Accuracy and stability of a lattice-Boltzmann model with subgrid scale boundary conditions

R. Verberg and A. J. C. Ladd*
Chemical Engineering Department, University of Florida, Gainesville, Florida 32611-6005
(Received 24 May 2001; published 17 December 2001)

A lattice-Boltzmann method has recently been developed to incorporate solid-fluid boundary conditions on length scales less than the grid spacing. By introducing a real numbered parameter, specified at each node and representing the fluid volume associated with that node, we were able to accurately simulate arbitrary geometries without the need to specify surface normals. In this paper a detailed description of the rules is presented and the accuracy and stability of the method is discussed, based on numerical results for flow in systems with planar surfaces and for flow through periodic arrays of disks and spheres.

DOI: 10.1103/PhysRevE.65.016701 PACS number(s): 83.85.Pt, 47.11.+j, 47.15.Gf, 47.55.Mh

I. INTRODUCTION

In the past decade, the lattice-Boltzmann method has become the simulation method of choice for a number of fluid dynamics problems [1], in particular for fluid flow in complex geometries. In this paper a modification of the conventional lattice-Boltzmann method is presented, which incorporates information about the solid surface on scales less than the resolution of the grid. In most lattice-Boltzmann simulations the solid-fluid boundary conditions are modeled by the bounce-back rule, in which particles encountering a solid surface are reflected back in the direction they came from. However, discretization of the solid surface introduces particular problems in simulations with moving boundaries or where the surface morphology is evolving due to erosion or deposition of dissolved solids [2]. In such cases, changes to the solid surfaces cannot be modeled smoothly, but only in discrete units of the grid spacing. To simulate these systems a boundary condition was devised for the lattice-Boltzmann method in which the location of the solid surface can change continuously, on scales less than the grid spacing [3].

These boundary rules, which are called continuous bounce-back (CBB) rules [3], are an extension of the link bounce-back (LBB) method [4] to include the reflections of distributed population densities from partially filled cells. In the LBB method the boundary nodes lie midway between the solid and fluid nodes [4], and fluid particles moving along the links between solid and fluid nodes interact at these boundary nodes. For planar surfaces aligned with one of the lattice directions it can be shown that the LBB rules simulate a hydrodynamic boundary that is located at the boundary nodes, with relative deviations that are second order in the lattice spacing [5]. Several other methods to obtain second order accurate boundary conditions for simple geometries have been suggested in the literature [6–15]. However, most of these methods share the drawback that they require information about the shape of the particle surface. For general three-dimensional objects, the resulting algorithms are complex and not necessarily well defined without additional constraints [7–10]. Extrapolation based methods [11,14,15] are problematic if surfaces are in close proximity, which is often the case in particle suspensions. Other recent developments use a volumetric approach [16–19], initially introduced by Benzi *et al.* [20]. Noble and Torczynski [16] introduce a modified collision operator to account for the interactions with solid obstacles within a lattice cell. Their method incorporates solid-fluid boundary conditions on length scales less than the grid spacing but a theoretical foundation is lacking. Chen *et al.* [17,18] developed a rigorous volumetric formulation of the lattice-Boltzmann method, which can also be applied to nonuniform meshes. However, information about the shape of the solid-fluid interface is required unless it coincides with the boundary between two lattice cells. Xi *et al.*'s approach [19] is applicable to arbitrarily complex geometries, but requires an unstructured grid to model the solid-fluid interface.

The continuous bounce-back rules [3] are also based on a volumetric interpretation [18,20], in which $n_i(\mathbf{r},t)$ represents the mean population density in the Wigner-Seitz cell surrounding node $\mathbf{r}$. However, the CBB rules do not require detailed information about the solid-fluid interface and are therefore applicable to arbitrarily complex geometries. Instead, a continuous variable $\alpha(\mathbf{r})$ is specified at each node, representing the fluid volume fraction associated with that node, and then rules are constructed relating the velocity distribution function after propagation to the postcollision distribution.

In Sec. II a detailed description of the CBB rules is presented. Section III analyzes the accuracy of the CBB rules for systems with planar surfaces and for periodic arrays of disks and spheres. Numerical results are presented that demonstrate that second order accurate results were obtained for flow in narrow channels with aligned walls that do not necessarily coincide with the grid. It is also shown that grid artifacts in simulations of flows past curved surfaces (disks or spheres) are much reduced by the CBB rules. Section IV analyzes the stability of the CBB rules over a large range of shear viscosities and for different commonly used collision operators. Numerical results using the CBB rules show that the exponential relaxation time (ERT) model is stable only for a limited range of kinematic viscosities. However, using a more general collision operator (see Appendix A and Ref. [21]) with separate eigenvalues for viscous and kinetic modes, these instabilities could always be prevented by a

*URL: http://www.che.ufl.edu/~ladd/Ladd.htm

suitable choice of the kinetic mode eigenvalues. Section V concludes this paper with a discussion of the main results.

II. METHODOLOGY

In the lattice-Boltzmann method the state of the system is characterized by the discretized one-particle velocity distribution function $n_i(\mathbf{r},t)$, which describes the density of fluid particles at a lattice node $\mathbf{r}$ at time t with a velocity $\mathbf{c}_i$. The mass density $\rho(\mathbf{r},t)$ and momentum density $\mathbf{j}(\mathbf{r},t)$ are moments of this velocity distribution function:

$$\rho(\mathbf{r},t)=\sum_i n_i(\mathbf{r},t), \tag{1}$$

$$\mathbf{j}(\mathbf{r},t)=\sum_i \mathbf{c}_i n_i(\mathbf{r},t), \tag{2}$$

where $\mathbf{j}(\mathbf{r},t)=\rho(\mathbf{r},t)\mathbf{u}(\mathbf{r},t)$ and $\mathbf{u}(\mathbf{r},t)$ is the macroscopic fluid velocity; the summation runs over the complete set of velocities $\{\mathbf{c}_i\}$. In this paper two velocity sets are used: a nine-velocity model for the two-dimensional simulations and a 19-velocity model for the three-dimensional simulations. Here, each velocity set contains the zero-velocity (rest particles) and the vectors connecting each node with its nearest and next-nearest neighbors.

The time evolution of $n_i(\mathbf{r},t)$ in the presence of an external force (e.g., an externally applied pressure gradient) is governed by the discretized Boltzmann equation [22]

$$n_i(\mathbf{r}+\mathbf{c}_i,t+1)=n_i(\mathbf{r},t)+\Delta_i[\mathbf{n}(\mathbf{r},t)]+f_i(\mathbf{r},t), \tag{3}$$

where $\Delta_i[\mathbf{n}(\mathbf{r},t)]$ describes the change in $n_i(\mathbf{r},t)$ as a result of collisions and $f_i(\mathbf{r},t)$ incorporates the effect of external forces. The collision operator $\Delta_i[\mathbf{n}(\mathbf{r},t)]$ depends on all the n_i's at the node, denoted collectively by $\mathbf{n}(\mathbf{r},t)$. A computationally useful form for the collision operator can be constructed by linearizing about the local equilibrium distribution $n_i^{\mathrm{eq}}(\mathbf{r},t)$ [23], i.e.,

$$\Delta_i[\mathbf{n}(\mathbf{r},t)]=\sum_j L_{ij}[n_j(\mathbf{r},t)-n_j^{\mathrm{eq}}(\mathbf{r},t)], \tag{4}$$

where L_{ij} are the matrix elements of the linearized collision operator $\mathbf{L}$.

The exponential relaxation time or lattice Bhatnagar-Gross-Krook (BGK) collision operator involves a single relaxation time τ, i.e., $L_{ij}=-\tau^{-1}\delta_{ij}$ [24,25],

$$\Delta_i[\mathbf{n}(\mathbf{r},t)]=-\tau^{-1}[n_i(\mathbf{r},t)-n_i^{\mathrm{eq}}(\mathbf{r},t)]. \tag{5}$$

A more general collision operator $\mathbf{L}$ can be defined by constructing a complete set of mutually orthogonal eigenvectors of $\mathbf{L}$ in the velocity space $\{\mathbf{c}_i\}$ [23,26–30], such that

$$\mathbf{n}(\mathbf{r},t)=\sum_k \frac{m_k(\mathbf{r},t)\mathbf{b}_k}{b_k^2}, \tag{6}$$

where $m_k(\mathbf{r},t)$ is the amplitude of mode k and the $\mathbf{b}_k$'s are the eigenvectors. Each mode $m_k(\mathbf{r},t)$ relaxes exponentially at a rate controlled by its associated eigenvalue (see Appendix A). If all the eigenvalues are equal, the collision operator is equivalent to that of the ERT model [Eq. (5)]. Suitable eigenvectors for the two-dimensional nine-speed [27,31] and the three-dimensional 19-speed model are given in Appendix A.

Lattice-Boltzmann models can also be constructed using a linearized collision operator with a smaller number of modes [4,21,32]. Among these are the three-dimensional 19-speed model with only ten modes [21] and its projection onto two dimensions (nine speeds with six modes). Here, the eigenvalues associated with the rest-particle mode and those associated with the kinetic modes are set to -1, projecting them out entirely from the postcollision velocity distribution function. These models cannot be exactly derived from the nine- or 19-mode models in Appendix A, but the difference in the macroscopic behavior occurs only in the error terms. The different models are identified in this paper by the number of velocities and the number of modes; for example, the C9M6 model refers to the nine-speed model with six modes.

Each lattice-Boltzmann update consists of two steps. The first step incorporates collisions and external forces; the velocity distribution function after this step is defined as $n_i^*(\mathbf{r},t)$, i.e.,

$$n_i^*(\mathbf{r},t)=n_i(\mathbf{r},t)+\Delta_i[\mathbf{n}(\mathbf{r},t)]+f_i(\mathbf{r},t). \tag{7}$$

In the second step the postcollision distributions $n_i^*(\mathbf{r},t)$ are propagated to the neighboring nodes in the direction of their velocities $\mathbf{c}_i$,

$$n_i(\mathbf{r}+\mathbf{c}_i,t+1)=n_i^*(\mathbf{r},t). \tag{8}$$

For simplicity all quantities in this paper are given in lattice units; i.e., the nearest-neighbor lattice spacing and the time step are both unity.

An external force density $\mathbf{f}(\mathbf{r},t)=\Sigma_i \mathbf{c}_i f_i(\mathbf{r},t)$ causes some ambiguity in calculating the momentum density. In most published work, the momentum density is measured before the application of the force density [cf. Eq. (2)], but it could equally well be measured afterward, i.e., $\mathbf{j}^*(\mathbf{r},t)=\Sigma_i \mathbf{c}_i n_i^*(\mathbf{r},t)$. It has been shown theoretically [21] and numerically [33] that the most consistent choice is to take the mean of the momentum density before and after forcing, $\mathbf{j}'(\mathbf{r},t)=[\mathbf{j}(\mathbf{r},t)+\mathbf{j}^*(\mathbf{r},t)]/2$. This is equivalent to measuring the momentum density after half the force density is applied:

$$\mathbf{j}'(\mathbf{r},t)=\mathbf{j}(\mathbf{r},t)+\mathbf{f}(\mathbf{r},t)/2. \tag{9}$$

The momentum density $\mathbf{j}'(\mathbf{r},t)$ leads to simulations of forced systems with the same level of accuracy as unforced ones, even when the force is spatially varying [21]. Hence, all simulated flow fields presented in this paper are obtained from the momentum density defined in Eq. (9).

To simulate the interactions between fluid and solid, the lattice-Boltzmann model must be modified to incorporate the boundary conditions imposed on the fluid by the solid phase. The continuous bounce-back rules [3] are based on the link bounce-back method, in which the boundary nodes lie midway between the solid and fluid nodes [4]. The CBB rules

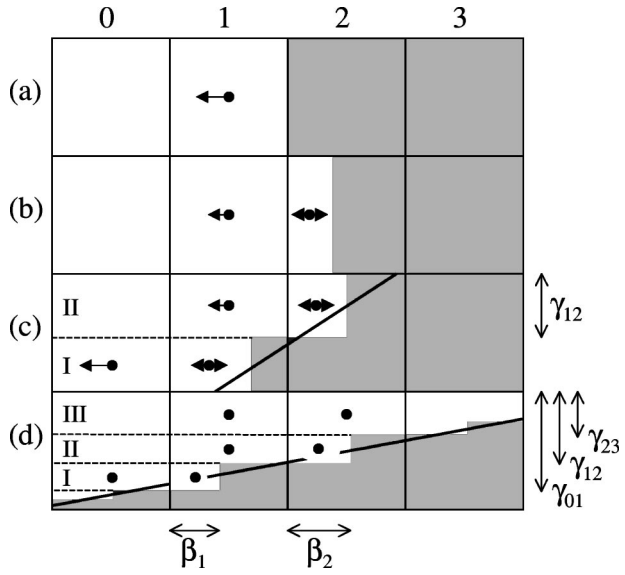

FIG. 1. Schematic representation of the different geometries discussed in the text. The gray areas represent the solid phase, the arrows indicate the population densities after propagation of $n_1^*(\mathbf{r}_1,t)$ [omitted in (d) for clarity], and the solid dots indicate the location of the mean population density in each cell or fraction of a cell. The population densities n_1 and n_2 are moving to the right and left, respectively. The heavy lines in (c) and (d) indicate an inclined boundary represented by the sequence of partially filled cells.

differ from most previous schemes, in that they attempt to model solid-fluid surfaces that are not mapped directly onto the lattice grid. They are illustrated with a set of four adjacent cells with positions $\mathbf{r}_0,\ldots,\mathbf{r}_3$. The propagation of the velocity distribution function in cells 1 and 2 is calculated for increasingly complex geometries. For illustrative purposes the four cells are taken to be along the x axis of a two-dimensional square lattice, but the results are equally valid for the other directions and in three dimensions. The fluid fraction in cell i is denoted by α_i, the velocity vector pointing from $\mathbf{r}_1$ to $\mathbf{r}_2$ by $\mathbf{c}_1$, and that in the opposite direction by $\mathbf{c}_2$; i.e., $\mathbf{r}_2=\mathbf{r}_1+\mathbf{c}_1$ and $\mathbf{c}_2=-\mathbf{c}_1$. Only the case with $\alpha_1>\alpha_2$ is considered explicitly; the update rules for $\alpha_1<\alpha_2$ can be obtained from the mirror image.

A. Basic CBB rules

The basic concept is introduced by assuming that the population densities are uniformly distributed throughout the volume of the Wigner-Seitz cell surrounding each node. Particles are propagated from each location within a cell, and reflected at the solid-fluid interface at the appropriate time. Each particle moves a total distance of 1 or $\sqrt{2}$ lattice spacings in one time step, depending on its speed. If the channel wall coincides with a face of a cell, one has the typical LBB rule [Fig. 1(a)]. Here, $n_1^*(\mathbf{r}_1,t)$ is completely reflected at the interface between cell 1 and 2 and returned to cell 1 with the opposite velocity, i.e., $n_2(\mathbf{r}_1,t+1)=n_1^*(\mathbf{r}_1,t)$. If cell 2 instead contains a fraction of fluid α_2 [Fig. 1(b)], $n_1^*(\mathbf{r}_1,t)$ and $n_1^*(\mathbf{r}_2,t)$ are split into different fractions. For $\alpha_2<0.5$, part of $n_1^*(\mathbf{r}_1,t)$ is propagated to cell 2, another part is reflected at the solid-fluid interface and ends up in cell 2, and a third part is reflected and returned to cell 1. The population density in cell 2, $n_1^*(\mathbf{r}_2,t)$, is completely reflected into cell 1 and $n_2^*(\mathbf{r}_2,t)$ is propagated to cell 1; both densities are multiplied by α_2 to account for mass conservation. Hence, for $\alpha_2<0.5$,

$$n_2(\mathbf{r}_1,t+1)=(1-2\alpha_2)n_1^*(\mathbf{r}_1,t)+\alpha_2 n_1^*(\mathbf{r}_2,t)+\alpha_2 n_2^*(\mathbf{r}_2,t), \tag{10a}$$

$$n_1(\mathbf{r}_2,t+1)=n_1^*(\mathbf{r}_1,t), \tag{10b}$$

$$n_2(\mathbf{r}_2,t+1)=n_1^*(\mathbf{r}_1,t). \tag{10c}$$

For $\alpha_2>0.5$, none of $n_1^*(\mathbf{r}_1,t)$ is returned to cell 1, but $n_1^*(\mathbf{r}_2,t)$ is split into two parts, both reflected at the solid-fluid interface: one part ends up in cell 1 and the other part remains in cell 2. Hence, for $\alpha_2>0.5$,

$$n_2(\mathbf{r}_1,t+1)=(1-\alpha_2)n_1^*(\mathbf{r}_2,t)+\alpha_2 n_2^*(\mathbf{r}_2,t), \tag{11a}$$

$$n_1(\mathbf{r}_2,t+1)=n_1^*(\mathbf{r}_1,t), \tag{11b}$$

$$n_2(\mathbf{r}_2,t+1)=\frac{1-\alpha_2}{\alpha_2}n_1^*(\mathbf{r}_1,t)+\frac{2\alpha_2-1}{\alpha_2}n_1^*(\mathbf{r}_2,t). \tag{11c}$$

Equations (10) and (11) correctly account for the mass transfer between cells of different fluid fractions and reduce to the LBB rules [Fig. 1(a)] when $\alpha_2=0$ or $\alpha_2=1$. However, only first order accuracy is obtained for Poiseuille flow in channels with solid-fluid boundaries at fractional positions. The assumption that the velocity distribution function is uniformly distributed in each cell is insufficient for second order accuracy. This shortcoming is also present in the boundary conditions suggested by Chen *et al.* [17]. Chen [18] overcomes this problem by taking account of gradients in the velocity distribution function, resulting in a higher order scheme.

Similarly, we proceed to a better approximation by taking into account the velocity gradient in the direction of propagation. Assuming that the population density varies linearly along the line connecting cells 1 and 2,

$$n_i^*(x,t)=\frac{2x}{1+\beta}[n_i^*(\mathbf{r}_2,t)-n_i^*(\mathbf{r}_1,t)]+n_i^*(\mathbf{r}_1,t) \tag{12}$$

for $i=1,2$. Here the origin is taken at the center of cell 1, with the positive x axis pointing in the $\mathbf{c}_1$ direction and the distance between the wall and the interface between cells 1 and 2 denoted by β [$\beta=\alpha_2$ for the particular geometry in Fig. 1(b)]. This results in the following propagation rules for $n_1^*(\mathbf{r}_1,t)$, $n_1^*(\mathbf{r}_2,t)$, and $n_2^*(\mathbf{r}_2,t)$:

$$n_2(\mathbf{r}_1,t+1)=\frac{1}{\alpha_1}\int_{2\beta-1/2}^{\beta+1/2}dx\,n_1^*(x,t)+\frac{1}{\alpha_1}\int_{1/2}^{\beta+1/2}dx n_2^*(x,t)$$

$$=(1-\beta)\left[\frac{1-2\beta}{1+\beta}n_1^*(\mathbf{r}_1,t)+\frac{3\beta}{1+\beta}n_1^*(\mathbf{r}_2,t)\right]+\beta n_2^*(\mathbf{r}_2,t), \tag{13a}$$

$$n_1(\mathbf{r}_2,t+1)=\frac{1}{\alpha_2}\int_{-1/2}^{\beta-1/2}dx n_1^*(x,t)$$

$$=\frac{\beta}{\alpha_2}\left[\frac{2}{1+\beta}n_1^*(\mathbf{r}_1,t)+\frac{\beta-1}{1+\beta}n_1^*(\mathbf{r}_2,t)\right], \tag{13b}$$

$$n_2(\mathbf{r}_2,t+1)=\frac{1}{\alpha_2}\int_{\beta-1/2}^{2\beta-1/2}dx n_1^*(x,t)$$

$$=\frac{\beta}{\alpha_2}\left[\frac{2-2\beta}{1+\beta}n_1^*(\mathbf{r}_1,t)+\frac{3\beta-1}{1+\beta}n_1^*(\mathbf{r}_2,t)\right]. \tag{13c}$$

The propagation rules in Eq. (13) are continuous functions of β for $0<\beta<1$; they conserve mass and momentum and reduce to the LBB rules in the limits $\beta=0$ and $\beta=1$. Note that, although the second term between square brackets in Eq. (13b) is negative as a result of the interpolation, the value of $n_1(\mathbf{r}_2,t+1)$ remains positive.

B. CBB rules for general geometries

In most situations of practical interest the system changes rapidly from fluid to solid, typically with only one partially filled cell in between. However, more complicated geometries do arise in which the solid-fluid interface extends over more than one cell. The propagation rules for general geometries can be described by considering only the propagation of $n_1^*(\mathbf{r}_1,t)$ and $n_2^*(\mathbf{r}_2,t)$. It is more convenient to express the propagation rules in terms of population fluxes instead of population densities, where the population flux $f(\mathbf{r},\mathbf{c}_i;\mathbf{r}',\mathbf{c}_j;t)$ is defined as the number (as opposed to the density) of fluid particles that propagates from node $\mathbf{r}$ with velocity $\mathbf{c}_i$ to node $\mathbf{r}'$ with velocity $\mathbf{c}_j$.

An inclined boundary extending over two cells is approximated by the geometry shown in Fig. 1(c). This geometry is completely determined by a single parameter, the height of layer II: $\gamma_{12}=(\alpha_1+\alpha_2)/2$. Then $\beta_1=(\alpha_1-\gamma_{12})/(1-\gamma_{12})$ is the distance between the wall in layer I and the interface between cells 0 and 1 and similarly $\beta_2=\alpha_2/\gamma_{12}$. Since for $\alpha_1>\alpha_2$ the entire postcollision distribution $n_2^*(\mathbf{r}_2,t)$ propagates to cell 1, the population flux $f(\mathbf{r}_2,\mathbf{c}_2;\mathbf{r}_1,\mathbf{c}_2;t)$ is simply $n_2^*(\mathbf{r}_2,t)$ times the fluid volume of cell 2, i.e.,

$$f(\mathbf{r}_2,\mathbf{c}_2,\mathbf{r}_1,\mathbf{c}_2;t)=\alpha_2 n_2^*(\mathbf{r}_2,t). \tag{14}$$

The propagation rule for $n_1^*(\mathbf{r}_1,t)$ is more involved. A fraction of $n_1^*(\mathbf{r}_1,t)$ is reflected in cell 1 (layer I), while the remainder bounces back in cell 2 (layer II). The total propagation rule for $n_1^*(\mathbf{r}_1,t)$ follows from the sum of the contributions from layers I and II, where the propagation rules in each layer are determined by a generalization of the rules given in reference to Fig. 1(b). The total population in layer I of cell 1 with velocity $\mathbf{c}_1$ is $\beta_1(1-\gamma_{12})n_1^*(\mathbf{r}_1,t)$ [Fig. 1(c)]. During the propagation step some of it ends up in cell 1 with velocity $\mathbf{c}_1$, while the remaining part is reflected to either cell 0 or cell 1. The population fluxes can be found from Eq. (13) by substituting β_1 for β, $\mathbf{r}_{i-1}$ for $\mathbf{r}_i$, by adding a multiplicative factor $(1-\gamma_{12})$ to take into account the height of layer I and by discarding α_2 to account for the fact that Eq. (13) is expressed in population densities instead of population fluxes. Hence, for $n_1^*(\mathbf{r}_1,t)$ one finds

$$f^I(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_0,\mathbf{c}_2;t)=\frac{3(1-\beta_1)}{1+\beta_1}\beta_1(1-\gamma_{12})n_1^*(\mathbf{r}_1,t), \tag{15a}$$

$$f^I(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_1,\mathbf{c}_1;t)=\frac{(\beta_1-1)}{1+\beta_1}\beta_1(1-\gamma_{12})n_1^*(\mathbf{r}_1,t), \tag{15b}$$

$$f^I(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_1,\mathbf{c}_2;t)=\frac{(3\beta_1-1)}{1+\beta_1}\beta_1(1-\gamma_{12})n_1^*(\mathbf{r}_1,t). \tag{15c}$$

Similarly, $\gamma_{12}n_1^*(\mathbf{r}_1,t)$ is the total population in layer II of cell 1 with velocity $\mathbf{c}_1$. During the propagation step some of it is propagated to cell 2, while the remainder bounces back to either cell 1 or cell 2. The population fluxes from layer II are therefore

$$f^{II}(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_1,\mathbf{c}_2;t)=\frac{(1-3\beta_2+2\beta_2^2)}{1+\beta_2}\gamma_{12}n_1^*(\mathbf{r}_1,t), \tag{16a}$$

$$f^{II}(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_2,\mathbf{c}_1;t)=\frac{2\beta_2}{1+\beta_2}\gamma_{12}n_1^*(\mathbf{r}_1,t), \tag{16b}$$

$$f^{II}(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_2,\mathbf{c}_2;t)=\frac{2\beta_2(1-\beta_2)}{1+\beta_2}\gamma_{12}n_1^*(\mathbf{r}_1,t). \tag{16c}$$

The total population fluxes are obtained by accumulating the contributions from layers I and II. Note that to obtain Eqs. (15) and (16) an interpolated population density profile was used in layer I of cells 0 and 1 and layer II of cells 1 and 2, respectively, just as for Eq. (13) in reference to Fig. 1(b). This is the reason why a fraction of $n_1^*(\mathbf{r}_1,t)$ can actually remain in cell 1 with the same velocity [cf. Eq. (15b)].

A general boundary extending over any number of cells is approximated by the geometry shown in Fig. 1(d). Although Fig. 1(d) assumes $1>\alpha_0>\alpha_1>\alpha_2>\alpha_3>0$, the results below are valid for any geometry with $\alpha_1\geqslant\alpha_2$, while the update rules for $\alpha_1<\alpha_2$ follow straightforwardly from the mirror image. Three parameters γ_{01}, γ_{12}, and γ_{23} are introduced that completely determine the actual configuration:

$$\gamma_{01}=\begin{cases}1, & \alpha_0=1\\ \max\{\alpha_1,(\alpha_0+\alpha_1)/2\}, & \alpha_0\neq 1,\end{cases} \tag{17a}$$

$$\gamma_{12}=\begin{cases}1, & \alpha_1=1\wedge\alpha_2\neq 0\\ (\alpha_1+\alpha_2)/2, & \alpha_1\neq 1\wedge\alpha_2\neq 0,\\ 0, & \alpha_2=0\end{cases} \tag{17b}$$

$$\gamma_{23}=\begin{cases}0, & \alpha_3=0\\ \min\{\alpha_2,(\alpha_2+\alpha_3)/2\}, & \alpha_3\neq 0.\end{cases} \tag{17c}$$

The values β_1 and β_2 follow from mass conservation:

$$\beta_1=\frac{\alpha_1-\gamma_{12}}{\gamma_{01}-\gamma_{12}}, \quad \gamma_{01}\neq\gamma_{12}, \tag{18a}$$

$$\beta_2=\frac{\alpha_2-\gamma_{23}}{\gamma_{12}-\gamma_{23}}, \quad \gamma_{12}\neq\gamma_{23}. \tag{18b}$$

The entire postcollision distribution $n_2^*(\mathbf{r}_2,t)$ propagates to cell 1 as before, so the population flux $f(\mathbf{r}_2,\mathbf{c}_2;\mathbf{r}_1,\mathbf{c}_2;t)$ is again given by Eq. (14). The propagation rule for $n_1^*(\mathbf{r}_1,t)$ is again a sum of different contributions. A fraction of $n_1^*(\mathbf{r}_1,t)$ is reflected in cell 1 (layer I), another fraction is reflected in cell 2 (layer II), while the remainder is propagated to cell 2 (layer III) [Fig. 1(d)]. The only difference between layer I in Fig. 1(c) and layer I in Fig. 1(d) is the height of the layer. Hence, the propagation rules for $n_1^*(\mathbf{r}_1,t)$ in layer I can be found from Eq. (15) by substituting $\gamma_{01}-\gamma_{12}$ for $1-\gamma_{12}$, i.e.,

$$f^I(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_0,\mathbf{c}_2;t)=\frac{3(1-\beta_1)}{1+\beta_1}\beta_1(\gamma_{01}-\gamma_{12})n_1^*(\mathbf{r}_1,t), \tag{19a}$$

$$f^I(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_1,\mathbf{c}_1;t)=\frac{(\beta_1-1)}{1+\beta_1}\beta_1(\gamma_{01}-\gamma_{12})n_1^*(\mathbf{r}_1,t), \tag{19b}$$

$$f^I(\mathbf{r}_1,\mathbf{c}_1,\mathbf{r}_1,\mathbf{c}_2;t)=\frac{(3\beta_1-1)}{1+\beta_1}\beta_1(\gamma_{01}-\gamma_{12})n_1^*(\mathbf{r}_1,t). \tag{19c}$$

Each population flux is well behaved in the limit $\gamma_{01}=\gamma_{12}$, i.e., when the height of layer I vanishes and $f^I(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_1,\mathbf{c}_1;t)=f^I(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_1,\mathbf{c}_2;t)=f^I(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_0,\mathbf{c}_2;t)=0$. Similarly, the propagation rules for $n_1^*(\mathbf{r}_1,t)$ in layer II are found from Eq. (16) by substituting $\gamma_{12}-\gamma_{23}$ for γ_{12}, i.e.,

$$f^{II}(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_1,\mathbf{c}_2;t)=\frac{(1-3\beta_2+2\beta_2^2)}{1+\beta_2}(\gamma_{12}-\gamma_{23})n_1^*(\mathbf{r}_1,t), \tag{20a}$$

$$f^{II}(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_2,\mathbf{c}_1;t)=\frac{2\beta_2}{1+\beta_2}(\gamma_{12}-\gamma_{23})n_1^*(\mathbf{r}_1,t), \tag{20b}$$

$$f^{II}(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_2,\mathbf{c}_2;t)=\frac{2\beta_2(1-\beta_2)}{1+\beta_2}(\gamma_{12}-\gamma_{23})n_1^*(\mathbf{r}_1,t). \tag{20c}$$

Finally, the total population in layer III of cell 1 with velocity $\mathbf{c}_1$ is simply propagated to cell 2 [Fig. 1(d)], i.e.,

$$f^{III}(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_2,\mathbf{c}_1;t)=\gamma_{23}n_1^*(\mathbf{r}_1,t). \tag{21}$$

The total population fluxes are again obtained by accumulating the contributions from the different layers [Eqs. (19), (20), and (21)], which results in the following general set of propagation rules for the postcollision distributions $n_1^*(\mathbf{r}_1,t)$ and $n_2^*(\mathbf{r}_2,t)$:

$$f(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_0,\mathbf{c}_2;t)=\frac{3(1-\beta_1)(\alpha_1-\gamma_{12})}{1+\beta_1}n_1^*(\mathbf{r}_1,t), \tag{22a}$$

$$f(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_1,\mathbf{c}_1;t)=\frac{(\beta_1-1)(\alpha_1-\gamma_{12})}{1+\beta_1}n_1^*(\mathbf{r}_1,t), \tag{22b}$$

$$f(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_1,\mathbf{c}_2;t)=\left[\frac{(3\beta_1-1)(\alpha_1-\gamma_{12})}{1+\beta_1}+\frac{(1-3\beta_2+2\beta_2^2)(\gamma_{12}-\gamma_{23})}{1+\beta_2}\right]n_1^*(\mathbf{r}_1,t), \tag{22c}$$

$$f(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_2,\mathbf{c}_1;t)=\left[\gamma_{23}+\frac{2(\alpha_2-\gamma_{23})}{1+\beta_2}\right]n_1^*(\mathbf{r}_1,t), \tag{22d}$$

$$f(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_2,\mathbf{c}_2;t)=\frac{2(1-\beta_2)(\alpha_2-\gamma_{23})}{1+\beta_2}n_1^*(\mathbf{r}_1,t), \tag{22e}$$

$$f(\mathbf{r}_2,\mathbf{c}_2;\mathbf{r}_1,\mathbf{c}_2;t)=\alpha_2 n_2^*(\mathbf{r}_2,t). \tag{22f}$$

Mass conservation follows straightforwardly from Eq. (22f) and by verifying that

$$\begin{aligned}\alpha_1 n_1^*(\mathbf{r}_1,t)=&f(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_0,\mathbf{c}_2;t)+f(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_1,\mathbf{c}_1;t)\\&+f(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_1,\mathbf{c}_2;t)+f(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_2,\mathbf{c}_1;t)\\&+f(\mathbf{r}_1,\mathbf{c}_1;\mathbf{r}_2,\mathbf{c}_2;t)\end{aligned} \tag{23}$$

[cf. Eqs. (22a–e)].

The total velocity distribution function after propagation is obtained by accumulating the population fluxes from the postcollision distributions $n_i^*(\mathbf{r},t)$ and $n_{i'}^*(\mathbf{r}+\mathbf{c}_i,t)$ for each $\mathbf{r}$ and each pair of velocities $\mathbf{c}_i$ and $-\mathbf{c}_i$ (denoted by i'). From Eq. (22) it follows that, depending on the local geometry, there are up to five different population fluxes that contribute to $n_i(\mathbf{r},t+1)$ for each lattice node $\mathbf{r}$ and each velocity $\mathbf{c}_i$:

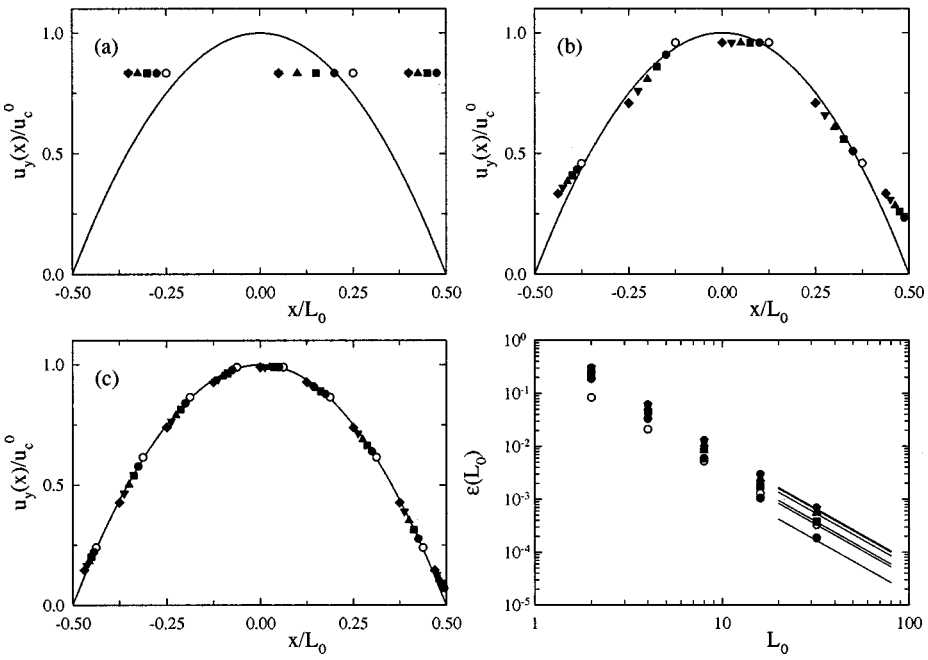

FIG. 2. Poiseuille flow in a shifted channel for a kinematic shear viscosity $\nu=1/6$. The open circles are the simulation results for a lattice where the walls coincide with the interfaces between two cells. The solid symbols are for lattices shifted with respect to the wall in increments of 1/10: in increasing order these go from solid circles (0.1) to diamonds (0.5). (a)–(c) Flow profile $u_y(x)$ for a channel width $L_0=2$, 4, and 8, relative to the exact center flow velocity $u_c^0=L_0^2\nabla_y p/8\rho\nu$. The solid line is the exact flow profile of Eq. (28). (d) Root mean square error $\epsilon(L_0)$ in the flow profile. The solid lines indicate the asymptotic $1/L_0^2$ decay of $\epsilon(L_0)$ for the different shifts [Eq. (35)].

$$n_i(\mathbf{r},t+1)=\{f(\mathbf{r}-\mathbf{c}_i,-\mathbf{c}_i;\mathbf{r},\mathbf{c}_i;t)+f(\mathbf{r}-\mathbf{c}_i,\mathbf{c}_i;\mathbf{r},\mathbf{c}_i;t)+f(\mathbf{r},-\mathbf{c}_i;\mathbf{r},\mathbf{c}_i;t)+f(\mathbf{r},\mathbf{c}_i;\mathbf{r},\mathbf{c}_i;t)+f(\mathbf{r}+\mathbf{c}_i,-\mathbf{c}_i;\mathbf{r},\mathbf{c}_i;t)\}/\alpha(\mathbf{r}). \quad (24)$$

Equations (17), (18), (22), and (24) completely determine the CBB rules for any geometry, including those in Figs. 1(a–c).

A drawback of the proposed boundary condition is that the limits for $\alpha_i\to 0$ or $\alpha_i\to 1$ are not continuous; for example, the shoulder in cells 1 and 2 of height $1-\gamma_{12}$ [Fig. 1(c)] vanishes discontinuously when $\alpha_1=1$ or $\alpha_2=0$. Although a different choice of γ_{12} can prevent this discontinuity, this particular choice was made to maintain a continuous profile in the general case shown in Fig. 1(d). The impact of this discontinuity was tested numerically and found to be small.

Note that the multiplicative factors in Eq. (22) depend only on the local geometry and have to be calculated just once for a fixed geometry. Once calculated, the propagation step is almost as straightforward as the original LBB rules, the only difference being that fractions of each population may propagate to more than one lattice node and velocity. The complete propagation step can be written compactly as a single matrix multiplication

$$\mathbf{N}(t+1)=\mathbf{T}\cdot\mathbf{N}^*(t), \quad (25)$$

with the velocity distributions at all the nodes written in terms of a single vector $\mathbf{N}(t)$, i.e.,

$$\mathbf{n}(\mathbf{r},t)=(n_0(\mathbf{r},t),\ldots,n_{J-1}(\mathbf{r},t)),\quad \mathbf{r}=\mathbf{r}_0,\ldots,\mathbf{r}_{N-1},$$

$$\mathbf{N}(t)=(\mathbf{n}(\mathbf{r}_0,t),\ldots,\mathbf{n}(\mathbf{r}_{N-1},t)), \quad (26)$$

and similar expressions for the postcollision distributions $\mathbf{n}^*(\mathbf{r},t)$ and $\mathbf{N}^*(t)$. Here J denotes the number of velocities in the lattice-Boltzmann model. Note that in this vector notation lower case symbols represent properties of the individual nodes and upper case symbols represent properties of the entire lattice. The matrix $\mathbf{T}$ can be stored compactly and the multiplication can be performed efficiently, since each column or row of $\mathbf{T}$ has a maximum of five nonzero elements, determined by the multiplicative factors in Eq. (22) [cf. Eq. (24)].

III. RESULTS

A. Two-dimensional flow in aligned channels

The basic CBB rules of Eq. (13) were tested for Poiseuille flow in aligned channels, i.e., in channels that were parallel to but not coincident with one of the lattice vectors. Simulations were performed with different channel widths, shear viscosities, and offsets between the wall and the nearest lattice node. The results were obtained using the C9M6 model (equivalent to the C19M10 model [21] projected down to two dimensions), where the postcollision distribution is composed of the conserved modes (mass and momentum density) and the viscous modes (two shear modes and one bulk mode). The eigenvalue associated with the shear modes, λ_S, sets the kinematic shear viscosity $\nu=-(1+2/\lambda_S)/6$. The eigenvalue associated with the bulk mode was set to -1, corresponding to a kinematic bulk viscosity $\nu_B=1/9$. The results were compared with those obtained with the nine-speed ERT model [24,25] for the same shear viscosity $\nu=(2\tau-1)/6$ [Eq. (5)].

Figures 2(a)–(c) show the flow profile $u_y(x)$ for Poiseuille flow in channels of integer widths $L_0=\Sigma_{\mathbf{r}\in L_0}\alpha(\mathbf{r})=2$, 4, and 8, for a kinematic shear viscosity $\nu=1/6$. In this case, $\lambda_S=-\tau=-1$, and the C9M6 model and the ERT model are equivalent. Here, the y axis points in the direction of the pressure gradient and the walls are placed at $x=\pm L_0/2$. It can be seen that even for a narrow channel ($L_0\geqslant 4$) the agreement with the analytic result is good; the larg-

est deviations occur for channels shifted by one-half of a lattice spacing. The root mean square (rms) error in the flow profile,

$$\epsilon(L_0)=\frac{1}{u_c^0}\left\{\frac{1}{L_0}\sum_{\mathbf{r}\in L_0}\alpha(\mathbf{r})[u_s(\mathbf{r})]^2\right\}^{1/2}, \tag{27}$$

is shown in Fig. 2(d), where the slip velocity $u_s(\mathbf{r})=u_y(\mathbf{r})-u_y^0(\mathbf{r})$ is defined as the difference between the simulated flow field $u_y(\mathbf{r})$ and the exact flow field

$$u_y^0(\mathbf{r})=u_c^0\left(1-\frac{4x^2}{L_0^2}\right). \tag{28}$$

Here, the exact flow velocity at the center of the channel is given by $u_c^0=L_0^2\nabla_y p/8\rho\nu$. For $L_0\geq 4$, $\epsilon(L_0)$ is less than 6%, while the error in the location of the hydrodynamic boundary, that is, the location of the zero-velocity plane with respect to the solid-fluid interface, is less than 0.06 lattice spacings. The solid lines in Fig. 2(d) have a slope of -2, indicating that the CBB rules give asymptotically second order convergence for Poiseuille flow in channels where the walls are aligned but not commensurate with the grid. Similar results were obtained for Poiseuille flow in channels of noninteger widths [43].

Theoretical analysis of the LBB rules for two-dimensional channel flow has shown that the hydrodynamic boundary is located at the boundary nodes, i.e., at the midpoints of links connecting lattice nodes on either side of the solid-fluid interface, with relative deviations of order L_0^{-2} [5]. Furthermore, it has been shown that the velocity field deviates from the exact solution by a constant slip velocity u_s, independent of channel width and position within the channel [5]. This leads to second order convergence in $\epsilon(L_0)$ for channels where the walls coincide with the interface between two cells [see open circles in Fig. 2(d)].

A similar analysis of the CBB rules for two-dimensional channel flow in aligned channels with arbitrary widths (including noninteger values) gives qualitatively similar results. The flow field in the bulk channel (the collection of cells that are completely filled with fluid) has a parabolic profile, which can be characterized by the hydrodynamic width of the channel, L, and the flow velocity u_c at the hydrodynamic center of the channel, x_c (Fig. 3), i.e.,

$$u_y(\mathbf{r})=u_c[1-4(x-x_c)^2/L^2]. \tag{29}$$

The parameters u_c, L, and x_c depend on L_0, ν, the collision model being used, and the fluid fractions α_1 and α_2 in the boundary cells (Fig. 3). The maximum flow velocity u_c relative to its exact value u_c^0 is given by

$$\frac{u_c}{u_c^0}=1+\frac{4}{3L_0^2}(6\nu)^n+\frac{C_1-1}{L_0^2}+\frac{C_2^2}{L_0^4}, \tag{30}$$

with $n=1$ for the C9M6 model and 2 for the ERT model and

$$C_1=2[\alpha_1(\alpha_1-1)+\alpha_2(\alpha_2-1)], \tag{31a}$$

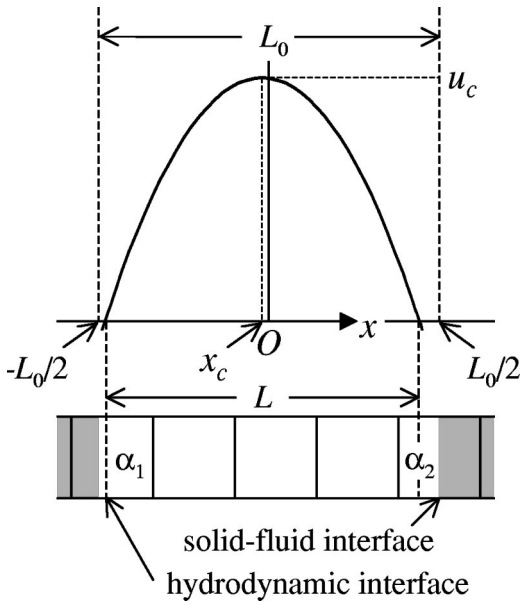

FIG. 3. Poiseuille flow in a channel with arbitrary width L_0. The flow profile in the bulk channel $u_y(\mathbf{r})$ can be characterized by the hydrodynamic width of the channel, L, and the flow velocity u_c at the hydrodynamic center of the channel, x_c. The gray area is the solid wall and α_1 and α_2 are the fluid fractions in the first and last cells.

$$C_2=\alpha_2(\alpha_2-1)-\alpha_1(\alpha_1-1). \tag{31b}$$

Note that u_c is the maximum flow velocity (Fig. 3) inferred from the parabolic profile and should not be confused with the slip velocity $u_s(\mathbf{r})$ defined in Eq. (27) or the theoretical centerline velocity u_c^0. The hydrodynamic channel width L relative to its exact value L_0 is given by

$$\frac{L}{L_0}=\left(\frac{u_c}{u_c^0}\right)^{1/2}, \tag{32}$$

while the hydrodynamic center of the channel follows from

$$x_c=\frac{C_2}{2L_0}. \tag{33}$$

The coefficients C_1 and C_2 vanish when both walls coincide with interfaces between adjacent cells. Then, $x_c=0$ and Eqs. (30) and (32) reduce continuously to the corresponding equation for the LBB rules [21]. Note that these results differ from those obtained by He *et al.* [5], because in their analysis these authors did not add half the force density to the momentum density before calculating the flow profile [21]; if they had their results would be equivalent with Eqs. (30) and (32) in the case $\alpha_1=\alpha_2=0$.

In the partially filled boundary cells deviations from Eq. (29) occur. A correction $\Delta u_y[\alpha(\mathbf{r})]$ must be added to the right hand side of Eq. (29) to account for deviations from the parabolic profile. Here,

$$\frac{L_0^2\Delta u_y(\alpha)}{u_c^0}=(\alpha-1)(12\nu+3\alpha+1). \tag{34}$$

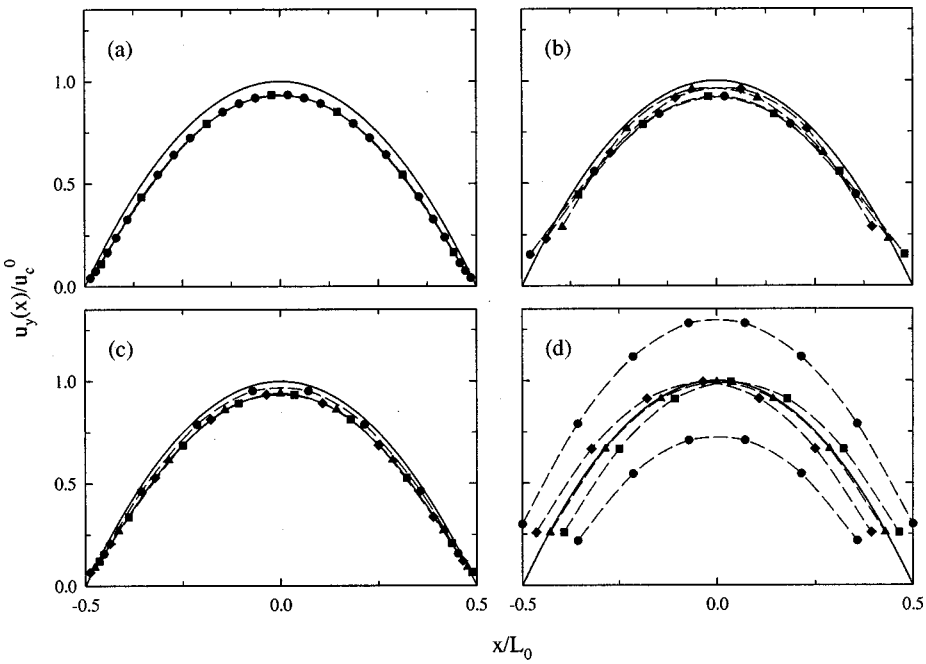

FIG. 4. Poiseuille flow in an inclined channel. Flow profile $u_y(x)/u_c^0$ for a channel width $L_0=24/5$ [(a) CBB rules; (b) LBB rules] and $L_0=7/\sqrt{2}$ [(c) CBB rules; (d) LBB rules], relative to the exact center flow velocity $u_c^0=L_0^2\nabla_y p/8\rho\nu$. The symbols are for lattices shifted with respect to the wall in the direction of one of the lattice vectors in increments of 1/4: in increasing order these go from circles to diamonds. The solid line is the exact result of Eq. (28). The circles in (d) are for the lattice where the walls coincide with the lattice nodes. Here the upper and the lower curves correspond to lattices where the half-occupied boundary cells are considered to be fluid and solid, respectively.

Note that $\Delta u_y(\alpha)$ vanishes for $\alpha=1$, and that the right hand side of Eq. (34) is only a function of ν and α; hence $\Delta u_y(\alpha)/u_c^0=O(L_0^{-2})$.

Several observations can be made from the results obtained so far. Equation (30) shows that $u_c/u_c^0=1+O(L_0^{-2})$, i.e., u_c approaches its exact value for large values of L_0, with deviations that are of order L_0^{-2}. Equations (30), (32), and (33) show that $L/L_0=1+O(L_0^{-2})$, while $x_c/L_0=O(L_0^{-2})$. Hence, the hydrodynamic boundary is located at the actual position of the solid-fluid interface with relative deviations that are of order L_0^{-2}. Equation (30) shows that $u_c/u_c^0=O(\nu)$ for the C9M6 model and $O(\nu^2)$ for the ERT model, making the ERT model less suitable for large shear viscosities. A special shear viscosity can be identified for $\alpha_1=\alpha_2=\alpha$, such that x_c vanish identically and $u_c=u_c^0$. For the C9M6 model this critical kinematic shear viscosity is $1/8+\alpha(1-\alpha)/2$, while for the ERT model it equals $[3+12\alpha(1-\alpha)]^{1/2}/12$. However, in general no such critical viscosity exists. Note that $L<L_0$ for shear viscosities smaller than the critical value. Finally, from Eqs. (27)–(34) it follows that the rms error in the flow profile for Poiseuille flow in aligned channels with arbitrary widths is given by

$$\epsilon(L_0)=\frac{1}{L_0^2}\left\{\left[\frac{4}{3}(6\nu)^n+C_1-1\right]^2+\frac{4}{3}C_2^2+O(L_0^{-1})\right\}^{1/2}, \tag{35}$$

indicating asymptotic second order convergence even when the walls are not commensurate with the grid [Fig. 2(d)]. Here, $n=1$ for the C9M6 model and 2 for the ERT model.

Equations (29)–(34) have been tested extensively for both integer as well as noninteger channel widths ranging from $L_0=4$ to 32, for different values of α_1 and α_2, for kinematic shear viscosities from $\nu=5/3$ down to $\nu=1/60\,000$, and for both the C9M6 model and the ERT model. The simulations using the C9M6 model agreed with Eqs. (29)–(34) for the entire shear viscosity range. However, when the ERT model was used, numerical instabilities were observed for $\nu<1/200$ and certain geometries with partially filled boundary cells. The nature of these instabilities and a way to improve the stability are discussed in more detail in Sec. IV.

B. Two-dimensional flow in inclined channels

The general CBB rules of Eqs. (17), (18), (22), and (24) for an inclined interface were tested for Poiseuille flow in inclined channels. Here, the boundaries and pressure gradient were placed at an angle θ with respect to one of the lattice vectors. All simulations used a kinematic shear viscosity $\nu=1/6$. Results were obtained for $\tan(\theta)=1/2$, 3/4, and 1 with channel widths of $L_0=12n/\sqrt{5}$, $24n/5$, and $7n/\sqrt{2}$, respectively, at different resolutions $n=1,2,4,8$. In addition, each system was simulated including a shift of 1/4, 1/2, and 3/4 lattice units in the direction of one of the lattice vectors.

Figure 4 shows the flow profile $u_y(x)$ for $\tan(\theta)=3/4$ and 1 at the lowest resolution ($n=1$). It shows that the accuracy of the CBB rules is insensitive to the actual position of the interface with respect to the lattice. In contrast, the LBB method is sensitive to the location of the interface relative to the position of the nodes, particularly when $\tan(\theta)=1$ and the wall coincides with the lattice nodes; the flow profile is about 40% too large if the half-occupied boundary cells are considered to be fluid and about 50% too small if they are treated as solid [Fig. 4(d)].

The flow field in the bulk fluid can be characterized by Eq. (29), just as for channels with aligned interfaces. For $\tan(\theta)=1$ and the walls coinciding with the lattice nodes the flow profile in the bulk channel exactly matches that of Poiseuille flow in an aligned channel with the same width and $\alpha_1=\alpha_2=0.5$. Hence, for this particular geometry and shear viscosity, $u_c/u_c^0=1-2/3L_0^2$, $L/L_0=1-1/3L_0^2+O(L_0^{-4})$, and $x_c=0$, i.e., the hydrodynamic boundary is still located at

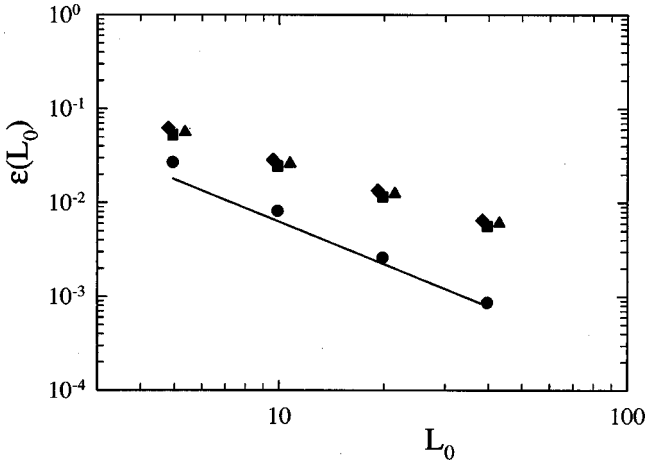

FIG. 5. Root mean square error $\epsilon(L_0)$ in the flow profile for $\tan(\theta)=1$ and the walls coinciding with the lattice nodes (circles) with its $L_0^{-3/2}$ asymptote (solid line), for $\tan(\theta)=1$ and the walls not commensurate with the grid (squares), for $\tan(\theta)=3/4$ (diamonds), and for $\tan(\theta)=1/2$ (triangles). Only one curve for each angle is plotted for $\tan(\theta)\neq 1$ or $\tan(\theta)=1$ and the walls not commensurate with the grid, since the results for different shifts were virtually indistinguishable.

the actual position of the solid-fluid interface with relative deviations that are of order L_0^{-2}. However, the velocity in the partially filled boundary cells deviates from Eq. (34). More specifically, it was found that $\Delta u_y(\alpha)=O(L_0)$ resulting in an asymptotic rms error in the flow profile of order $L_0^{-3/2}$ (Fig. 5). For the more general inclined channels [$\tan(\theta)\neq 1$ or $\tan(\theta)=1$ and the walls not commensurate with the grid] sequences of partially filled cells with different volume fractions occur as sketched in Figs. 1(c–d). In this case only first order convergence is obtained; $u_c/u_c^0=1+O(L_0^{-1})$, $L/L_0=1+O(L_0^{-1})$, and $x_c/L_0=O(L_0^{-1})$, i.e., the hydrodynamic boundary in a very wide channel is no longer located at the actual position of the solid-fluid interface, and the rms error in the flow profile is of order L_0^{-1} (Fig. 5). The lack of second order convergence for Poiseuille flow in inclined channels is a result of several approximations in the derivation of the general CBB rules: from the stepwise implementation of an inclined solid-fluid interface, from taking into account only the velocity gradient in the direction of propagation, and from treating the propagation in the [110] directions in a manner similar to that in the orthogonal directions. However, our approach was motivated by the need for simple but accurate boundary conditions for complex and not necessarily stationary surfaces. A more elaborate volumetric approach [18] should be able to maintain second order convergence for Poiseuille flow in inclined channels, but at the price of a substantial increase in complexity. Such a scheme requires the location and orientation of the surface (as opposed to only the fluid fraction in each cell) as well as a two-dimensional interpolation scheme to obtain the correct distribution of the population density in the boundary cells. As a result the CBB rules are only first order convergent for general geometries. The primary advantage of the CBB rules is the insensitivity of the flow field to the position of the solid-fluid interface with respect to the underlying lattice.

C. Drag coefficient for a periodic array of disks or spheres

The general CBB rules of Eqs. (17), (18), (22), and (24) were also tested by calculating the drag coefficient F_D of a square array of disks and a cubic array of spheres for different system sizes, shear viscosities, positions with respect to the lattice and different collision operators. The results for disks were obtained with the same C9M6 model as used in Secs. III A and III B. Those for spheres were obtained using the C19M10 model [21], i.e., a three-dimensional 19-speed model with ten modes: the conserved modes (mass and momentum density) and the viscous modes (five shear modes and one bulk mode). The results are compared with those obtained with the nine-speed and 19-speed ERT models [Eq. (5)]. The same equations relating the relaxation time τ and the eigenvalue associated with the shear modes, λ_S, with the kinematic shear viscosity also hold for the C19M10 model and the 19-speed ERT model.

The reduced drag coefficient $F_D'=F_D/\nu\langle j_y(\mathbf{r})\rangle=-L_0^D\nabla_y p/\nu\langle j_y(\mathbf{r})\rangle$ of a periodic array of disks ($D=2$) or spheres ($D=3$) can be obtained directly from the mean momentum flow $\langle j_y(\mathbf{r})\rangle=L_0^{-D}\Sigma_{r\in V}\alpha(\mathbf{r})j_y(\mathbf{r})$ in a periodic lattice with a unit cell that contains one disk or sphere. Here, L_0 is the length of the unit cell in each dimension. Figure 6 shows F_D' for a periodic array of disks and spheres as a function of system size and for different positions of the center of the particle with respect to the lattice. The nominal radius R_0 of the disk was $3L_0/8$ and that of the sphere was $7L_0/16$, giving nominal porosities of $\phi_0=0.44$ and $\phi_0=0.35$, respectively; $L_0=4n$, with $n=1,\ldots,10$, and the kinematic shear viscosity was set to 1/6. The results show two important improvements over the LBB method: (i) The drag coefficient obtained with the CBB rules is virtually independent of the position of the center of the particle with respect to the lattice, and (ii) the error in F_D' is much smaller and F_D' itself converges much more smoothly to its asymptotic value for large L_0. For a cubic array of spheres the spread in F_D' was even smaller than for disks, most probably due to a higher degree of averaging over the different types of boundaries.

Figure 6 shows that F_D' converges asymptotically to a value that is slightly different from the theoretical drag coefficient [34,35] at the nominal porosity. This implies that the hydrodynamic boundary has been displaced from the nominal radius of the disks or spheres. For flow through a periodic array of disks or spheres a hydrodynamic radius R can be defined such that the theoretical drag coefficient [34,35] for the porosity based on R equals the value found in the simulations. This hydrodynamic radius varies with shear viscosity, particle size, and particle position relative to the lattice. However, for large enough system sizes the difference between R and R_0 converges to a constant value Δ for each shear viscosity and particle position, independent of system size, with deviations that are of order L_0^{-1}, i.e.,

$$R-R_0=\Delta+O(L_0^{-1}). \tag{36}$$

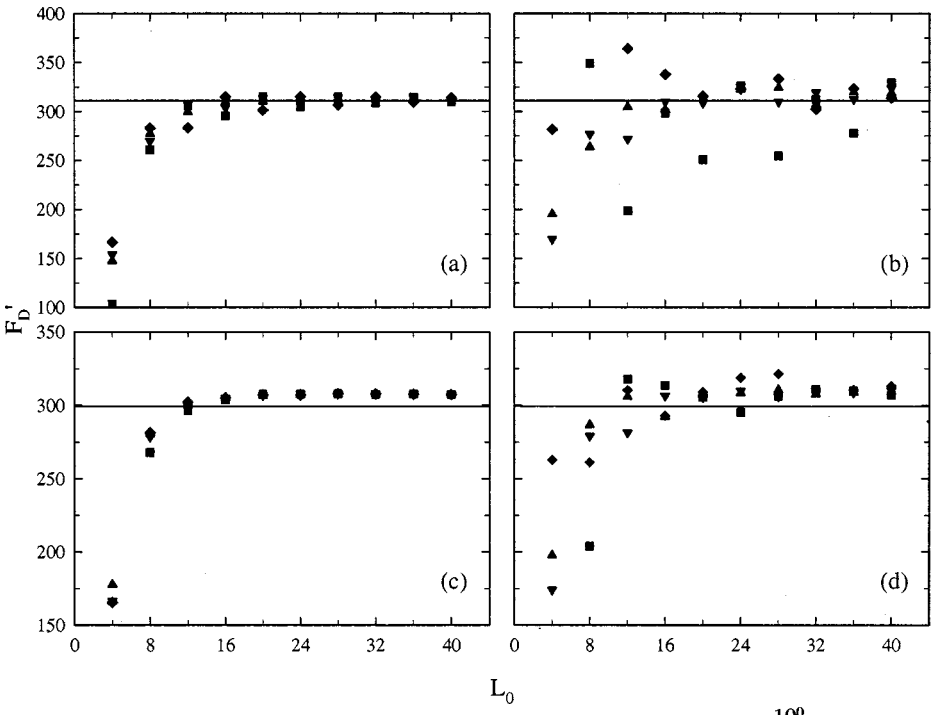

FIG. 6. Reduced drag coefficient F_D' of a periodic array of disks [(a) CBB rules; (b) LBB rules] and spheres [(c) CBB rules; (d) LBB rules] as a function of system size and for different positions of the center of the particle with respect to the lattice: in the center of a cell (♦), in the corner of a cell (■), and at two random positions (▲ and ▼). The kinematic shear viscosity was set to 1/6. The solid lines are the theoretical results [34,35] at the nominal porosities.

For $\nu=1/6$ the value of Δ is close to zero (Fig. 6), but significant deviations can occur for arbitrary values of ν. Figure 7 shows the relative error in R with respect to the asymptotic hydrodynamic radius $R_0+\Delta$ for the same systems as in Fig. 6 and for $\nu=1/6$. For the disks the value of Δ was zero, within the accuracy of the simulation. For the spheres a value $\Delta=0.1$ was found by fitting R with Eq. (36) over the entire range $L_0=4,\ldots,40$. For both the disks and the spheres, Δ was found to be nearly independent of the particle position. Figure 7 shows that $[R-(R_0+\Delta)]/R_0$ decreases as L_0^{-2}. Moreover, the relative error in the F_D' decreases as L_0^{-2} with respect to its asymptotic value (not shown), just like the relative error in R with respect to $R_0+\Delta$. For the LBB rules, qualitatively similar results were found, albeit with larger errors and larger fluctuations in the error.

Figure 8 shows F_D' for a square array of disks at different shear viscosities, for both the C9M6 model and the ERT model. The disks were located at the center of a lattice cell; $R_0=3L_0/8$ and $L_0=4(2)^n$, with $n=0,\ldots,5$. The results show that the simulation underestimates the value of F_D' for kinematic shear viscosities larger than about 1/6 and overestimates it for kinematic shear viscosities smaller than 1/6. This is equivalent to saying that the hydrodynamic radius increases as the shear viscosity decreases, while it is roughly equal to the nominal radius for $\nu=1/6$. Figure 8 shows that the deviations between F_D' and its theoretical value [34] at the nominal porosity, $F_{D,0}'$, increase rapidly for kinematic shear viscosities deviating from 1/6. This large deviation and the slow decay to their asymptotic values makes it impractical to study the error in the drag coefficient and the hydrodynamic radius with respect to their asymptotic values as was done above for $\nu=1/6$. Instead, Fig. 9(a) shows the relative error in F_D' with respect to $F_{D,0}'$. It shows first order convergence with increasing system size indicating that the

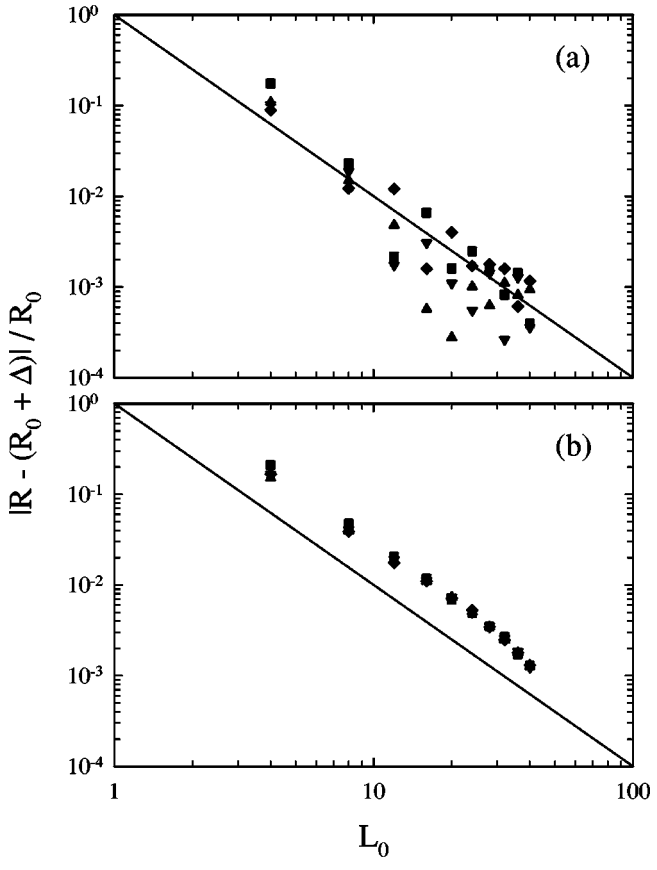

FIG. 7. Relative error in the hydrodynamic radius R with respect to its asymptotic value $R_0+\Delta$ for a periodic array of disks (a) and spheres (b) (both with the CBB rules) and for different positions of the disk with respect to the lattice: centered in the center of a cell (♦), centered in the corner of a cell (■), and centered at two random positions in the cell (▲ and ▼). For the disks a value of $\Delta=0$ was used, while for the spheres $\Delta=0.1$. The solid line indicates second order convergence with increasing system size.

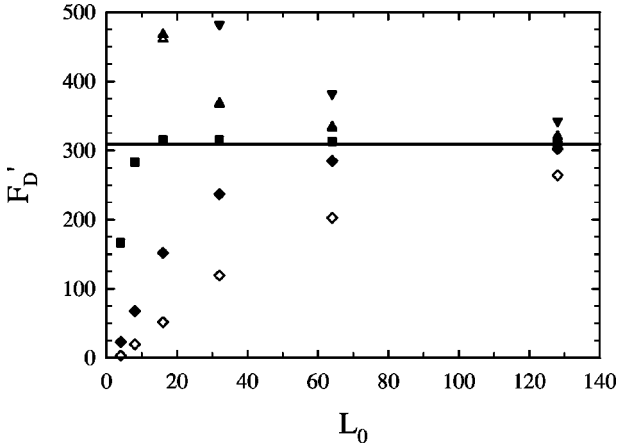

FIG. 8. Reduced drag coefficient F_D' for a square array of disks as a function of system size and shear viscosity. The disks were centered with respect to the lattice cells. The kinematic shear viscosities were 5/3 (♦), 1/6 (■), 1/60 (▲), and 1/600 (▼). The solid symbols are for the C9M6 model and the open symbols for the ERT model (unstable at $\nu=1/600$ and equivalent to the C9M6 model for $\nu=1/6$). The straight solid line is the theoretical value at the nominal porosity [34].

asymptotic drag coefficient deviates significantly from the theoretical value based on the nominal porosity.

The large error in F_D' with respect to $F_{D,0}'$ clearly shows that R deviates significantly from the nominal radius R_0 for smaller particle sizes and kinematic shear viscosities not close to 1/6. It is clear that this shift in the location of the hydrodynamic boundary must be accounted for if quantitative results are to be obtained with computationally useful particle sizes (typically less than ten lattice units) and kinematic shear viscosities significantly different from 1/6. An effective way to calibrate the particle radius is to use the hydrodynamic radius obtained by fitting the drag coefficient at a low porosity, where the dependence on particle size and shear viscosity is much less severe [21,33]. Figure 9(b) shows the relative error in F_D' of a square array of disks with respect to the theoretical value based on the hydrodynamic radius of the same size disks at a porosity of 10%. It shows that, although the convergence with increasing system size is still roughly first order, the error is reduced by about an order of magnitude (except for $\nu=1/6$, where $\Delta\approx 0$). The reason that no second order convergence is found is that the value of Δ differs slightly for similar particles at different porosities. Hence, the hydrodynamic radius obtained by fitting the drag coefficient at a low porosity is still different from $R_0+\Delta$.

Figures 8 and 9 also show that the ERT model is much less suitable for kinematic shear viscosities larger than 1/6, but that both models give almost identical results for $\nu<1/6$, just as for Poiseuille flow in an arbitrary channel [cf. Eq. (30)]. However, the ERT model was numerically unstable for a kinematic shear viscosity of 1/600, a point that is discussed in more detail in the next section.

IV. STABILITY OF THE CBB RULES

In most of the simulations described above numerical instabilities were observed below a certain shear viscosity.

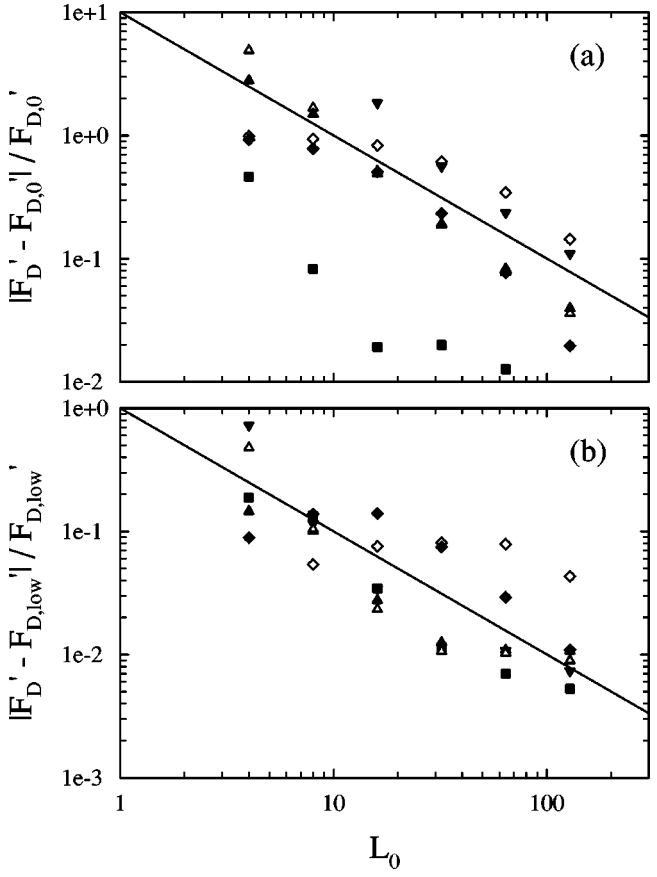

FIG. 9. Relative error in F_D' for a square array of disks as a function of system size and shear viscosity. The disks were centered with respect to the lattice cells. The error in F_D' was calculated with respect to its theoretical value based on the nominal radius (a) and on the hydrodynamic radius obtained from a calibration run at a porosity of 10% (b). The kinematic shear viscosities were 5/3 (♦), 1/6 (■), 1/60 (▲), and 1/600 (▼). The solid symbols are for the C9M6 model and the open symbols for the ERT model (unstable at $\nu=1/600$ and equivalent to the C9M6 model for $\nu=1/6$). The straight solid line indicates first order convergence with increasing system size.

These instabilities were not present for the LBB rules and are consequently a result of the boundary conditions in the CBB model. To summarize: Poiseuille flow in shifted channels, simulated with the C9M6 model, was found to be stable over the entire range of kinematic shear viscosities from $\nu=5/3$ down to $\nu=1/60\,000$. However, the simulations using the ERT model showed numerical instabilities for $\nu<1/200$ in certain geometries with partially filled boundary cells. In more general geometries, like Poiseuille flow in inclined channels and flow through a periodic array of disks or spheres, instabilities showed up for both the C9M6 and the C19M10 models and the ERT model, although more severely for the latter; the C9M6 and C19M10 models were found to be stable for $\nu\geqslant 1/600$, while the critical kinematic shear viscosity for the ERT model lay between 1/60 and 1/200.

It was found that the numerical instabilities at low viscosities were a result of the interpolation of the population

density. All instabilities vanished when the CBB rules were based on Eqs. (10) and (11) instead of Eq. (13). Furthermore, it was found that the instabilities could be suppressed by using integer instead of floating point arithmetic, suggesting that round-off errors are driving the instability. Although in theory the CBB rules conserve mass and momentum, the fact that each population after propagation is a sum of contributions from different postcollision populations [Eq. (24)] makes the CBB rules more sensitive to round-off errors than the underlying LBB rules, where each postcollision population is either completely propagated in the direction of its velocity or completely reflected at a boundary node. It has been suggested [36] that the instabilities are most likely caused by the excitation of staggered invariants, as a result of the interpolation. Furthermore, it was suggested that they could be suppressed by reducing the magnitude of the eigenvalues of the kinetic modes [36].

In order to investigate the influence of the rest-particle mode, the bulk mode, and the kinetic modes on the stability of the lattice-Boltzmann equation, an algorithm was developed using a more general collision operator than the one used in Secs. III A–III C; a linearized collision operator was used with a complete set of eigenvectors, instead of one with eigenvectors corresponding only to the conserved and viscous modes. The lattice-Boltzmann equation [Eq. (3)] can then be written in terms of the moments $m_k(\mathbf{r},t)$, $k=0,\ldots,J-1$, of the velocity distribution function with respect to this basis of eigenvectors (see Appendix A). Suitable sets of eigenvectors are given for both the two-dimensional nine-speed model and the three-dimensional 19-speed model in Appendix A (Tables I and II). The eigenvalues associated with the rest-particle mode, the bulk mode, and the kinetic modes are denoted by λ_0, λ_B, and λ_K. The symmetry of the cubic lattice allows for only two distinct kinetic eigenvalues: λ_{K_3}, associated with the eigenvectors that are third order polynomials in $\mathbf{c}_i$, and λ_{K_4}, associated with the eigenvectors that are fourth order polynomials in $\mathbf{c}_i$ (absent in the nine-speed model).

For low-Reynolds-number flow the lattice-Boltzmann equation can be written as a linear system of equations in terms of the moments $m_k(\mathbf{r},t)$ (see Appendix B), i.e.,

$$\mathbf{M}(t+1)=\mathbf{P}\cdot\mathbf{T}\cdot[\mathbf{E}\cdot\mathbf{M}(t)+\mathbf{F}(t)]. \tag{37}$$

Here, the matrices **P** and **E** are defined in Appendix B, the matrix **T** is defined in Eq. (25), and the vector notation of Eq. (26) is adopted for the moments $m_k(\mathbf{r},t)$ and the external forces $f_i(\mathbf{r},t)$. The stability of Eq. (37) can be analyzed by calculating the eigenvalues of the matrix $\mathbf{A}=\mathbf{P}\cdot\mathbf{T}\cdot\mathbf{E}$. Global stability requires the magnitude of all of these eigenvalues to be less than or equal to 1. Hence, in order to investigate the influence of the rest-particle mode, the bulk mode, and the kinetic modes on the stability of the lattice-Boltzmann equation the eigenvalues of the matrix **A** were calculated for different values of λ_0, λ_B, λ_{K_3}, and λ_{K_4}. Note that local stability of the lattice-Boltzmann equation requires all eigenvalues of the collision operator to be between -2 and 0 [23].

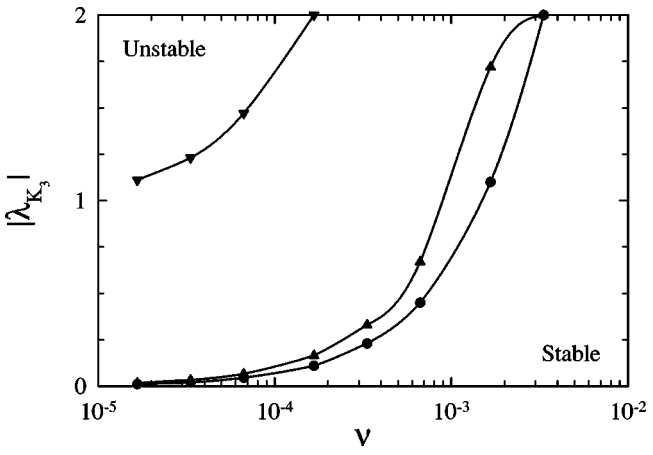

FIG. 10. The stability diagram for a square array of disks. The symbols are for the C9M9 model with $R_0=3.0$, $L_0=8$, and the center of the disk in the corner of a lattice cell (●), and for the C19M19 model with $R_0=1.5$, $L_0=4$, and the center of the disk in the center (▲) or in the corner (▼) of a lattice cell. The symbols indicate the largest magnitude of λ_{K_3} for which the simulations were stable. Hence, the area below the curves indicates the stable region. The line is added as a guide to the eye.

The eigenvalues of the matrix **A** for the general nine-speed model (the C9M9 model) were calculated for a square array of disks at kinematic shear viscosities between $\nu=5/3$ and $\nu=1/60\,000$ and for different positions with respect to the lattice. The systems that were studied were identical to those in Figs. 6(a,b). First, the influence of the eigenvalue associated with the kinetic modes, λ_{K_3}, was studied, while keeping $\lambda_0=\lambda_B=-1$. It was found that for $\lambda_{K_3}=-1$ each system was stable at the lowest resolution ($R_0=1.5$ and $L_0=4$). The smallest system that showed instabilities for $\lambda_{K_3}=-1$ was one with $R_0=3.0$, $L_0=8$, the center of the disk in the corner of a lattice cell, and a kinematic shear viscosity below 1/1500. It was found that decreasing the magnitude of λ_{K_3} increased the numerical stability, and that global stability could always be regained for values of $|\lambda_{K_3}|$ below a certain critical value. As an example, Fig. 10 shows the stability diagram for the system with $R_0=3.0$, $L_0=8$, and the center of the disk in the corner of a lattice cell. It shows that the system was stable for $\nu>1/300$ for any value of λ_{K_3} between 0 and -2. Furthermore, it shows that for $\nu<1/300$ stability could always be regained by reducing the magnitude of λ_{K_3}. Hence, the instabilities in the lattice-Boltzmann equation at low shear viscosities induced by the boundary rules employed here can be suppressed by reducing the magnitude of the eigenvalue of the kinetic modes. Next, the influence of the eigenvalue associated with the rest-particle mode, λ_0, was studied, while keeping $\lambda_B=\lambda_{K_3}=-1$. It was found that each system, including those that were stable for $\lambda_0=-1$, became unstable for values of λ_0 close to 0 or -2. Hence, changing the rest-particle mode eigenvalue to a value different from -1 decreases the global stability of the lattice-Boltzmann equation. Finally, the influence of the eigenvalue associated with the bulk mode, λ_B, was studied, while keep-

ing $\lambda_0=\lambda_{K_3}=-1$. It was found that decreasing the magnitude of λ_B, increased the stability, just as for λ_{K_3}. However, the effect was not sufficient to stabilize those systems that were even moderately unstable. Furthermore, just as for λ_0, even those systems that were stable for $\lambda_B=-1$ became unstable for values of λ_B close to -2, an observation that signifies an important drawback of the ERT model as discussed in more detail below.

The eigenvalues of the matrix **A** for the general 19-speed model (the C19M19 model) were calculated for the same systems as for the C9M9 model, i.e., for cylinders with only one layer in the symmetry (z) direction, since the periodic arrays of spheres were too large to make a study of the eigenvalues feasible. Qualitatively similar results were found for the influence of λ_0, λ_B, and λ_{K_3} on the stability of the lattice-Boltzmann equation. The main difference between the two models was a lower overall stability of the C19M19 model; the system with the center of the cylinder located in the center of a lattice cell was already unstable at the lowest resolution below a critical shear viscosity of about 1/1000 and for $\lambda_0=\lambda_B=\lambda_{K_3}=\lambda_{K_4}=-1$. However, just as for the C9M9 model, stability could always be regained by reducing the magnitude of λ_{K_3} (Fig. 10). The decrease in stability of the C19M19 model for these systems is most probably a result of additional invariants as a result of the symmetry in the z direction. No significant effect of λ_{K_4} on the stability was found, so a value of -1 can be used for all practical purposes.

In addition to improving the stability it was found that reducing the magnitude of λ_{K_3} decreased the drag coefficient of the disks and spheres. Hence, the hydrodynamic radius of a disk or sphere depends on λ_{K_3}, in addition to the shear viscosity, the nominal radius, and the position relative to the lattice. This is consistent with earlier results of Cornubert *et al.* [26] and Ginzbourg and Adler [28], who showed that the position of the zero-velocity plane for two-dimensional channel flow depends on the shear viscosity, the nominal channel width, the angle of the walls with respect to one of the lattice directions, and the kinetic eigenvalue. They showed that for a given shear viscosity one can tune the kinetic eigenvalue, such that the position of the hydrodynamic boundary coincides with the physical boundary. Reducing the magnitude of λ_{K_3} both stabilizes simulations at low viscosities and also decreases the error in the location of the hydrodynamic boundary.

The lower stability of the ERT model (Figs. 8 and 9) at low shear viscosities is a result of the nonhydrodynamic modes. Low shear viscosities are obtained by choosing a relaxation time τ close to 0.5, implying that all the eigenvalues of the collision operator, including of course λ_0, λ_B, and λ_{K_3}, are close to -2, which severely reduces the stability compared to the C9M9 and C19M19 models. It is found that choosing different relaxation rates for the kinetic modes leads to a model that is significantly more stable than the ERT model, in agreement with the conclusion of Lallemand and Luo [31].

V. DISCUSSION

In this paper a detailed description is given of the continuous bounce-back rules [3], a recently developed set of boundary rules for the lattice-Boltzmann model. Although the CBB rules are more complicated to implement, the additional computational overhead is small. The key advance is that the accuracy of these boundary conditions is insensitive to the position of the interface with respect to the lattice. This allows for a reduction in resolution in typical simulations by a factor of 2 or 4, corresponding to at least a 16-fold reduction in computer time and an eightfold reduction in memory. We anticipate that the most important applications of this idea will be to simulations of changes in morphology in porous media, arising from erosion or deposition of solid carried by the fluid [37]. The CBB rules allow for a continuous variation in the position of the solid surfaces, and can be adopted to simulations of particle suspensions by modifying the reflected population densities to take account of the moving interface [4]. In simulations of particle suspensions, particles are discretized by assigning each lattice node to either the fluid or the solid phase, depending on the position of the node with respect to the particle interface. This leads to fluctuations in the particle's volume when it moves over the grid, resulting in fluctuations in the drag force [38]. The CBB rules should prevent these fluctuations, since the particle volume is independent of its position with respect to the grid, resulting in smoother and more accurate particle trajectories. The modification of the CBB rules to extend the method to moving interfaces is in preparation.

Numerical and theoretical analysis of Poiseuille flow in aligned channels with arbitrary widths L_0 showed that the hydrodynamic boundary, i.e., the zero-velocity plane, is located at the actual position of the solid-fluid interface with relative deviations that are of order L_0^{-2}. This leads to asymptotic second order convergence of the rms error in the flow field [Fig. 2(d)]. For more general geometries, like flow in inclined channels or flow through periodic arrays of disks or spheres, the hydrodynamic boundary is generally displaced from the physical one. This displacement varies with shear viscosity and geometry, extending into the fluid for kinematic shear viscosities smaller than about 1/6 and into the solid for values above about 1/6. For kinematic shear viscosities far below 1/6 the hydrodynamic boundary extends significantly into the fluid and the simulation results should be interpreted in terms of the actual hydrodynamic boundary instead of the physical one, determined by the location of the solid-fluid interfaces [4]. For example, the relative error in the drag coefficient of a periodic array of disks or spheres decreases as L_0^{-2} with respect to its value based on the hydrodynamic radius (Fig. 7), but only as L_0^{-1} with respect to its value based on the nominal radius.

Stability analysis of the lattice-Boltzmann equation with the CBB rules indicated that most of the simulations described in this paper were numerically unstable below a certain critical shear viscosity. This poses a potential problem for simulations of nonzero-Reynolds-number flow, when it is often necessary to reduce the shear viscosity in order to keep the Mach number small. For example, in a recent study of

inertial flow in ordered and random arrays of spheres [39] a kinematic shear viscosity of 0.01 was used. This is well within the stability range of all the collision operators except possibly the ERT model.

It was found that the instabilities were a result of the interpolation of the population density in the CBB rules to account for the velocity gradient in the direction of propagation and that the critical shear viscosity below which the instabilities occur depends on the collision operator that is being used. The ERT model [24,25] had the highest critical kinematic shear viscosity; its value for a general geometry lay between 1/60 and 1/200. Both the C9M6 and the C19M10 models [21] were found to be more stable. These models produced stable results for $\nu \geqslant 1/600$ for each of the simulations described in this paper. Furthermore, it was found that the stable region of these models could be increased by using the more general linearized collision operators described in Appendix A. In that case, stability could always be regained by reducing the magnitude of the eigenvalues associated with the kinetic modes (Fig. 10). In addition this also reduces the error in the position of the hydrodynamic boundary. Hence, being able to choose different eigenvalues to separate the relaxation of the hydrodynamic modes and the kinetic modes greatly enhances the stability of the lattice-Boltzmann method, giving the more general linearized collision operators described in Appendix A a significant advantage over the ERT model.

ACKNOWLEDGMENTS

The authors gratefully acknowledge helpful discussions with Dr. Li-Shi Luo (ICASE, NASA Langley Research Center). This work was supported by the U.S. Department of Energy, Office of Basic Energy Sciences (Grant No. DE-FG02-98ER14853) and the National Science Foundation via the Engineering Research Center for Particle Science and Technology at the University of Florida (Grant No. EEO-94-02989).

APPENDIX A: GENERAL LINEARIZED COLLISION OPERATOR

In this appendix a general linearized collision operator for the lattice-Boltzmann equation is constructed for both the two-dimensional nine-speed model (the C9M9 model) and the three-dimensional 19-speed model (the C19M19 model).

First a complete set of pairwise orthogonal vectors $\mathbf{b}_k$, $k=0,\ldots,J-1$, is constructed in the velocity space spanned by $\mathbf{c}_i$, $i=0,\ldots,J-1$. A suitable set of vectors for the C9M9 [27,31] and the C19M19 models is given in Tables I and II.

Next, the linearized collision operator $\mathbf{L}$ is defined such that these basis vectors are its eigenvectors with corresponding eigenvalues λ_k,

$$\mathbf{L}\cdot\mathbf{b}_k=\lambda_k\mathbf{b}_k. \tag{A1}$$

The collision step of the lattice-Boltzmann update [Eq. (7)] with the collision operator as defined in Eq. (4) can then be written in terms of the $\mathbf{b}_k$'s, their associated eigenvalues λ_k, the force density $f_i(\mathbf{r},t)$, and the moments of the velocity distribution function, $n_i(\mathbf{r},t)$, and of the equilibrium velocity distribution function, $n_i^{\mathrm{eq}}(\mathbf{r},t)$, with respect to these basis vectors, i.e.,

$$\mathbf{n}^*(\mathbf{r},t)=\sum_k b_k^{-2}\{m_k(\mathbf{r},t)+\lambda_k[m_k(\mathbf{r},t)-m_k^{\mathrm{eq}}(\mathbf{r},t)]\}\mathbf{b}_k + \mathbf{f}(\mathbf{r},t), \tag{A2}$$

where the vector notation of Eq. (26) is used. Here, the moments $m_k(\mathbf{r},t)$ and $m_k^{\mathrm{eq}}(\mathbf{r},t)$ are defined as the projections of $n_i(\mathbf{r},t)$ and $n_i^{\mathrm{eq}}(\mathbf{r},t)$ on the $\mathbf{b}_k$'s, i.e.,

$$m_k(\mathbf{r},t)=\sum_i b_{ki}n_i(\mathbf{r},t)=\mathbf{b}_k\cdot\mathbf{n}(\mathbf{r},t) \tag{A3}$$

and

TABLE I. Basis vectors, equilibrium moments, and eigenvalues for the C9M9 model [27,31]. Here $\mathbf{b}_0^*=(1,0,\ldots,0)$ and $c_i=\|\mathbf{c}_i\|$.

k	$\mathbf{b}_k$	$m_k^{\mathrm{eq}}(\mathbf{r},t)$	λ_k
0	$9\mathbf{b}_0^*-\mathbf{b}_1+\mathbf{b}_4$	$\rho-3\rho u^2$	λ_0
1	$(1,\ldots,1)$	ρ	
2	$\{c_{ix}\}$	ρu_x	
3	$\{c_{iy}\}$	ρu_y	
4	$\{3c_i^2-4\}$	$3\rho u^2-2\rho$	λ_B
5	$\{c_{ix}^2-c_{iy}^2\}$	$\rho u_x^2-\rho u_y^2$	λ_S
6	$\{c_{ix}c_{iy}\}$	$\rho u_x u_y$	λ_S
7	$\{c_{iy}(3c_i^2-5)\}$	$-\rho u_y$	λ_{K_3}
8	$\{c_{ix}(3c_i^2-5)\}$	$-\rho u_x$	λ_{K_3}

TABLE II. Basis vectors, equilibrium moments, and eigenvalues for the C19M19 model. Here $\mathbf{b}_0^*=(1,0,\ldots,0)$ and $c_i=\|\mathbf{c}_i\|$.

k	$\mathbf{b}_k$	$m_k^{\mathrm{eq}}(\mathbf{r},t)$	λ_k
0	$399\mathbf{b}_0^*-21\mathbf{b}_1+5\mathbf{b}_5$	$\rho-3\rho u^2$	λ_0
1	$(1,\ldots,1)$	ρ	
2	$\{c_{ix}\}$	ρu_x	
3	$\{c_{iy}\}$	ρu_y	
4	$\{c_{iz}\}$	ρu_z	
5	$\{19c_i^2-30\}$	$19\rho u^2-11\rho$	λ_B
6	$\{3c_{ix}^2-c_i^2\}$	$3\rho u_x^2-\rho u^2$	λ_S
7	$\{c_{iy}^2-c_{iz}^2\}$	$\rho u_y^2-\rho u_z^2$	λ_S
8	$\{c_{ix}c_{iy}\}$	$\rho u_x u_y$	λ_S
9	$\{c_{iy}c_{iz}\}$	$\rho u_x u_z$	λ_S
10	$\{c_{iz}c_{ix}\}$	$\rho u_z u_z$	λ_S
11	$\{c_{ix}(5c_i^2-9)\}$	$-2\rho u_x/3$	λ_{K_3}
12	$\{c_{ix}(c_{iy}^2-c_{iz}^2)\}$	0	λ_{K_3}
13	$\{c_{iy}(5c_i^2-9)\}$	$-2\rho u_y/3$	λ_{K_3}
14	$\{c_{iy}(c_{iz}^2-c_{ix}^2)\}$	0	λ_{K_3}
15	$\{c_{iz}(5c_i^2-9)\}$	$-2\rho u_z/3$	λ_{K_3}
16	$\{c_{iz}(c_{ix}^2-c_{iy}^2)\}$	0	λ_{K_3}
17	$\{(c_i^2-3c_{iz}^2)(3c_i^2-5)\}$	$(3\rho u_z^2-\rho u^2)/2$	λ_{K_4}
18	$\{(c_{ix}^2-c_{iy}^2)(3c_i^2-5)\}$	$(-\rho u_x^2+\rho u_y^2)/2$	λ_{K_4}

$$m_k^{\mathrm{eq}}(\mathbf{r},t)=\sum_i b_{ki} n_i^{\mathrm{eq}}(\mathbf{r},t)=\mathbf{b}_k\cdot\mathbf{n}^{\mathrm{eq}}(\mathbf{r},t). \tag{A4}$$

The reciprocal relations

$$\mathbf{n}(\mathbf{r},t)=\sum_k \frac{m_k(\mathbf{r},t)\mathbf{b}_k}{b_k^2} \tag{A5}$$

and

$$\mathbf{n}^{\mathrm{eq}}(\mathbf{r},t)=\sum_k \frac{m_k^{\mathrm{eq}}(\mathbf{r},t)\mathbf{b}_k}{b_k^2} \tag{A6}$$

follow straightforwardly from Eqs. (A3) and (A4), since by construction $\mathbf{b}_i\cdot\mathbf{b}_j=b_i^2\delta_{ij}$, where $b_i=\|\mathbf{b}_i\|$.

Finally, the eigenvalues λ_k and the equilibrium moments $m_k^{\mathrm{eq}}(\mathbf{r},t)$ are defined. Note that some of these moments and eigenvalues are related to physical quantities, but that others have no obvious physical meaning. For the C9M9 model all the equilibrium moments, except m_0^{eq}, are fixed in order to recover the Navier-Stokes equations. A suitable choice of m_0^{eq} is $m_0^{\mathrm{eq}}=\rho-3\rho u^2$; for this particular choice an equilibrium distribution is recovered equal to that introduced by Qian *et al.* [25] and later obtained by systematically discretizing the continuous Boltzmann equation in both time and phase space [40,41], i.e.,

$$n_i^{\mathrm{eq}}=\rho w_i\left(1+3\mathbf{u}\cdot\mathbf{c}_i+\frac{9}{2}\mathbf{uu}:\mathbf{c}_i\mathbf{c}_i-\frac{3}{2}u^2\right), \tag{A7}$$

with the weights w_i in two dimensions (2D) given by

$$w_i=\begin{cases} 4/9, & c_i^2=0 \\ 1/9, & c_i^2=1 \\ 1/36, & c_i^2=2 \end{cases} \quad (2\mathrm{D}). \tag{A8}$$

The eigenvalues associated with the viscous modes are related to the bulk and shear viscosities, i.e.,

$$\eta=\frac{\rho}{6}\left(1+\frac{2}{\lambda_{\mathrm{S}}}\right) \tag{A9}$$

and

$$\eta_{\mathrm{B}}=\frac{\rho}{6}\left(1+\frac{2}{\lambda_{\mathrm{B}}}\right) \quad (2\mathrm{D}). \tag{A10}$$

The equilibrium moments and the eigenvalues for the C9M9 model are summarized in Table I.

For the C19M19 model a similar analysis gives the equilibrium moments and the eigenvalues that are summarized in Table II. The equilibrium moments $m_1^{\mathrm{eq}},\ldots,m_{16}^{\mathrm{eq}}$ are determined by the form of the Navier-Stokes equations. The remaining ones, m_0^{eq}, m_{17}^{eq}, and m_{18}^{eq}, can again be chosen to obtain the equilibrium distribution of Eq. (A7). In this case the weights are given by

$$w_i=\begin{cases} 1/3, & c_i^2=0 \\ 1/18, & c_i^2=1 \\ 1/36, & c_i^2=2 \end{cases} \quad (3\mathrm{D}). \tag{A11}$$

The eigenvalues λ_{B} and λ_{S} are again related to the bulk and shear viscosities. While the expression for the shear viscosity is the same as for the C9M9 model [cf. Eq. (A9)], the corresponding expression for the bulk viscosity is

$$\eta_{\mathrm{B}}=\frac{\rho}{9}\left(1+\frac{2}{\lambda_{\mathrm{B}}}\right) \quad (3\mathrm{D}). \tag{A12}$$

APPENDIX B: MATRIX FORMULATION FOR LOW-REYNOLDS-NUMBER FLOW

In order to investigate the stability of the lattice-Boltzmann equation a low-Reynolds-number approximation was used to write the lattice-Boltzmann equation as a linear system of equations in terms of the moments with respect to the eigenvectors of the general linearized collision operator (Appendix A). The global stability of these equations can then be determined by calculating its eigenvalues.

The collision step of the lattice-Boltzmann update was already obtained in Appendix A [Eq. (A2)]:

$$\mathbf{n}^*(\mathbf{r},t)=\sum_k b_k^{-2}\{m_k(\mathbf{r},t)+\lambda_k[m_k(\mathbf{r},t)-m_k^{\mathrm{eq}}(\mathbf{r},t)]\}\mathbf{b}_k + \mathbf{f}(\mathbf{r},t). \tag{B1}$$

Here $m_k(\mathbf{r},t)$ and $m_k^{\mathrm{eq}}(\mathbf{r},t)$ are the moments of $n_i(\mathbf{r},t)$ and $n_i^{\mathrm{eq}}(\mathbf{r},t)$ with respect to the eigenvectors $\mathbf{b}_k$, $k=0,\ldots,J-1$ [cf. Eqs. (A3) and (A4)], $b_k=\|\mathbf{b}_k\|$, and λ_k is the eigenvalue associated with the eigenvector $\mathbf{b}_k$.

For low-Reynolds-number flows, the equilibrium velocity distribution function $n_i^{\mathrm{eq}}(\mathbf{r},t)$ is required only to linear order in the fluid velocity and the equilibrium moments $m_k^{\mathrm{eq}}(\mathbf{r},t)$ can be written as a linear combination of the conserved quantities $\rho(\mathbf{r},t)$ and $\mathbf{j}(\mathbf{r},t)$ [cf. Eqs. (A4) and (A7)]. The collision step of the lattice-Boltzmann update [Eq. (B1)] can then be written as a linear system of equations,

$$\mathbf{n}^*(\mathbf{r},t)=\mathbf{e}\cdot\mathbf{m}(\mathbf{r},t)+\mathbf{f}(\mathbf{r},t), \tag{B2}$$

$$\mathbf{N}^*(t)=\mathbf{E}\cdot\mathbf{M}(t)+\mathbf{F}(t). \tag{B3}$$

Here, the matrix $\mathbf{E}$ is a $JN\times JN$ block-diagonal matrix, $\mathbf{E}=\mathrm{diag}\{\mathbf{e},\ldots,\mathbf{e}\}$, and $\mathbf{e}$ is the $J\times J$ expansion matrix, defined by its elements

$$e_{ij}=\frac{b_{ij}}{b_j^2}(1+\lambda_j)-\sum_k \frac{b_{ik}}{b_k^2}\lambda_k g_{kj}, \tag{B4}$$

where b_{ij} is the ith element of $\mathbf{b}_j$ and the matrix $\mathbf{g}$ projects $m_k^{\mathrm{eq}}(\mathbf{r},t)$ onto $\mathbf{m}(\mathbf{r},t)$, i.e., $m_k^{\mathrm{eq}}(\mathbf{r},t)=\Sigma_j g_{kj} m_j(\mathbf{r},t)$.

The propagation step was already given in Eq. (25); hence,

$$\mathbf{N}(t+1)=\mathbf{T}\cdot[\mathbf{E}\cdot\mathbf{M}(t)+\mathbf{F}(t)]. \tag{B5}$$

Finally, the velocity distribution after propagation is projected back to its moments,

$$\mathbf{m}(\mathbf{r},t+1)=\mathbf{p}\cdot\mathbf{n}(\mathbf{r},t+1),$$

$$\mathbf{M}(t+1)=\mathbf{P}\cdot\mathbf{N}(t+1), \tag{B6}$$

where the matrix $\mathbf{P}$ is a $JN\times JN$ block-diagonal matrix, $\mathbf{P}=\mathrm{diag}\{\mathbf{p},\ldots,\mathbf{p}\}$, and $\mathbf{p}$ is the $J\times J$ projection matrix, which is defined such that its kth row equals the eigenvector $\mathbf{b}_k$ [cf. Eq. (A3)]. From Eqs. (B5) and (B6) we obtain

$$\mathbf{M}(t+1)=\mathbf{P}\cdot\mathbf{T}\cdot[\mathbf{E}\cdot\mathbf{M}(t)+\mathbf{F}(t)], \tag{B7}$$

which completes the formulation of the lattice-Boltzmann equation as a linear system of equations in terms of the moments of the velocity distribution function with respect to the eigenvectors of the collision operator.

A similar approach was used in Ref. [42] to directly solve the steady-state solution of the lattice-Boltzmann equation for low-Reynolds-number flow in porous media. Here, the eigenvalues associated with the rest-particle mode and those associated with the kinetic modes were set to -1, projecting them out entirely from the postcollision distribution. Hence, this Appendix gives a straightforward extension of the results in Ref. [42] to a general linearized collision operator.

[1] S. Chen and G. D. Doolen, Annu. Rev. Fluid Mech. **30**, 329 (1998).
[2] J. A. Kaandorp, C. P. Lowe, D. Frenkel, and P. M. A. Sloot, Phys. Rev. Lett. **77**, 2328 (1996).
[3] R. Verberg and A. J. C. Ladd, Phys. Rev. Lett. **84**, 2148 (2000).
[4] A. J. C. Ladd, J. Fluid Mech. **271**, 285 (1994).
[5] X. He, Q. Zou, L.-S. Luo, and M. Dembo, J. Stat. Phys. **87**, 115 (1997).
[6] P. A. Skordos, Phys. Rev. E **48**, 4823 (1993).
[7] D. R. Noble, S. Chen, J. G. Georgiadis, and R. O. Buckius, Phys. Fluids **7**, 203 (1995).
[8] T. Inamuro, M. Yoshino, and F. Ogino, Phys. Fluids **7**, 2928 (1995).
[9] T. Inamuro, M. Yoshino, and F. Ogino, Phys. Fluids **8**, 1124 (1996).
[10] R. S. Maier, R. S. Bernard, and D. W. Grunau, Phys. Fluids **8**, 1788 (1996).
[11] S. Chen, D. Martinez, and R. Mei, Phys. Fluids **8**, 2527 (1996).
[12] I. Ginzbourg and D. d'Humières, J. Stat. Phys. **84**, 927 (1996).
[13] Q. Zou and X. He, Phys. Fluids **9**, 1591 (1997).
[14] O. Filippova and D. Hänel, J. Comput. Phys. **147**, 219 (1998).
[15] R. Mei, L.-S. Luo, and W. Shyy, J. Comput. Phys. **155**, 307 (1999).
[16] D. R. Noble and J. R. Torczynski, Int. J. Mod. Phys. C **9**, 1189 (1998).
[17] H. Chen, C. Teixeira, and K. Molvig, Int. J. Mod. Phys. C **9**, 1281 (1998).
[18] H. Chen, Phys. Rev. E **58**, 3955 (1998).
[19] H. Xi, G. Peng, and S.-H. Chou, Phys. Rev. E **60**, 3380 (1999).
[20] R. Benzi, S. Succi, and M. Vergassola, Phys. Rep. **222**, 145 (1992).
[21] A. J. C. Ladd and R. Verberg, J. Stat. Phys. **104**, 1191 (2001).
[22] U. Frisch, D. d'Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, and J.-P. Rivert, Complex Syst. **1**, 649 (1987).
[23] F. J. Higuera, S. Succi, and R. Benzi, Europhys. Lett. **9**, 345 (1989).
[24] S. Chen, H. Chen, D. Martinez, and W. Matthaeus, Phys. Rev. Lett. **67**, 3776 (1991).
[25] Y. H. Qian, D. d'Humières, and P. Lallemand, Europhys. Lett. **17**, 479 (1992).
[26] R. Cornubert, D. d'Humières, and D. Levermore, Physica D **47**, 241 (1991).
[27] D. d'Humières, in *Rarefied Gas Dynamics: Theory and Simulations*, edited by B. D. Shizgal and D. P. Weaver (AIAA, Washington, D.C., 1992), pp. 450–458.
[28] I. Ginzbourg and P. M. Adler, J. Phys. II **4**, 191 (1994).
[29] C. Appert and D. d'Humières, Phys. Rev. E **51**, 4335 (1995).
[30] D. d'Humières, M. Bouzidi, and P. Lallemand, Phys. Rev. E **63**, 066702 (2001).
[31] P. Lallemand and L.-S. Luo, J. Comput. Phys. **61**, 6546 (2000).
[32] G. McNamara and B. Alder, Physica A **194**, 218 (1993).
[33] A. J. C. Ladd, J. Fluid Mech. **271**, 285 (1994).
[34] A. S. Sangani and A. Acrivos, Int. J. Multiphase Flow **8**, 193 (1982).
[35] A. A. Zick and G. M. Homsy, J. Fluid Mech. **115**, 13 (1982).
[36] L.-S. Luo (private communication).
[37] R. Verberg and A. J. C. Ladd (unpublished).
[38] A. J. C. Ladd, Phys. Fluids **9**, 491 (1997).
[39] R. J. Hill, D. L. Koch, and A. J. C. Ladd (unpublished).
[40] X. He and L.-S. Luo, Phys. Rev. E **55**, R6333 (1997).
[41] X. He and L.-S. Luo, Phys. Rev. E **56**, 6811 (1997).
[42] R. Verberg and A. J. C. Ladd, Phys. Rev. E **60**, 3366 (1999).
[43] In our previous work the force density was incorrectly multiplied by the fluid fraction of the cell. This leads to lower velocities in the partially filled cells, and is in better agreement with analytic results for narrow channels ($L<5$). Nevertheless, the force density is not dependent on the fluid fraction and this mistake has now been corrected.