removed. Numerical results have shown that this midpoint bounce-back method has a good numerical accuracy [26].

For the Dirichlet concentration boundary condition, recently Zhang *et al.* [16] have extended the original Ladd approach [24] to modify the boundary PDFs g_i,

$$g_{\bar{i}} = -g_i^* + 2\omega_i C_b \left[1 + \frac{(\mathbf{e}_i \cdot \mathbf{u}_b)^2}{2c_s^4} - \frac{\mathbf{u}_b \cdot \mathbf{u}_b}{2c_s^2} \right], \tag{18}$$

where C_b is the concentration value on the boundary surface. Again, since the physical surface is replaced be a stairwise

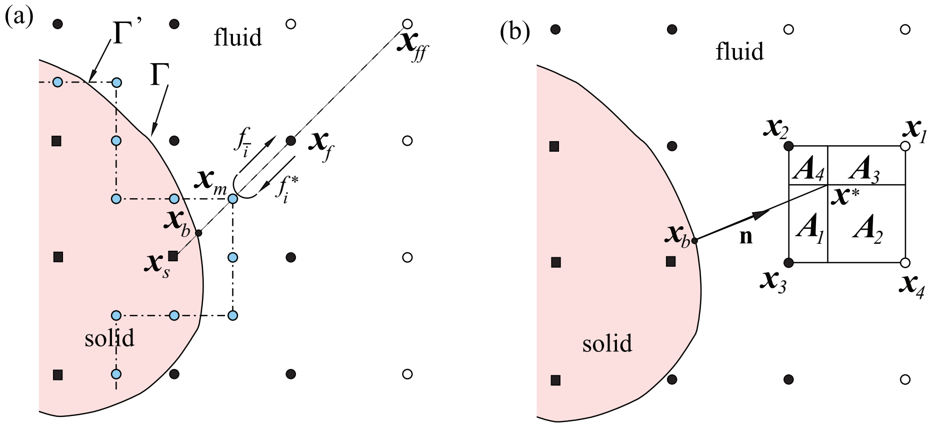

FIG. 1. (Color online) Schematic illustrations for the midpoint bounce-back scheme (a) and the normal-direction finite-difference estimation of boundary value from boundary gradient (b). See text for details.

surface along lattice link midpoints, the spatial resolution of the boundary geometry is low, especially for surface with high curvatures (for example, particles and porous structures). To improve the spatial accuracy, the midpoint bounce-back method described above for flow field can be easily applied to the Dirichlet concentration boundary condition by replacing the concentration C_b and velocity $\mathbf{u}_b$ on the physical boundary with the concentration C_m and velocity $\mathbf{u}_m$ at the midpoint $\mathbf{x}_m$ as follows:

$$g_{\bar{i}} = -g_i^* + 2\omega_i C_m \left[1 + \frac{(\mathbf{e}_i \cdot \mathbf{u}_m)^2}{2c_s^4} - \frac{\mathbf{u}_m \cdot \mathbf{u}_m}{2c_s^2} \right], \tag{19}$$

and C_m can be obtained in a similar way as that for $\mathbf{u}_m$ via an appropriate interpolation or extrapolation,

$$C_m = \frac{\frac{1}{2}C_b + \left(\frac{1}{2} - \Delta\right)C_f}{1 - \Delta}, \quad \Delta \leqslant 1/2, \tag{20}$$

$$C_m = \frac{\frac{3}{2}C_b - \left(\Delta - \frac{1}{2}\right)C_{ff}}{2 - \Delta}, \quad \Delta > 1/2. \tag{21}$$

C. The finite difference scheme for boundary concentration gradient

In chemical reaction and heat transfer systems, the gradient of the interesting macroscopic variable (C here) is often involved in defining the boundary condition [6]. However, in our above formulation for the boundary treatment, the boundary concentration value is required to adjust the boundary PDFs. A finite-difference approximation is thus necessary to obtain the required concentration value from the available concentration gradient at the boundary. In Ref. [16], the following finite-difference scheme is employed:

$$\left(\frac{\partial C}{\partial n} \right)_b \approx \frac{C_b - C_f}{(\delta x/2)\mathbf{n} \cdot \mathbf{e}_i}, \tag{22}$$

where $\mathbf{n}$ is the unit vector in the normal direction on the boundary, pointing into the simulated domain. Again here it is assumed the boundary locates at the midpoint of the boundary lattice link (indicating be the $\delta x/2$ in the denominator). In addition to this spatial inaccuracy, this scheme is conducted along the boundary lattice link direction. It should be noticed that the boundary gradient is defined in the boundary normal direction, which often differs from the boundary lattice direction (at one boundary location, there is only one normal direction but, very likely, several boundary lattice links), as shown in Fig. 1(b). The finite-different scheme in Eq. (22) is actually only correct for flat boundaries with a uniform boundary gradient.

To address these issues, here we adopt the finite-difference scheme recently developed by Oulaid and Zhang [27]. This method has considered the real boundary location, and the finite-difference approximation is performed along the boundary normal direction. Consider the situation in Fig. 1(b). To obtain the concentration value at the boundary point $\mathbf{x}_b$, first we find the concentration value at the location $\mathbf{x}^*$, which is of a distance δ from $\mathbf{x}_b$ into the simulated domain along the outward normal direction $\mathbf{n}$ (i.e., $\mathbf{x}^* = \mathbf{x}_b + \delta\mathbf{n}$), for example, via a bilinear interpolation as

$$C(\mathbf{x}^*) \approx \frac{\sum_{i=1}^{4} A_i C(\mathbf{x}_i)}{\delta x^2}, \tag{23}$$

with $\mathbf{x}_i$ the four nearest lattice nodes and A_i the corresponding fractional areas in the lattice cell [Fig. 1(b)]. The boundary value can then be related to the boundary gradient by

$$\left(\frac{\partial C}{\partial n} \right)_b \approx \frac{C(\mathbf{x}^*) - C(\mathbf{x}_b)}{\delta}. \tag{24}$$

For the Neumann boundary condition with specified concentration gradient, the boundary concentration value is directly available from Eq. (24),

$$C(\mathbf{x}_b) \approx C(\mathbf{x}^*) - \delta \left(\frac{\partial C}{\partial n} \right)_b, \tag{25}$$

and this estimated boundary value then can be utilized in the evaluation of midpoint concentration value for the PDF modification. As for the general mixed Robin boundary condition

$$\left(a_1 \frac{\partial C}{\partial n} + a_2 C \right)_b = a_3, \tag{26}$$

where a_1, a_2, and a_3 are prescribed constants, the expression of the boundary value can be solved as

$$C(\mathbf{x}_b) \approx \frac{a_1 C(\mathbf{x}^*) - a_3\delta}{a_1 - a_2\delta}. \tag{27}$$

In simulations, the value of δ should be appropriately selected: An unnecessarily large δ can induce larger finite-difference errors; while a too-small δ value may result in an interpolation point $\mathbf{x}^*$ too close to the solid boundary even with some of its neighboring lattice nodes $\mathbf{x}_i$ in the solid domain [see Fig. 1(b) inset], and this will damage the correct interpolation of $C(\mathbf{x}^*)$ in Eq. (23). In this study, we use $\delta = 1.5\delta x$ in all simulations with boundary concentration gradient involved.

III. DEMONSTRATION SIMULATIONS AND DISCUSSION

To examine the capacity and performance of the above described boundary treatments, we have carefully designed several test systems, including the convection-diffusion systems between two horizontal plates, the pure diffusion situations between inclined plates and coaxial circular surfaces, and the convection-diffusion problem of a cylinder moving in a straight channel. The results are presented in nondimensional LBM units.

A. Spatial resolution: Convection-diffusion between horizontal plates with Dirichlet conditions

Following the numerical tests in Ref. [16], we first consider the convection-diffusion systems between two horizontal plates, both aligned along the lattice grid lines. The distance between these two plates is H. The constant concentration values on the upper and lower plates are, respectively, C_u and C_l. Two flow situations are considered, the classical Poiseuille flow between stationary walls driven by a body force and the Couette flow with transboundary injections,

$$u(y = 0) = 0, \quad u(y = H) = U_0, \tag{28}$$

$$v(y = 0) = v(y = H) = V_0, \tag{29}$$

where U_0 in the shear velocity on the top plate and V_0 is the injection velocity. The respective flow and concentration solutions for these systems are [16]

$$u = 4U_0\left[\frac{y}{H} - \left(\frac{y}{H}\right)^2\right], \quad v = 0, \tag{30}$$

$$C = C_l + (C_u - C_l)\frac{y}{H}, \tag{31}$$

for the Poiseuille flow (U_0 here is the maximum velocity at the channel centerline, and it is related to the applied body force G along the channel direction via $U_0 = GH^2/8\nu$) and

$$u = U_0\frac{e^{Re\frac{y}{H}} - 1}{e^{Re} - 1}, \quad v = V_0, \tag{32}$$

$$C = C_l + (C_u - C_l)\frac{e^{Pe\frac{y}{H}} - 1}{e^{Pe} - 1}, \tag{33}$$

for the injection Couette flow. The Reynolds and Peclet numbers are defined as $\mathrm{Re} = U_0H/\nu$ and $\mathrm{Pe} = V_0H/D$, respectively. The concentration distribution actually is not affected by the Poiseuille flow in Eq. (31), since the flow direction (horizontal) is exactly perpendicular to the concentration gradient direction (vertical), and thus the convection term $\nabla \cdot (\rho\mathbf{u})$ is 0 in the convection-diffusion equation [Eq. (13)]. Nevertheless, this system can still serve as a simple case to validate the program and examine the model performance.

Our simulations are conducted over a rectangular D2Q9 domain with periodic boundary conditions applied at the left and right domain edges. The channel width is $H = 6$ for the Poiseuille flow simulations, and it is $H = 20$ for the injection Couette flow calculations. The injection flow velocity V_0 is set to be the same as the top plate shear velocity U_0 in the injection Couette flows. The relaxation parameters are set as $\tau_f = \tau_g = 1$ for all simulations in this work, and their effects have been examined in Ref. [16]. The body force for the Poiseuille flow and boundary velocities for the injection Couette flow are calculated to have $\mathrm{Re} = \mathrm{Pe} = 10$, as employed in Ref. [16]. In addition, the channel centerline location relative to the underlying lattice grid lines is adjusted to test the result sensitivity to boundary locations. The vertical distance between the channel centreline and the closest grid line below is defined as the centerline offset α. Three different offset values are considered with $\alpha = 0.2$, 0.5, and 0.7 for both the Poiseuille and injection Couette flows. At $\alpha = 0.5$, both the upper and lower plates locate exactly along the midplane between two lattice grid lines. For comparison, both the original bounce-back scheme in Ref. [16] and the midpoint bounce-back method presented in this paper are utilized.

The calculated velocity and concentration distributions across the channel are displayed in Fig. 2. The transverse velocity v is not shown there since it is uniform and 0 in the Poiseuille flows and V_0 in the injection flows. Overall, both boundary methods have successfully reproduced the general variation trends (black lines). However, when we vary the relative position of the channel centerline from $\alpha = 0.2$, 0.5, to 0.7, the original bounce-back scheme does not respond to such boundary location change: The same value is produced at a particular lattice node even when the surface has been shifted upward or downward. On the other side, with our improved midpoint method, the simulated profiles correctly follow the theoretical prediction based on the current centerline position. The still relatively evident deviations from the theoretical curves are due to the narrow channel width $H = 6$ for the Poiseuille flows and the large nonlinear slopes in velocity

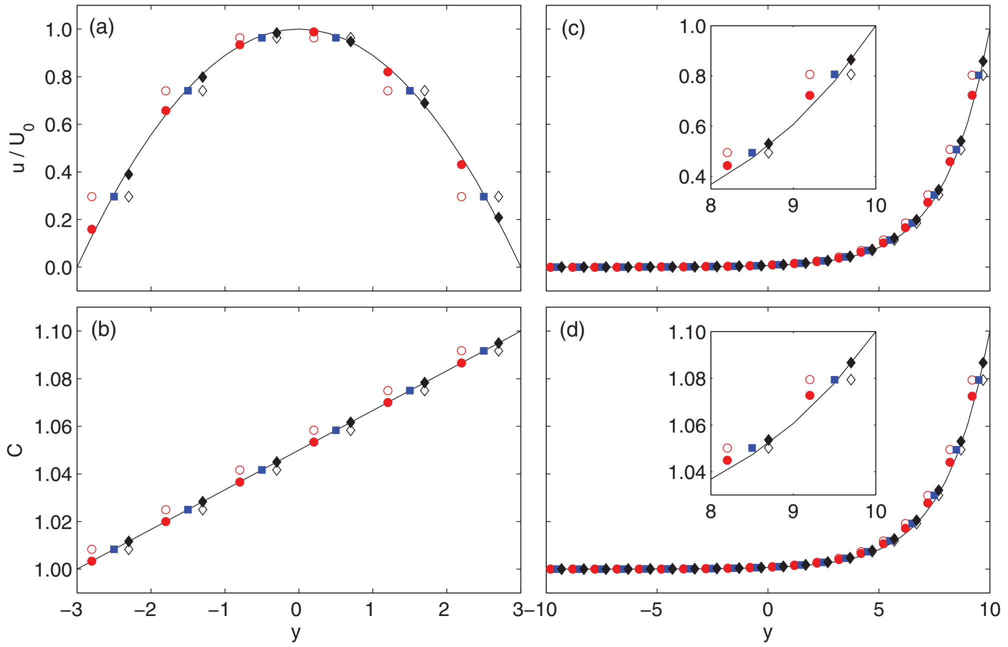

FIG. 2. (Color online) The velocity (upper panels) and concentration (lower panels) distributions for the Poiseuille (left panels) and injection Couette (right panels) flows. The black lines are theoretical solutions, and the symbols are results from LBM simulations: open symbols for the original bounce-back scheme [16], and filled symbols for our improved midpoint method. The symbol shapes and colors are used to indicate the different centerline offsets: red circles for $\alpha = 0.2$, blue squares for $\alpha = 0.5$, and black diamonds for $\alpha = 0.7$. The range of $8 < y < 10$ in the right panels have been enlarged in the insets to show more details.

and concentration near the top plate in the injection Couette flows. For offset $\alpha = 0.5$, the results from the original and the midpoint methods are identical, and the open squares are completely overlapped by the filled squares in Fig. 2. This identity is expected since with $\alpha = 0.5$ we have $\mathbf{x}_b = \mathbf{x}_m$ for all boundary lattice links.

B. Boundary gradient approximation: Diffusion with sinusoidal concentration gradient

To test the improvement of our finite-difference approximation in the normal boundary direction in Eq. (24), we consider the pure diffusion problem between the two horizontal plates with $H = 20$ and $\alpha = 0.5$. A constant-value Dirichlet boundary is applied on the lower plate, and a sinusoidal concentration gradient is imposed on the upper plate,

$$C(y = 0) = C_0, \tag{34}$$

$$\left(\frac{\partial C}{\partial y}\right)_{y=H} = C' \sin \beta x, \tag{35}$$

where C' is the gradient variation amplitude and β is the wave number. The flow velocity is 0 everywhere by setting the two plates stationary and applying no body force. This makes the system analytically solvable, and the exact solution of this problem is

$$C(x,y) = C_0 + \frac{C'}{\beta}\frac{\sinh \beta y}{\cosh \beta H}. \tag{36}$$

In our calculation, we use $C_0 = 1$, $C' = 0.01$, and $\beta = \pi/24$. The rectangular simulation domain has a width of 48 lattice units to include one entire period of the gradient variation, and thus the periodic boundary condition can be applied in the horizontal direction. Moreover, both the original [Eq. (22)] and our improved [Eq. (24)] finite-difference schemes have been considered. The distance δ for our normal-direction finite-difference approximation is selected as 1.5 lattice units to avoid any possible involvement of a solid node in the bilinear interpolation for the concentration value at $\mathbf{x}^*$.

Figure 3 displays the LBM results (color symbols) in comparison with the analytical solutions (black lines) at several representative horizontal and vertical locations. Unsurprisingly, our finite-difference scheme along the boundary normal direction exhibits a much better performance than that along the boundary lattice link, particularly in the region near the top plate on which the nonuniform concentration gradient is imposed. For example, the theoretical value at location ($x = 12$, $y = 19.5$) is 1.0707, that from our normal-direction scheme is 1.0687, and that from the lattice-direction finite-difference scheme is 1.0555 (7.6 times larger in relative errors). Since the centerline offset α is 0.5 in these simulations,

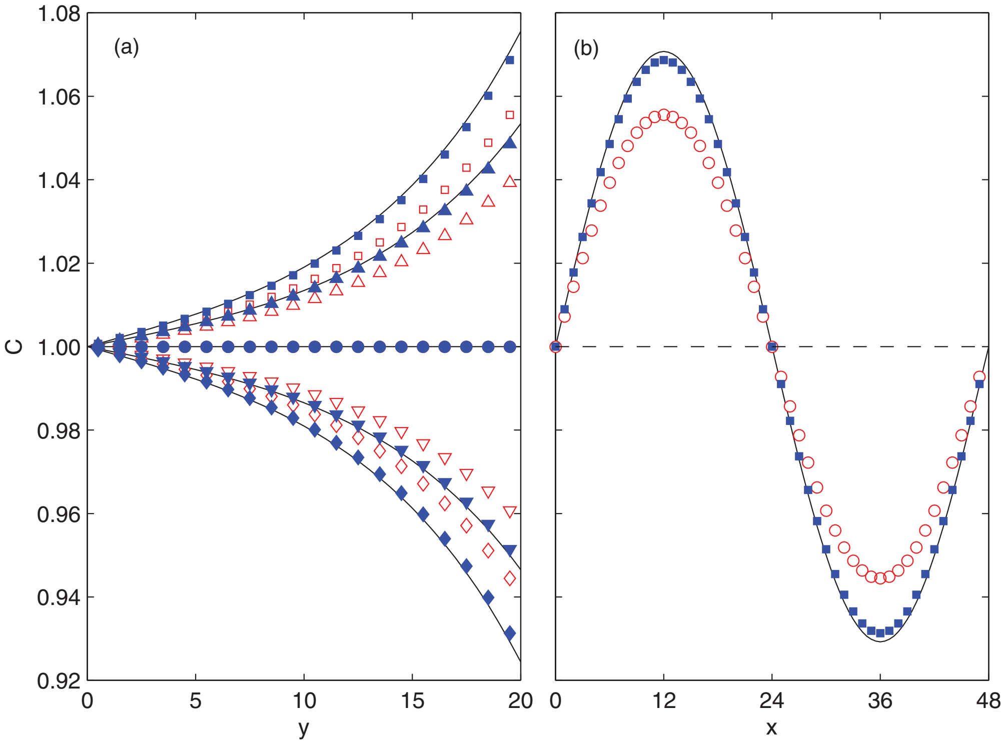

FIG. 3. (Color online) The concentration distributions between a constant-value and a sinusoidal gradient boundaries. Calculated results at five x locations ($x = 12$, 18, 24, 30, and 36, from top to bottom and in the order of squares, upward triangles, circles, downward triangles, and diamonds, left panel) and one y location ($y = 19.5$, the first layer of lattice nodes from the top plate, right panel) are selected to display the concentration field over the two-dimensional domain. The theoretical solutions are plotted as black curves, and results from LBM simulations are displayed as symbols (red open symbols for results from the finite-difference scheme along the boundary lattice link [16], and the blue filled symbols for those from our finite-difference scheme in the boundary normal direction).

the difference observed here must be from the different finite-difference schemes.

It is interesting to note that at $x = 24$, both finite-difference schemes yield identical results [circles in Fig. 3(a)]. We have expected to see the most significant difference between the two schemes at this location, since this is the place where the steepest gradient variation occurs. Through a careful examination, we find the reason of these identical results is due to the antisymmetric gradient variation at this particular location. For the upper boundary lattice node at $x = 24$ and $y = 19.5$ (the upper wall locates at $y = 20$), there are three boundary PDFs, g_2^*, g_5^*, and g_6^*, to be bounced back from the top surface, and they return to this same node as g_4, g_7, and g_8, respectively. For the $g_2^* \rightarrow g_4$ bouncing-back process, the lattice direction is just the boundary normal direction, and therefore no difference exists for the two different finite-difference schemes. For the $g_5^* \rightarrow g_7$ and $g_6^* \rightarrow g_6$ processes, the lattice directions deviated from the boundary normal direction; however, the inaccuracy by neglecting the gradient variation (antisymmetric here) from these two sides has the same magnitude and opposite signs. When these two PDFs g_7 and g_8 arrive the boundary node, the concentration value there is calculated by Eq. (22), and their individual inaccuracy will be simply canceled out. Similar analysis can be applied to the upper boundary nodes at $x = 12$ and 36, where the symmetric gradient variations take place, and therefore the largest errors from the lattice-direction finite-difference scheme are observed.

C. General conditions on curved boundaries: Diffusion between circular surfaces

As reviewed in the Introduction, previous boundary methods for convection-diffusion LBM models are typically limited to flat surfaces or certain boundary conditions or suffer low geometry resolutions. Here we simulate the pure diffusion situation between two circular surfaces with radii R_{in} and R_{out}, respectively. Again the fluid flow is excluded for simplicity. The inner surface is assigned a constant concentration value of C_{in} (Dirichlet condition), while three different situations are considered for the outer surface:

(1) Case A: the Dirichlet condition with constant concentration value C_{out};

(2) Case B: the Neumann condition with constant concentration gradient C'_{out};

(3) Case C: the mixed Robin condition as given in Eq. (26).

The respective solutions for these axisymmetric systems are

$$C(r) = C_{\text{in}} + (C_{\text{out}} - C_{\text{in}})\frac{\ln(r/R_{\text{in}})}{\ln(R_{\text{out}}/R_{\text{in}})} \quad \text{for Case A,} \tag{37}$$

$$C(r) = C_{\text{in}} - R_{\text{out}}C'_{\text{out}}\ln(r/R_{\text{in}}) \quad \text{for Case B,} \tag{38}$$

and

$$C(r) = C_{\text{in}} + \frac{a_2 C_{\text{in}} - a_3}{a_2 \ln(R_{\text{in}}/R_{\text{out}}) - a_1/R_{\text{out}}} \times \ln(r/R_{\text{out}}) \quad \text{for Case C.} \tag{39}$$

In our simulation, the boundary radii are $R_{\text{in}} = 40$ and $R_{\text{out}} = 80$, and the inner boundary concentration value is $C_{\text{in}} = 1.5$. For Case A the boundary value on the outer surface is set as $C_{\text{out}} = 0.5$, and for Case B the boundary gradient on the out surface is $C'_{\text{out}} = 0.018$. The parameter in the Robin condition on the outer surface for Case C are $a_1 = 18$, $a_2 = 0.5$, and $a_3 = -0.1$. Such a particular set of parameters generate similar concentration distributions for these three different boundary combinations. The simulated concentration profiles from the boundary treatments presented in this paper are displayed in Fig. 4. A good agreement between the simulated and theoretical distributions, for all cases considered, can be observed there. Slightly larger errors are noticed near the outer surface for Case B due to the strong dependence on boundary gradient via a first-order finite-difference approximation. This set of simulations demonstrates that our boundary method is capable of dealing with various boundary conditions on curved surfaces.

D. Numerical accuracy: Flow and diffusion between inclined plates and circular surfaces

As for typical LBM boundary models, we now study the numerical accuracy of our boundary treatment. Two systems

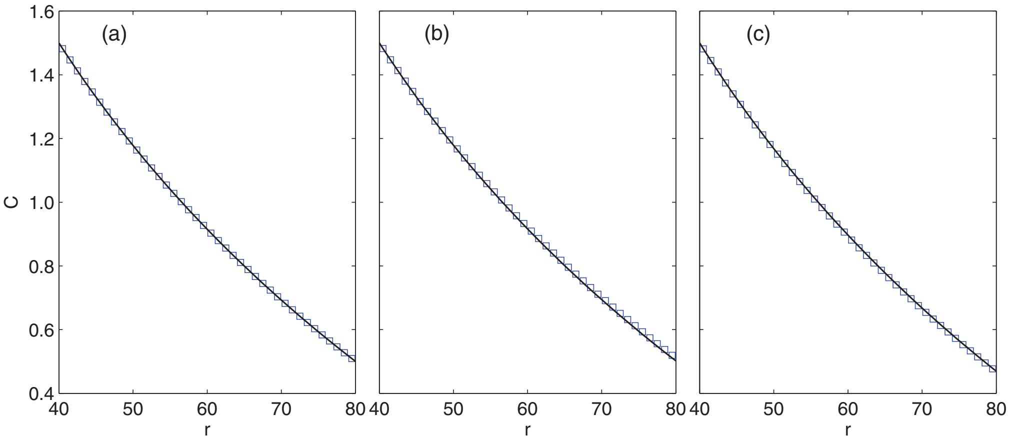

FIG. 4. (Color online) The calculated concentration distributions between two circular surfaces with the Dirichlet condition on the inner surface and difference conditions on the outer surface: (a) the Dirichlet condition, (b) the Neumann condition, and (c) the mixed Robin condition. The symbols are from our LBM simulations and the black lines are from the analytical solutions.

are considered here: the simple Couette shear flows between two inclined parallel plates and the circular Couette flows between two coaxial circular surfaces. For the inclined plate system, the inclination angle is $\theta = \tan^{-1} 5/8$ [16], and a periodic domain is employed [26]. The boundary concentrations are $C_l = 0$ on the lower, stationary plate, and $C_u = 1.0$ on the upper, moving plate with a shear velocity $U_0 = 10^{-4}$. For the circular Couette flows, we maintain the outer radius as twice of the inner radius, i.e., $R_{\text{out}} = 2R_{\text{in}}$. The boundary concentration values are $C_{\text{in}} = 1.5$ on the inner, stationary surface and $C_{\text{out}} = 1.0$ on the outer, rotating surfaces. The rotation speed Ω of the outer surface is adjusted with the system size such that $\text{Re} = \Omega R_{\text{in}}^2/\nu = 5$ [26,29]. These boundary parameters are held constant, while different plate separations H for the inclined-plate system and annular gaps $R_{\text{out}} - R_{\text{in}} = R_{\text{in}}$ (since $R_{\text{out}} = 2R_{\text{in}}$ in this work) for the circular system are utilized to examine the numerical accuracy.

The global relative error is defined as

$$E_A = \left[\frac{\sum (A_{\text{LBM}} - A_{\text{th}})^2}{\sum A_{\text{th}}^2} \right]^{1/2}, \tag{40}$$

where both summations are performed over all lattice nodes in the simulated domain. The property A could be the concentration C or the velocity magnitude $U = (u^2 + v^2)^{1/2}$, and the subscripts LBM and th indicate the calculated and theoretical values.

The calculated global errors are plotted in Fig. 5, and linear fittings are conducted in these logarithmic graphs, with the line slopes also displayed in Fig. 5. The fitting slope is usually considered as the accuracy order of a numerical model. Here, similarly to those observed in typical LBM boundary methods, slopes for both flow and concentration from the circular system are about 2, indicating a second-order overall accuracy for the circular system. However, for the system with inclined plates, the accuracy orders are much lower: 1.383 for flow and 1.009 for concentration. The slope of 1.009 for concentration is similar to that from the original boundary method by Zhang *et al.* (1.03 there) [16,30]; however, the error magnitude is orders smaller ($\sim 10^{-9}$ from our improved method vs $\sim 10^{-3}$ from the original method), indicating more accurate results have been produced from our improved method. It is interesting to note that different slopes (accuracy orders) can be observed from different systems with the same numerical method. A qualitative understanding can be established by considering the fact that the global errors actually include the inaccuracy from the LBM algorithm as well as the assumptions in the boundary method. For the inclined-plates system, since the LBM algorithm mathematically can be considered a second-order finite-difference solver of differential equations [16,31,32], the LBM solutions for the linear flow and concentration fields between two parallel plates should be perfectly exact (except the inevitable computer round-off errors). However, any boundary inaccuracy will affect the entire linear distribution and yield an incorrect slope between the two plates, as shown in Ref. [31]. Thus, the relative error between the theoretical line (connecting the real boundary values with the correct slope) and the simulated line (connecting the shifted boundary values due to boundary inaccuracy with a slightly different slope) is proportional to the original error at boundary, which is directly related to the boundary resolution. This has been confirmed by our further experiments. For the $H = 16$ separation, when we change the tolerance value in the calculation of the boundary node location Δ from its current value 10^{-7} to 10^{-6} (i.e., to set the boundary location less precisely), the global error in concentration

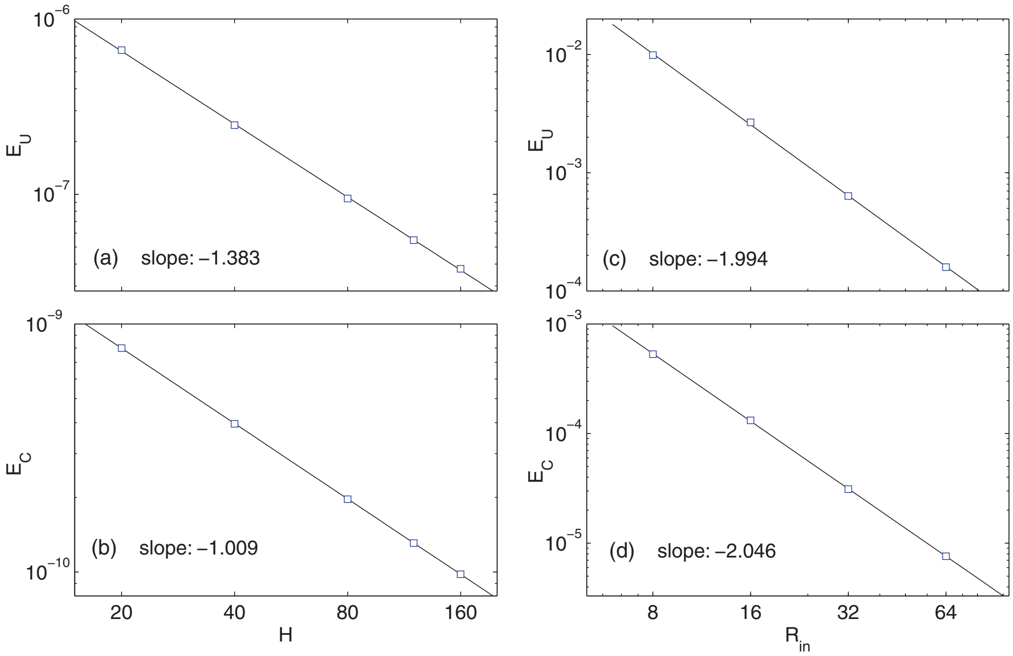

FIG. 5. (Color online) The global relative errors for flow (upper panels) and concentration (lower panels) between inclined plates (left panels) and circular surfaces (right panels) with different separations. The black lines are from linear fittings of the relative errors in the logarithmic scale. The slope of the fitting line is also shown in the lower-left corner in each panel.

increases from 1.002×10^{-9} to 1.259×10^{-8}. On the other hand, in the circular system, the flow and concentration distributions are neither exactly linear (first-order) nor parabolic (second-order). It appears that the error from the second-order LBM algorithm in describing the nonlinear and nonparabolic flow or concentration fields suppresses the relatively small boundary inaccuracy and dominates the overall global error behavior. As a result, the error-resolution relationship yields an approximately second-order accuracy. Accuracy orders between 1 and 2 have also been reported from other LBM boundary methods [22,27,33], indicating the combined effects from boundary and LBM algorithm.

In addition to the different slopes, the absolute error magnitudes from the inclined-plate system are orders lower than those from the circular system. This might be mainly due to the particular interpolation or extrapolation schemes employed to estimate the midpoint velocity and concentration values. For the inclined-plate system, the velocity and concentration distributions are linear, and the linear interpolation or extrapolation schemes can yield excellent predictions of the midpoint values. This will not be the case for the circular system, where neither the flow nor the concentration distributions is linear. Moreover, compared to the flat walls in the inclined-plate system, the curved surfaces in the circular system could also introduce larger numerical errors. At last, we also notice that the relatively larger error magnitudes for flow fields than those for concentration fields, in both the inclined-plate and circular systems, are mainly due to the particular formulation for the relative error in Eq. (40). The velocity magnitude varies from 0 to $U_0 = 10^{-4}$ in the inclined-plate system or from 0 to 0.02 in the circular system, while the concentration values cover a range of [0, 1] in the inclined-plate system and a range of [1,1.5] in the circular system. These relatively larger A_{th} values in the denominator for concentration calculations cause the corresponding relative errors to be much smaller than those of the flow calculations.

E. Galilean invariance: Accuracy with different reference velocities

So far our simulations are performed in relatively simple systems, and all the boundary surfaces are not moving. At last, we consider a more general system with a circular cylinder of radius R moving at a constant velocity U_0 in a horizontal channel with width H along the centerline. Constant concentrations are applied on the cylinder surface (C_c) and the channel surfaces (C_s). In addition, we also consider the system with the cylinder stationary but with the channel walls moving in the opposite direction $-U_0$. According to the Galilean invariance principle, these two approaches should yield the same concentration field, and the flow velocities should have a difference of U_0 between them. The parameters we use here are $H = 160$, $R = 40$, $U_0 = 5 \times 10^{-3}$, $C_c = 1.1$, and $C_s = 1.0$. For the moving-cylinder system, the cylinder surface is moving in space, and some fluid lattice nodes in front of the cylinder can be covered, as well as some solid lattice nodes at the back of the cylinder can be released. Here we adopt the treatment described in Ref. [34], for both the flow and concentration PDFs at these nodes. The PDFs for the covered nodes will be simply ignored, while the PDFs for a newly released node $\mathbf{x}_n$ are assigned via a weighted extrapolation from existing neighboring fluid nodes,

$$f_i(\mathbf{x}_n) = \frac{\sum_j \omega_j[2f_i(\mathbf{x}_n + \mathbf{e}_j\delta t) - f_i(\mathbf{x}_n + 2\mathbf{e}_j\delta t)]}{\sum_j \omega_j}, \tag{41}$$

$$g_i(\mathbf{x}_n) = \frac{\sum \omega_j[2g_i(\mathbf{x}_n + \mathbf{e}_j\delta t) - g_i(\mathbf{x}_n + 2\mathbf{e}_j\delta t)]}{\sum_j \omega_j}, \tag{42}$$

where the lattice index i runs from 0 to 8, and the summations in the right-hand side are taken over all lattice directions j at $\mathbf{x}_n$ pointing to existing fluid nodes.

The results from these two simulations are displayed in Fig. 6. The pressure and velocity fields between such two systems have been examined in our previous publication [34], and it is confirmed again in this study that the pressure distributions are similar, while the velocity fields have a U_0 difference. The concentration fields also appear identical from the contour lines and the centerline profiles have excellent agreement with each other [Figs. 6(a)–6(c)]. The asymmetric distribution about the cylinder reflects the convection effect on the diffusion process. For a more quantitative comparison, similar to Eq. (40), here we define a global difference in concentration between such two systems as

$$E_C = \left[\frac{\sum(C_{\text{mc}} - C_{\text{mw}})^2}{\sum C_{\text{mw}}^2}\right]^{1/2}, \tag{43}$$

where C_{mc} and C_{mw} are, respectively, the simulated concentration values from the moving-cylinder and moving-wall systems. The moving-wall system has been selected as the comparison reference in the denominator, since there the curved cylinder surface is stationary and only the flat wall surfaces are moving tangentially, and the simulated results are supposed to be closer to the real values.

The calculated differences by use of Eq. (43) with different relative velocity U_0 are displayed in Fig. 6(d). At the extreme situation with $U_0 = 0$, the two systems become identical, and a zero difference is expected. When a relative motion is introduced, a linear relationship between the difference E_C and relative velocity U_0 (fitting slope = 0.1852) can be seen there. The increase of E_C with U_0 is not surprising. In general, there are three sources that are responsible for the numerical error in our LBM simulations: the second-order LBM algorithm which may not be able to describe the flow and concentration distributions exactly; the linear interpolations or extrapolations in evaluating the midpoint values $\mathbf{u}_m$ and C_m; and the assumptions employed in the bounce-back boundary scheme (see the appendix). As a matter of fact, a large relative velocity U_0 produces larger velocity and concentration gradients around the cylinder surface; and these larger gradients will affect all the error sources negatively. However, detailed analysis of the $E_2 \sim U_0$ relationship for this cylinder-channel system could be difficult, since the imposed velocity U_0 affects the concentration field both directly (larger U_0 → stronger convection effect → larger concentration gradient in front of cylinder → less accurate LBM concentration results) and indirectly [larger U_0 → larger velocity gradient near cylinder → less accurate LBM flow results → less accurate LBM concentration results via Eq. (9)].

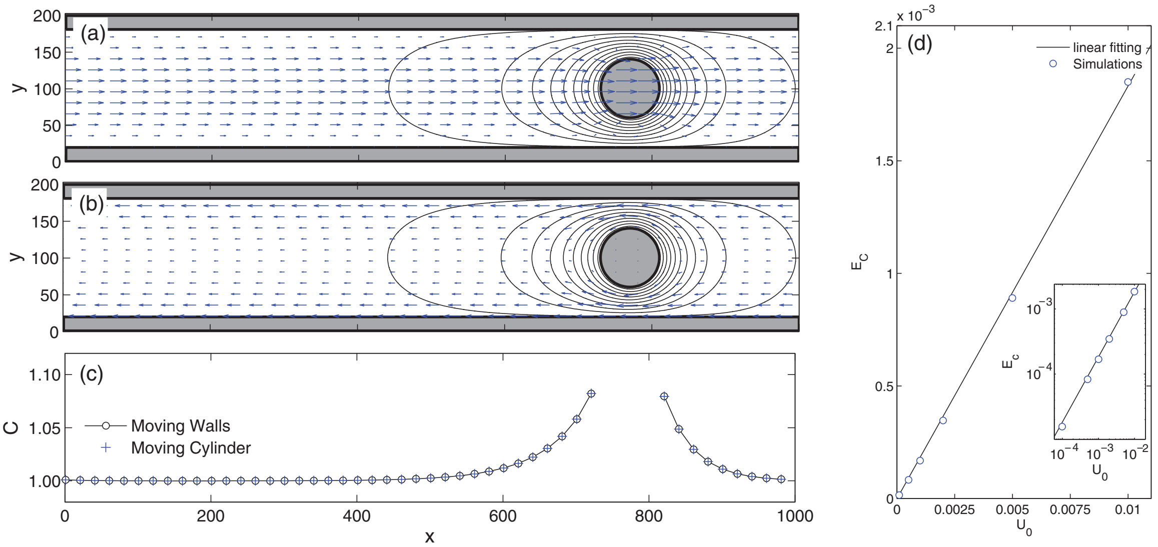

FIG. 6. (Color online) The flow (arrows) and concentration (contour lines) distributions for the moving-cylinder (a) and moving-wall (b) systems with the wall-cylinder relative velocity $U_0 = 5 \times 10^{-3}$. The concentration variations along the channel centerline from these two systems in (a) and (b) are also plotted in (c) for comparison. The global differences in concentration between the moving-cylinder and moving-wall systems with different relative velocity U_0 are plotted in (d) (the inset shows the same data but in logarithmic scales).

For simplicity, below we consider the Case A in Sec. III C, however, with both inner and outer surfaces moving at a same velocity U_0. The flow velocity should be constant U_0 everywhere, and the concentration field should not be affected by this uniform velocity and can still be described by Eq. (37). Here we use $R_{\text{in}} = 40$, $R_{\text{out}} = 80$, and $\tau_f = \tau_g = 1$. With different boundary velocities U_0, two boundary concentration conditions are adopted to examine the effect of concentration gradient on the global error E_2: (1) $C_{\text{in}} = C_{\text{out}} = 1.0$ for a uniform concentration distribution and (2) $C_{\text{in}} = 1.5$ and $C_{\text{out}} = 0.5$ for a nonuniform concentration field. First, we look at the uniform concentration case with $C_{\text{in}} = C_{\text{out}} = 1.0$. For this situation, all the error sources have been eliminated to our best: The LBM algorithm can describe the uniform flow and concentration fields exactly; the midpoint values can be evaluated with linear approximation along boundary lattice links accurately; and all approximations with the bounce-back scheme (see the appendix) are perfectly satisfied. Our calculations show that the surface velocity U_0 has not evident influence on the simulation accuracy, and the relative error E_2 varies in a range of $10^{-17} \sim 10^{-14}$ (black squares in Fig. 7). We believe such tiny differences between the simulated and theoretical concentration values are from the inevitable numerical errors associated with the finite precision in computers. This is confirmed by looking at the simulated fluid velocity, and we find that the difference between the LBM velocity and the imposed boundary velocity U_0 is typically of $10^{-15} \sim 10^{-18}$. On the other hand, for the nonuniform system with $C_{\text{in}} = 1.5$ and $C_{\text{out}} = 0.5$, several-order-larger errors are observed (blue circles in Fig. 7). An error of $E_2 = 5.06 \times 10^{-5}$ exists even when the surfaces are not moving at all ($U_0 = 0$). Clearly, this error is directly related to the concentration variation in the space. When the surface motion is introduced, again we see that the global error increases with U_0 approximately linearly, except in the low-velocity region $U_0 < 10^{-3}$. The constant and uniform boundary and fluid velocity U_0, in principle, should has no effects on accuracies of the LBM algorithm and midpoint value evaluations (i.e., U_0 does not change the spatial concentration distribution). However, the flow velocity near the boundary is directly involved in Eqs. (A7)–(A9), and a larger velocity U_0 will make these approximations less accurate, especially the difference in the linear velocity terms in Eq. (A7).

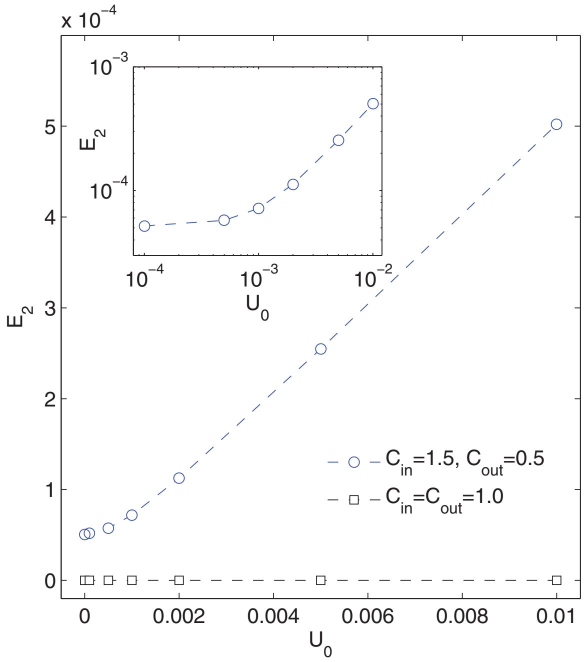

FIG. 7. (Color online) The global relative errors for concentration between two coaxial cicular surfaces moving at reference velocity U_0 with same concentration field ($C_{\text{in}} = C_{\text{out}} = 1.0$, black squares) or different concentration field ($C_{\text{in}} = 1.5$ and $C_{\text{out}} = 0.5$, blue circles). The inset shows the same data of the latter case but in logarithmic scales.

Therefore, the nonuniform concentration distribution is mainly responsible for the increase in E_2 with the boundary velocity U_0 in Fig. 7.

The above simulations and analysis indicate that, as the overall accuracy discussed in Sec. III D, the performance of a LBM model in preserving the fundamental Galilean invariance is a complicated phenomenon. Several factors, including the LBM algorithm accuracy, the boundary method, and the particular flow and concentration distributions being simulated, are all playing roles, and their effects are highly interconnected. For the LBM algorithm and boundary method employed in this study, the Galilean invariance performance is generally satisfactory. The global relative error could become significant for a larger relative or reference velocity U_0; however, such a large U_0 is not favorable for incompressible flows [2] and should be avoided in practical applications.

In addition, our up-to-date literature search indicates that, in previous LBM simulations of convection-diffusion or heat transfer processes, the system boundaries are typically stationary, or with some boundary velocities (e.g., shearing velocity, injection velocity, or rotating velocity) imposed at fixed spatial boundary locations, and no physical boundary movement and displacement have been considered. Our simulation of the moving cylinder demonstrates that our boundary treatments (including those for the boundary velocity, the boundary concentration, and the node status change) can be a good choice for simulating systems with physical boundary movement, such as the convection-diffusion and heat transfer processes in particulate flows.

IV. SUMMARY

We have extended the recent midpoint boundary method for LBM flow simulations to the convection-diffusion processes. We have also proposed to evaluate the concentration boundary value from the concentration gradient via a finite-difference scheme in the normal direction to consider nonuniform boundary conditions. Compared with other existing LBM boundary methods, our boundary treatments described in this paper can work with arbitrary curved geometry, boundary velocity and motion, and various surface conditions (the Neumann, Dirichlet, or mixed Robin conditions). To examine the capacity and performance of the present boundary methods in describing flow and concentration fields, several carefully designed simulations have been conducted. The simulation results have been compared with analytical solutions and numerical results from a representative boundary method. These simulations and comparisons show that our methods have good performances in dealing with complex boundary situations, including the precise boundary location, the accurate solution near the boundary with a nonuniform gradient distribution, the capability to work for different boundary conditions on curved surfaces, and an improved numerical accuracy. In addition, we have examined the performance of our method in preserving the Galilean invariance, and the effect of reference velocity on numerical accuracy has been investigated. The moving-cylinder simulation also demonstrates the potential usefulness of our methods in simulating systems with physical boundary movement. The relatively simple algorithm is particularly advantageous for transport processes in particulate and porous systems.

Last, we point out that, although the model description as well as the simulations presented in this paper are all two-dimensional with the D2Q9 lattice structure and the single-relaxation-time LBGK algorithm, extending these methods to three-dimensional situations, other lattice structures, and/or MRT LBM models should be straightforward and present no technical difficulty. The present methods are also readily applicable to other processes and phenomena that can be described by convection-diffusion-type differential equations, for example, heat transfer and convection-diffusion-reaction processes. More complicated boundary conditions such as chemical reactions at boundary can also be implemented [6].

ACKNOWLEDGMENTS

This work was supported by the Natural Science and Engineering Research Council of Canada (NSERC), the Laurentian University Research Fund (LURF), the Research Fund for the Doctoral Program of Higher Education, and the Natural Science Foundation of Jiangsu Province, China (BK20131348). J.Z. is grateful to T. Zhang, B. Shi, and Z. Guo at Huazhong University of Science and Technology (Wuhan, China) for clarifying the global error calculation in Ref. [16]. Q.C. acknowledges the visiting studentship from Nanjing University of Science and Technology for his study in Laurentian University. This work was made possible by the facilities of Westgrid (www.westgrid.ca) and SHARCNet (the Shared Hierarchical Academic Research Computing Network, www.sharcnet.ca).

APPENDIX

In this appendix, we discuss how the boundary PDF $g_{\bar{i}}$ in Eq. (18) can be derived and the assumptions and approximations involved. Here we follow Zhang *et al.* [16] and only consider the situation that the boundary locates at the midposition between two vertical lattice lines [Fig. 8(a)]. After the collision step at the boundary fluid node $\mathbf{x}_f$, the density distribution $g_i^*(\mathbf{x}_f)$ leaves toward the surface wall, and it is then bounced back at the wall $\mathbf{x}_b$ ($\mathbf{x}_w$ in Ref. [16]) with a modified magnitude as $g_{\bar{i}}(\mathbf{x}_f)$, which will be utilized in the next collision operation at $\mathbf{x}_f$. In Fig. 8, we have the boundary lattice directions $i = 1$ and $\bar{i} = 3$. On the other hand, we can consider the solid node at $\mathbf{x}_s$ as a ghost fluid node. The postcollision distribution $g_{\bar{i}}^*(\mathbf{x}_s)$ will move to $\mathbf{x}_f$ during the regular propagation process [Fig. 8(b)],

$$g_{\bar{i}}(\mathbf{x}_f) = g_{\bar{i}}^*(\mathbf{x}_s). \tag{A1}$$

To obtain an estimate of $g_{\bar{i}}(\mathbf{x}_f)$ or $g_{\bar{i}}^*(\mathbf{x}_s)$, several approximations are necessary. First, we assume that all postcollision PDFs are close to their individual equilibrium values [2], i.e.,

$$g_i^* \approx g_i^{\mathrm{eq}}, \tag{A2}$$

and applying Eq. (9) to nodes $\mathbf{x}_f$ and $\mathbf{x}_s$, respectively, yields

$$g_i^*(\mathbf{x}_f) \approx \omega_i C_f \left[1 + \frac{\mathbf{e}_i \cdot \mathbf{u}_f}{c_s^2} + \frac{(\mathbf{e}_i \cdot \mathbf{u}_f)^2}{2c_s^4} - \frac{\mathbf{u}_f \cdot \mathbf{u}_f}{2c_s^2} \right], \tag{A3}$$

$$g_{\bar{i}}^*(\mathbf{x}_s) \approx \omega_{\bar{i}} C_s \left[1 + \frac{\mathbf{e}_{\bar{i}} \cdot \mathbf{u}_s}{c_s^2} + \frac{(\mathbf{e}_{\bar{i}} \cdot \mathbf{u}_s)^2}{2c_s^4} - \frac{\mathbf{u}_s \cdot \mathbf{u}_s}{2c_s^2} \right]. \tag{A4}$$

Here subscripts f and s have been adopted to indicate node locations for flow velocity $\mathbf{u}$ and concentration C. Adding

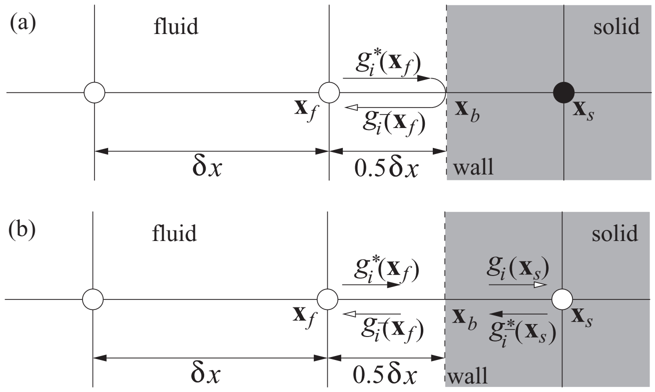

FIG. 8. Schematics showing (a) the bouncing-back process of density distribution at a solid surface and (b) the propagation process between the ghost (solid) and fluid nodes across the boundary. Filled arrows are used to indicate postcollision density distributions leaving lattice nodes, and open arrows are for the precollision distributions arriving at nodes after one time step.

Eqs. (A3) and (A4), and noticing $\mathbf{e}_i = -\mathbf{e}_{\bar{i}}$ and $\omega_i = \omega_{\bar{i}}$, one can have

$$g_i^*(\mathbf{x}_f) + g_{\bar{i}}^*(\mathbf{x}_s) \approx \omega_i \left[(C_f + C_s) + \frac{\mathbf{e}_i \cdot (C_f \mathbf{u}_f - C_s \mathbf{u}_s)}{c_s^2} + \frac{C_f(\mathbf{e}_i \cdot \mathbf{u}_f)^2 + C_s(\mathbf{e}_i \cdot \mathbf{u}_s)^2}{2c_s^4} - \frac{C_f \mathbf{u}_f \cdot \mathbf{u}_f + C_s \mathbf{u}_s \cdot \mathbf{u}_s}{2c_s^2} \right]. \tag{A5}$$

Since C_s and $\mathbf{u}_s$ at the ghost node $\mathbf{x}_s$ are not available, we assume slow variations in concentration C and velocity $\mathbf{u}$, as well as any product combinations from them, between $\mathbf{x}_f$ and $\mathbf{x}_s$. We then can approximate the right-hand side terms in the above equation with the twice of the corresponding midpoint values for addition and zeros for subtraction as follows:

$$C_f + C_s \approx 2C_b, \tag{A6}$$

$$C_f \mathbf{u}_f - C_s \mathbf{u}_s \approx \mathbf{0}, \tag{A7}$$

$$C_f(\mathbf{e}_i \cdot \mathbf{u}_f)^2 + C_s(\mathbf{e}_i \cdot \mathbf{u}_s)^2 \approx 2C_b(\mathbf{e}_i \cdot \mathbf{u}_b)^2, \tag{A8}$$

$$C_f \mathbf{u}_f \cdot \mathbf{u}_f + C_s \mathbf{u}_s \cdot \mathbf{u}_s \approx 2C_b \mathbf{u}_b \cdot \mathbf{u}_b, \tag{A9}$$

where C_b and $\mathbf{u}_b$ are, respectively, the concentration and velocity at the boundary. Substituting these approximations back to Eq. (A5), we have

$$g_i^*(\mathbf{x}_f) + g_{\bar{i}}^*(\mathbf{x}_s) \approx \omega_i \left[2C_b + \frac{C_b(\mathbf{e}_i \cdot \mathbf{u}_b)^2}{c_s^4} - \frac{C_b \mathbf{u}_b \cdot \mathbf{u}_b}{c_s^2} \right]. \tag{A10}$$

At last, we replace $g_{\bar{i}}^*(\mathbf{x}_s)$ with $g_{\bar{i}}(\mathbf{x}_f)$ according to the propagation operation in LBM Eq. (A1), and move the left-hand side term $g_i^*(\mathbf{x}_f)$ to the right,

$$g_{\bar{i}}(\mathbf{x}_f) \approx -g_i^*(\mathbf{x}_f) + 2\omega_i C_b \left[1 + \frac{(\mathbf{e}_i \cdot \mathbf{u}_b)^2}{2c_s^4} - \frac{\mathbf{u}_b \cdot \mathbf{u}_b}{2c_s^2} \right], \tag{A11}$$

which is identical to Eq. (18).

[1] S. Chen and G. D. Doolen, Annu. Rev. Fluid Mech. **30**, 329 (1998).
[2] S. Succi, *The Lattice Boltzmann Equation* (Oxford University Press, Oxford, 2001).
[3] J. Zhang, Microfluidics Nanofluidics **10**, 1 (2011).
[4] Q. Chen and X. Zhang, Int. J. Mod. Phys. C **23**, 1250074 (2012).
[5] S. P. Dawson, S. Chen, and G. D. Doolen, J. Chem. Phys. **98**, 1514 (1993).
[6] X. He, N. Li, and B. Goldstein, Mol. Simul. **25**, 145 (2000).
[7] I. Rasin, S. Succi, and W. Miller, J. Comput. Phys. **206**, 453 (2005).
[8] R. G. M. van der Sman and M. H. Ernst, J. Comput. Phys. **160**, 766 (2000).
[9] X. Zhang, A. G. Bengough, J. W. Crawford, and I. M. Young, Adv. Water Resour. **25**, 1 (2002).
[10] I. Ginzburg, Adv. Water Resour. **28**, 1171 (2005).
[11] B. Shi, B. Deng, R. Du, and X. Chen, Comput. Math. Appl. **55**, 1568 (2008).
[12] J. Eggels and J. Somers, Int. J. Heat Fluid Flow **16**, 357 (1995).
[13] X. He, S. Chen, and G. Doolen, J. Comput. Phys. **146**, 282 (1998).
[14] Z. Guo, C. Zheng, B. Shi, and T. S. Zhao, Phys. Rev. E **75**, 036704 (2007).
[15] J. Wang, D. Wang, P. Lallemand, and L. Luo, Comput. Math. Appl. **65**, 262 (2013).
[16] T. Zhang, B. Shi, Z. Guo, Z. Chai, and J. Lu, Phys. Rev. E **85**, 016701 (2012).
[17] Q. Kang, D. Zhang, S. Chen, and X. He, Phys. Rev. E **65**, 036318 (2002).
[18] Q. Kang, P. C. Lichtner, and D. Zhang, Water Resour. Res. **43**, W12S14 (2007).
[19] S. D. C. Walsh and M. O. Saar, Phys. Rev. E **82**, 066703 (2010).
[20] H. Yoshida and M. Nagaoka, J. Comput. Phys. **229**, 7774 (2010).
[21] I. Ginzburg, Adv. Water Resour. **28**, 1196 (2005).
[22] L. Li, R. Mei, and J. F. Klausner, J. Comput. Phys. **237**, 366 (2013).
[23] S. Chen, K. H. Luo, and C. Zheng, J. Comput. Phys. **231**, 8278 (2012).
[24] A. J. C. Ladd, J. Fluid Mech. **271**, 285 (1994).
[25] L. Chen, Q. Kang, B. A. Robinson, Y. He, and W. Tao, Phys. Rev. E **87**, 043306 (2013).
[26] X. Yin and J. Zhang, J. Comput. Phys. **231**, 4295 (2012).
[27] O. Oulaid and J. Zhang (unpublished).
[28] R. Mei, L. Luo, and W. Shyy, J. Comput. Phys. **155**, 307 (1999).
[29] Z. Guo, C. Zheng, and B. Shi, Phys. Fluids **14**, 2007 (2002).
[30] T. Zhang, B. Shi, Z. Guo, Z. Chai, and J. Lu, Phys. Rev. E **88**, 029903(E) (2013).
[31] G. Le and J. Zhang, Phys. Rev. E **79**, 026701 (2009).
[32] X. He, Q. Zou, L. Luo, and M. Dembo, J. Stat. Phys. **87**, 115 (1997).
[33] D. Hlushkou, D. Kandhai, and U. Tallarek, Int. J. Numer. Methods Fluids **46**, 507 (2004).
[34] X. Yin, G. Le, and J. Zhang, Phys. Rev. E **86**, 026701 (2012).