Simulation of low-Reynolds-number flow via a time-independent lattice-Boltzmann method

R. Verberg and A. J. C. Ladd
Chemical Engineering Department, University of Florida, Gainesville, Florida 32611-6005

(Received 14 April 1999)

We present a numerical method to solve the equations for low-Reynolds-number (Stokes) flow in porous media. The method is based on the lattice-Boltzmann approach, but utilizes a direct solution of time-independent equations, rather than the usual temporal evolution to steady state. Its computational efficiency is 1–2 orders of magnitude greater than the conventional lattice-Boltzmann method. The convergence of the permeability of random arrays of spheres has been analyzed as a function of mesh resolution at several different porosities. For sufficiently large spheres, we have found that the convergence is quadratic in the mesh resolution. [S1063-651X(99)05309-X]

PACS number(s): 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]. Initially developed by McNamara and Zanetti [2] as a means to eliminate the statistical noise inherent in a lattice-gas automaton (LGA) [3] and developed into a practical simulation tool by Higuera *et al.* [4,5], it has recently been derived directly from the Boltzmann equation by discretization in time and phase space [6–8]. In contrast with the LGA, the lattice-Boltzmann model is Galilean invariant and has a velocity-independent equation of state [9,10]. It retains the capability, inherited from LGA, to incorporate complex boundary surfaces in a straightforward and reliable manner [11–14]. The lattice-Boltzmann method is at least comparable in speed and accuracy with competing methods [15–18].

Lattice-Boltzmann simulations have been applied to simulate fluid flow in different porous structures, most recently a packed bed of fibers [19]. However, the calculations are limited by the temporal evolution of the flow fields to steady state. The inherent slowness of diffusive momentum transport, combined with the irregularity of the geometrical structures, often require simulations of 10^4-10^5 cycles to determine the steady-state flow field. In addition, the slowness of convergence often makes if difficult to decide when the system has actually reached a steady state. It is possible to artificially accelerate the convergence of the permeability, but the reduction in the number of cycles is typically only a factor of 2 or 3 [14]. In this paper we present a more radical approach, which involves a direct solution of the time-independent equations. The key idea is to solve for the steady-state mass and momentum densities directly, rather than allow them to evolve diffusively in time; this requires a reformulation of the normal lattice-Boltzmann algorithm.

In Sec. II we outline the essential ingredients of the lattice-Boltzmann method and in Sec. III we present a method to compute the stationary solution of a lattice-Boltzmann model. In Secs. IV and V we compare the results of the method with the standard time-dependent lattice-Boltzmann method in two and three dimensions. We analyze the rate of convergence to steady state as a function of the porosity and lattice size. In Sec. V we also present precise results for the permeability of random arrays of spheres as a function of mesh resolution and porosity. We end with a short discussion.

II. LATTICE BOLTZMANN METHOD

In this section we summarize the theoretical background to the lattice-Boltzmann method; more extensive discussions can be found in the literature [1,2,4,9,20]. The state of the system is characterized by the discretized one-particle velocity distribution function $n_i(\mathbf{r},t)$, which describes the number of fluid particles at a lattice node $\mathbf{r}$ at time t with a velocity $\mathbf{c}_i$. Here $\mathbf{r}$, t, and $\mathbf{c}_i$ are discrete, whereas $n_i(\mathbf{r},t)$ itself is a continuous variable. The mass density $\rho(\mathbf{r},t)$ and the momentum density $\mathbf{j}(\mathbf{r},t)$ are defined by moments of this velocity distribution function

$$\rho(\mathbf{r},t)=\sum_{i=1}^{J} n_i(\mathbf{r},t), \tag{1}$$

$$\mathbf{j}(\mathbf{r},t)=\sum_{i=1}^{J} \mathbf{c}_i n_i(\mathbf{r},t). \tag{2}$$

Here $\mathbf{j}(\mathbf{r},t)=\rho(\mathbf{r},t)\mathbf{u}(\mathbf{r},t)$, with $\mathbf{u}(\mathbf{r},t)$ the macroscopic fluid velocity; the summation runs over the complete set of velocities $\{\mathbf{c}_i\}$. For the two-dimensional lattice model $J=9$, and for the three-dimensional model $J=18$.

The time evolution of the velocity distribution function in the presence of an external force (e.g., an externally applied pressure gradient or gravitational field) is governed by the discretized Boltzmann equation

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

where $\Delta_i(\mathbf{r},t)$ 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. Each lattice-Boltzmann update consists of two steps. The first step includes the effects of collisions and external forces: for later use we define the velocity distribution function after this step as $n_i^*(\mathbf{r},t)$,

$$n_i^*(\mathbf{r},t)=n_i(\mathbf{r},t)+\Delta_i(\mathbf{r},t)+f_i(\mathbf{r},t). \tag{4}$$

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

$$n_i(\mathbf{r}+\mathbf{c}_i,t+1)=n_i^*(\mathbf{r},t). \tag{5}$$

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

For low-Reynolds-number flows, the equilibrium distribution function $n_i^{\mathrm{eq}}(\mathbf{r})$ is only required to linear order in the fluid velocity [20]; i.e.,

$$n_i^{\mathrm{eq}}(\mathbf{r})=\rho(\mathbf{r})[a_0^i+a_1^i\mathbf{c}_i\cdot\mathbf{u}(\mathbf{r})]. \tag{6}$$

The coefficients a_0^i and a_1^i are determined by the conditions that the shear viscosity is isotropic and that the velocity moments of $n_i^{\mathrm{eq}}(\mathbf{r})$ reproduce the correct hydrodynamic fields for low-Reynolds-number flows: i.e., $\Sigma_i n_i^{\mathrm{eq}}(\mathbf{r})=\rho(\mathbf{r})$, $\Sigma_i\mathbf{c}_i n_i^{\mathrm{eq}}(\mathbf{r})=\mathbf{j}(\mathbf{r})$, and $\Sigma_i\mathbf{c}_i\mathbf{c}_i n_i^{\mathrm{eq}}(\mathbf{r})=c_s^2\rho(\mathbf{r})\mathbf{1}$, where $c_s=\sqrt{1/2}$ is the speed of sound and $\mathbf{1}$ is the unit tensor. Numerical values for the coefficients a_0^i and a_1^i, in two and three dimensions, are given below.

The collision operator $\Delta_i(\mathbf{r},t)$ can be simplified by linearizing about the equilibrium distribution function [4,20]. It takes a particularly simple form when the kinematic viscosity $\nu=1/6$ [20]

$$\Delta_i(\mathbf{r},t)=-[n_i(\mathbf{r},t)-n_i^{\mathrm{eq}}(\mathbf{r})], \tag{7}$$

which is equivalent to the exponential relaxation time model [9,21] with a relaxation time $\tau=1$. This choice of collision operator is especially suitable for low-Reynolds-number flows, since the viscosity is not an important parameter in this case.

We incorporate an external pressure gradient $\boldsymbol{\nabla}p^{\mathrm{ext}}$ by applying a uniform force density $\mathbf{g}=-\boldsymbol{\nabla}p^{\mathrm{ext}}$ to the fluid, i.e.,

$$f_i=a_1^i\mathbf{c}_i\cdot\mathbf{g}. \tag{8}$$

Thus a constant increment of momentum $\Delta\mathbf{j}=\mathbf{g}$ is added to each node at each time step. Using Eqs. (6)–(8), we write the lattice-Boltzmann equation [Eq. (3)] for low-Reynolds-number flow, in the presence of a uniform force density, as

$$n_i(\mathbf{r}+\mathbf{c}_i,t+1)=\sum_{j=1}^{J}(a_0^i+a_1^i\mathbf{c}_i\cdot\mathbf{c}_j)n_j(\mathbf{r},t)+a_1^i\mathbf{c}_i\cdot\mathbf{g}. \tag{9}$$

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. Fixed solid objects were introduced into lattice-gas models by replacing the normal collision rules at a specified set of nodes by the ''bounce-back'' collision rule [22], in which incoming particles are reflected back towards the nodes they came from. Detailed analysis of the bounce-back rule for two-dimensional Poiseuille flow has shown that the location of the zero-velocity plane is shifted from the location of the boundary nodes, into the fluid, by an amount $0.5+\alpha$ [23–25], with α close to zero for kinematic viscosities near 1/6. If the hydrodynamic boundary is assumed to lie along the boundary nodes, the convergence of the flow field is only linear in the mesh resolution. Approximate second-order convergence can be achieved by assuming that the hydrodynamic boundary is displaced by half a lattice spacing from the physical one, and then choosing a kinematic viscosity close to 1/6, so that α is essentially zero [24]. However, this method is still only first order for arbitrary orientations of the boundary plane [25]. Several attempts to obtain more accurate boundary conditions have been suggested in the literature [26–31], but they are not easily applied to irregular geometries.

A fundamental improvement to the nodal bounce-back rule is a bounce-back mechanism where the boundary nodes lie along the links, midway between the solid and fluid nodes [32]. The key difference is that a particle at a node adjacent to the solid surface hits the surface and returns in one time step, whereas it takes two time steps to return when the boundary nodes are located at the lattice nodes. In this case it can be shown that the hydrodynamic boundary is located at the boundary nodes, i.e., midway between the solid and fluid nodes, with deviations that are second order in the mesh resolution [33,34]. In contrast with the second-order schemes cited above, this method is applicable to surfaces of arbitrary shape, without additional complications [13].

In the presence of a uniform force density, the calculation of the flow field is somewhat ambiguous, owing to the discreteness of the lattice-Boltzmann update [34]. Although in most published work the momentum density is measured before the application of the force density, it could equally well be measured afterwards. Since the momentum density before and after forcing are related, $\mathbf{j}_+(\mathbf{r},t)=\mathbf{j}_-(\mathbf{r},t)+\mathbf{g}$, the value of $\mathbf{j}(\mathbf{r},t)$ depends on the point in the update cycle where it is measured. To decide on the correct definition of the momentum density, drag coefficients obtained by driving the fluid flow with a uniform force density have been compared with those obtained using a constant velocity boundary condition [13]. The results agreed exactly if the mean of the velocity field before and after forcing, i.e., $[\mathbf{j}_-(\mathbf{r},t)+\mathbf{j}_+(\mathbf{r},t)]/2$, was used. Therefore, we calculate the momentum flow fields after half the force density is applied, which is equivalent.

Given a discretized model of a porous structure, defined by the boundary nodes at the solid-fluid interfaces, Eq. (9) can be successively iterated in time to determine the steady-state momentum flow field. It can be shown that this long-time limit, with bounce-back conditions at the solid boundaries, approximates the creeping flow or Stokes equations in the presence of an externally applied pressure gradient, $\boldsymbol{\nabla}p^{\mathrm{ext}}=-\mathbf{g}$, with stick boundary conditions at the solid walls. In the next section we show that the steady-state solution can also be obtained directly, by solving a set of linear equations using a conjugate-gradient technique.

III. MATRIX FORMULATION

Although it may seem that the stationary solution of equation Eq. (9) can be found by setting $n_i(\mathbf{r}+\mathbf{c}_i,t+1)=n_i(\mathbf{r},t)$, a more careful analysis reveals that most of the equations are linearly dependent. In fact, due to mass and momentum conservation during collisions, the number of constraints for each fluid node is equal to $D+1$, with D being the spatial dimension of the lattice. Hence, the total number of constraints for the entire lattice is $N_f(D+1)$, with

N_f being the number of fluid nodes. Since the number of equations is JN_f, and J is greater than $D+1$ for all relevant velocity models, no unique stationary solution of Eq. (9) can be found in this way. The equations must be supplemented by the equilibrium distribution function [Eq. (6)] relating the n_i's to the $D+1$ conserved quantities $\rho(\mathbf{r},t)$ and $\mathbf{j}(\mathbf{r},t)$. In fact, it is much more efficient to solve directly for $\rho(\mathbf{r},t)$ and $\mathbf{j}(\mathbf{r},t)$.

Following this reasoning we write the lattice Boltzmann equation [Eq. (9)] in terms of the mass and momentum densities, $\rho(\mathbf{r},t)$ and $\mathbf{j}(\mathbf{r},t)$, using Eqs. (1), (2), and (6). A vector notation is used, in which lower case symbols represent properties of the individual nodes and upper case symbols represent properties of the entire lattice. The velocity distribution function $n_i(\mathbf{r},t)$ is then written as

$$\mathbf{n}(\mathbf{r},t)=\{n_1(\mathbf{r},t),\ldots,n_J(\mathbf{r},t)\};\quad \mathbf{r}=\mathbf{r}_1,\ldots,\mathbf{r}_{N_f},$$
$$\mathbf{N}(t)=\{\mathbf{n}(\mathbf{r}_1,t),\ldots,\mathbf{n}(\mathbf{r}_{N_f},t)\}. \tag{10}$$

A similar notation is used for the post-collision velocity distribution function $n_i^*(\mathbf{r},t)$ and $\mathbf{N}^*(t)$. Since the force density is uniform,

$$\mathbf{f}=\{f_1,\ldots,f_J\},\quad \mathbf{F}=\{\mathbf{f},\ldots,\mathbf{f}\}. \tag{11}$$

Finally, $\mathbf{M}(t)$, the vector of conserved quantities is defined by

$$\mathbf{m}(\mathbf{r},t)=\{\rho(\mathbf{r},t),\mathbf{j}(\mathbf{r},t)\};\quad \mathbf{r}=\mathbf{r}_1,\ldots,\mathbf{r}_{N_f},$$
$$\mathbf{M}(t)=\{\mathbf{m}(\mathbf{r}_1,t),\ldots,\mathbf{m}(\mathbf{r}_{N_f},t)\}. \tag{12}$$

The dimension of the vectors $\mathbf{n}$, $\mathbf{n}^*$, and $\mathbf{f}$ is J, while that of the corresponding vectors $\mathbf{N}$, $\mathbf{N}^*$, and $\mathbf{F}$ is JN_f. The dimensions of $\mathbf{m}$ and $\mathbf{M}$ are $D+1$ and $(D+1)N_f$, respectively.

We proceed by writing the lattice-Boltzmann update in terms of the vector of conserved quantities $\mathbf{M}(t)$. First, we write the collision step in vector notation,

$$\mathbf{n}^*(\mathbf{r},t)=\mathbf{e}\cdot\mathbf{m}(\mathbf{r},t)+\mathbf{f},$$
$$\mathbf{N}^*(t)=\mathbf{E}\cdot\mathbf{M}(t)+\mathbf{F}. \tag{13}$$

The matrix $\mathbf{E}$ is a $JN_f\times(D+1)N_f$ block-diagonal matrix, $\mathbf{E}=\mathrm{diag}\{\mathbf{e},\ldots,\mathbf{e}\}$, of $J\times(D+1)$ expansion matrices $\mathbf{e}$, which are defined such that the ith row of $\mathbf{e}$ is given by [c.f., Eq. (6)]

$$\mathbf{e}_i=\{a_0^i,a_1^i\mathbf{c}_i\}. \tag{14}$$

Note that the right-hand side of Eq. (13) is already written in terms of the conserved quantities.

The propagation step is incorporated by writing $\mathbf{N}(t+1)$ in terms of $\mathbf{N}^*(t)$. This requires shifting the elements of $\mathbf{N}^*(t)$ in accordance with the propagation of the velocity distribution functions, $n_i(\mathbf{r},t)$, to the neighboring nodes $\mathbf{r}+\mathbf{c}_i$. The propagation step can be written as a simple matrix multiplication

$$\mathbf{N}(t+1)=\mathbf{T}\cdot\mathbf{N}^*(t), \tag{15}$$

where $\mathbf{T}$ is the $JN_f\times JN_f$ transition matrix, with $T_{ij}=1$ for valid transitions and $T_{ij}=0$ elsewhere. In the case of normal propagation (i.e., if $\mathbf{r}+\mathbf{c}_i$ is a fluid node), the only nonzero elements of $\mathbf{T}$ have column indexes corresponding to node $\mathbf{r}$ and velocity $\mathbf{c}_i$ paired with row indexes corresponding to node $\mathbf{r}+\mathbf{c}_i$ and velocity $\mathbf{c}_i$. If $\mathbf{r}+\mathbf{c}_i$ is a solid node, the midway bounce-back rule is applied. In that case the only nonzero element in the column corresponding to node $\mathbf{r}$ and velocity $\mathbf{c}_i$ is the row corresponding to node $\mathbf{r}$ and velocity $\mathbf{c}_{i'}$, where $\mathbf{c}_{i'}$ is in the opposite direction to $\mathbf{c}_i$.

Finally, we project the velocity distribution functions after propagation onto the conserved quantities,

$$\mathbf{m}(\mathbf{r},t+1)=\mathbf{p}\cdot\mathbf{n}(\mathbf{r},t+1),$$
$$\mathbf{M}(t+1)=\mathbf{P}\cdot\mathbf{N}(t+1), \tag{16}$$

with the $(D+1)N_f\times JN_f$ block-diagonal matrix $\mathbf{P}=\mathrm{diag}\{\mathbf{p},\ldots,\mathbf{p}\}$ and the $(D+1)\times J$ projection matrix $\mathbf{p}$ defined such that its jth column is given by [c.f., Eqs. (1) and (2)]

$$\mathbf{p}_j=\{1,\mathbf{c}_j\}^T. \tag{17}$$

Using Eqs. (13), (15), and (16), we obtain

$$\mathbf{M}(t+1)=\mathbf{P}\cdot\mathbf{T}\cdot(\mathbf{E}\cdot\mathbf{M}(t)+\mathbf{F}), \tag{18}$$

which completes the formulation of the lattice-Boltzmann equation as a set of linear equations in terms of the vector of conserved quantities $\mathbf{M}$. We note that Eq. (18) is equivalent to Eq. (9), coupled with the midway bounce-back rule at the boundary nodes, but there is no apparent computational advantage in this formulation of the time-dependent problem. Equation (18) is equivalent to a set of finite-difference equations with a particular set of stencils that are based on the lattice-Boltzmann update. For comparison with other finite-difference methods we give example stencils for propagation and bounce-back in Table I.

At steady state $\mathbf{M}(t+1)=\mathbf{M}(t)$, reducing Eq. (18) to the linear system of equations

$$\mathbf{A}\cdot\mathbf{M}=\mathbf{B}, \tag{19}$$

with

$$\mathbf{A}=\mathbf{1}-\mathbf{P}\cdot\mathbf{T}\cdot\mathbf{E} \tag{20}$$

and

$$\mathbf{B}=\mathbf{P}\cdot\mathbf{T}\cdot\mathbf{F}. \tag{21}$$

Despite appearances, Eq. (19) cannot be solved in its present form. In going from the time-evolution equation [Eq. (18)] to the time-independent equation [Eq. (19)], we have neglected the fact that the steady-state solution depends on the initial mass density, $\rho(\mathbf{r},t=0)$, as well as on the constraints imposed by mass and momentum conservation. The steady-state momentum density $\mathbf{j}(\mathbf{r})$ is completely determined by the boundary conditions and force density, but since only $\nabla\rho$ couples to the momentum flux, the steady-state mass density $\rho(\mathbf{r})$ is undetermined to within an additive constant. This implies that one can always express the density at a certain fluid node in terms of the densities at all the

TABLE I. Stencils for propagation and bounce-back in the (1,0,0) and (1,1,0) directions, based on the lattice-Boltzmann update. The updated vector of conserved quantities at a particular node *i* is a sum of contributions from neighboring fluid nodes *j*. Each contribution is obtained by a matrix multiplication of the appropriate propagation stencil (normalized by dividing by 24) with the vector of conserved quantities at the neighboring node *j*. In the case of a solid neighboring node, the bounce-back stencil is used, together with the conserved quantities at node *i*. All other stencils can be obtained straightforwardly by symmetry operations.

Propagation		Bounce-back	
(1,0,0)	(1,1,0)	(1,0,0)	(1,1,0)
$\begin{pmatrix} 2 & 4 & 0 & 0 \\ 2 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 2 & 2 & 0 \\ 1 & 2 & 2 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -4 & 0 & 0 \\ 2 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & -2 & -2 & 0 \\ 1 & -2 & -2 & 0 \\ 1 & -2 & -2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

other nodes, and consequently Eqs. (19) still have one linearly dependent row. In order to obtain a unique solution that satisfies both the boundary conditions and the initial conditions, we replaced the first equation of the system [Eq. (19)] with an equation that sets the total fluid mass to a constant. We have verified that the steady-state solution of the momentum density is independent of the value of the total fluid mass, which we set to zero for convenience. The resulting linear problem is regular, but with a nonsymmetric coefficient matrix **A**, and can be solved using the biconjugate gradient algorithm [35].

If the sample contains disconnected pores, the total fluid mass in each region must be specified independently. It is simplest to treat these disconnected clusters, common in porous media, as individual systems. We used a cluster analysis program to separate the different clusters and determined the flow in each cluster individually. The volumetric flow rates through all the percolating clusters were added together to obtain the total volumetric flow, while nonpercolating clusters were treated as solid since they do not contribute to the permeability.

A final complication arises from the fact that the mass and momentum densities do not uniquely determine the velocity distribution function. Therefore, applying Eq. (6) to the steady-state mass and momentum densities does not lead to the steady-state velocity distribution function. To obtain the correct velocity distribution function from the solution vector **M** [Eq. (19)], a complete lattice-Boltzmann cycle [Eqs. (13) and (15)] is necessary. Only one cycle is required because every velocity distribution function with the correct steady-state mass and momentum densities relaxes to the steady-state velocity distribution function in one time step [Eq. (7)]. The components of the resulting vector **N** are equivalent to those obtained from the long-time limit of Eq. (9). This has also been verified numerically.

IV. TWO-DIMENSIONAL FLOWS

We have simulated fluid flow in periodic and random arrays of disks. All simulations used a square $L \times L$ lattice with periodic boundary conditions in both directions. The total number of lattice nodes occupied by fluid is N_f, giving a porosity $\phi = N_f/N$, where $N = L^2$. The external pressure gradient is applied in the y direction, such that $\nabla p^{\rm ext} = -\hat{\mathbf{e}}_y$; i.e., the uniform force density $\mathbf{g} = \hat{\mathbf{e}}_y$. We used a nine-speed model comprising rest particles and particles with velocities in the {10} and the {11} directions [20]; it is isotropic and Galilean invariant. The coefficients for the rest particles, a_0^0 and a_1^0, are 1/6 and 1/3, respectively, and the coefficients for the remaining velocities, a_0^i and a_1^i, are equal to 1/6 and 1/3 for the {10} directions and 1/24 and 1/12 for the {11} directions.

To check the new algorithm, which we refer to as the matrix method, we first simulated a square array of disks at two porosities, $\phi = 0.6$ and $\phi = 0.9$, and three mesh resolutions, for comparison with Ref. [13]. The volumetric flow rate $\langle \mathbf{j} \rangle$ is defined as the volume-averaged momentum density

$$\langle \mathbf{j} \rangle = \frac{1}{2} \phi \mathbf{g} + \frac{1}{N} \sum_{N_f} \mathbf{j}(\mathbf{r}), \tag{22}$$

and the total drag force $\mathbf{F}^D$ is given by $\mathbf{F}^D = N\mathbf{g} = N\hat{\mathbf{e}}_y$. In Eq. (22) we have accumulated half the applied force density at each fluid node, as discussed in Sec. II. The numerical results for the reduced drag force, $\mathbf{F}^D/\nu\langle \mathbf{j} \rangle$, obtained with the matrix method are identical to those in Table IV of Ref. [13].

In Fig. 1 we show the relative error,

$$\varepsilon_c = \frac{|\langle \mathbf{j} \rangle - \langle \mathbf{j} \rangle_{\rm convg}|}{|\langle \mathbf{j} \rangle_{\rm convg}|}, \tag{23}$$

as a function of the number of iterations, for the matrix method $I_{\rm mat}$, and the conventional time-dependent lattice-Boltzmann method $I_{\rm lbe}$. The errors are measured relative to the fully converged volumetric flow rate, $\langle \mathbf{j} \rangle_{\rm convg}$. Results were obtained for radius $R = 7.5$ disks at porosities $\phi = 0.6$ ($L = 21$) and $\phi = 0.9$ ($L = 42$). For the high porosity sample the matrix method converges approximately 30 times faster than the time-dependent method, whereas at low porosity the gain is less, approximately a factor 5.

In Fig. 2 we show the number of iterations required to reduce the relative error ε_c, to less then 10^{-6} as a function of the linear dimension L. The slope of the lower solid line (matrix) is 1 and that of the upper solid line (lbe) is 2. The vertical bars span a relative error between 10^{-4} (lowest point) and 10^{-8} (highest point). The results indicate that the time to convergence increases approximately linearly with

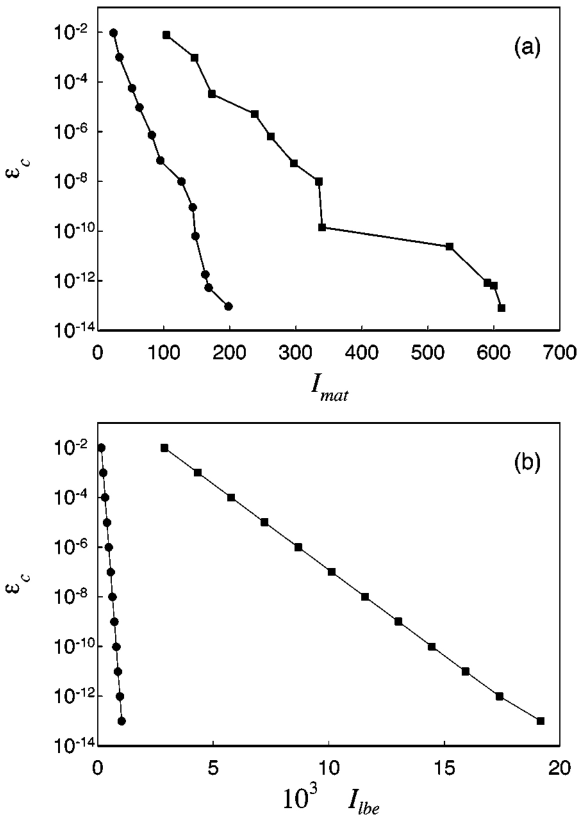

FIG. 1. Relative error, ε_c, as a function of the number of iterations, for the matrix method, I_{mat} (a) and the time-dependent method I_{lbe} (b). The system is a square array of disks with radius $R=7.5$ and porosity $\phi=0.6$ (circles) and $\phi=0.9$ (squares).

system size using the matrix method, as opposed to the quadratic dependence expected for diffusive dynamics (i.e., the time-dependent lattice-Boltzmann method). For the highest resolution ($L=126$) and the highest porosity ($\phi=0.9$) the gain in convergence rate is approximately two orders of magnitude.

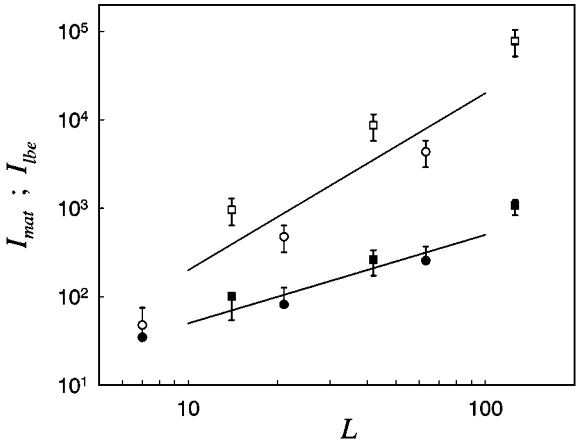

FIG. 2. Number of iterations required to reduce the relative error ε_c to less than 10^{-6} as a function of the linear dimension, L (in lattice units). The solid symbols are the results of the matrix method (I_{mat}) and the open symbols those of the time-dependent method (I_{lbe}). The system is a square array of disks with radius $R=7.5$ and porosities $\phi=0.6$ (circles) and $\phi=0.9$ (squares). The vertical bars indicate a range of convergence criteria between 10^{-4} (lowest point) and 10^{-8} (highest point). The solid lines are a guide to the eye; the lower line has a slope of 1 and the upper line has a slope of 2.

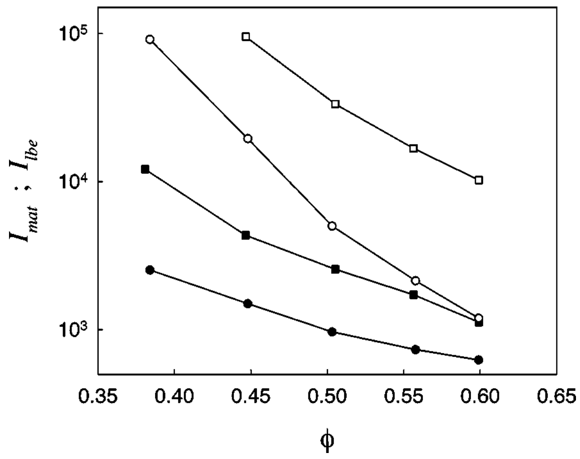

FIG. 3. Number of iterations to convergence, $\varepsilon_c<10^{-6}$, in a random array of disks as a function of porosity ϕ. The solid symbols are the results of the matrix method (I_{mat}) and the open symbols those of the time-dependent method (I_{lbe}) for 81 disks (circles) and 729 disks (squares).

Next we analyze the rate of convergence in idealized two-dimensional porous structures composed of random arrays of disks. We generated three systems, containing 9, 81, and 729 disks per unit cell with linear dimensions $L=63$, 189, and 567, respectively. The disk radius and porosity of each system were $R=7.5$ and $\phi=0.6$. For each system size, five configurations were generated by Monte Carlo simulation and from each of those 15 configurations four additional configurations were created, with porosities 0.55, 0.50, 0.45, and 0.40, by increasing the radius of the disks. Each configuration was cluster analyzed to detect percolating pore networks. As discussed in Sec. III, we simulated flow in each percolating pore structure individually and then accumulated the volumetric flow rates at the same pressure gradient. Nonpercolating pores were treated as solid nodes in these simulations, since they do not contribute to the permeability.

In Fig. 3 we show the number of iterations required to reduce the relative error to less than 10^{-6} as a function of porosity ϕ. The results are averages over five random configurations of 81 or 729 disks; for clarity the series with nine disks has not been shown, but the same qualitative results were obtained. At each lattice size, the number of iterations to convergence increases smoothly with decreasing porosity; at $\phi=0.40$ the matrix method is about 40 times faster than the time-dependent method. At this porosity, the relative error for the 729 disk system failed to converge beyond 10^{-2}, even after 10^5 steps; at this point the calculation was terminated. Thus for random arrays, the matrix method becomes much more efficient than the time-dependent method at *low* porosity, in contrast to results for periodic arrays. The reason for the slow convergence of the time-dependent method in low-porosity random structures is unknown, but it is a universal feature in both two and three dimensions.

V. THREE-DIMENSIONAL FLOWS

We have simulated flow through idealized three-dimensional porous structures composed of random arrays of

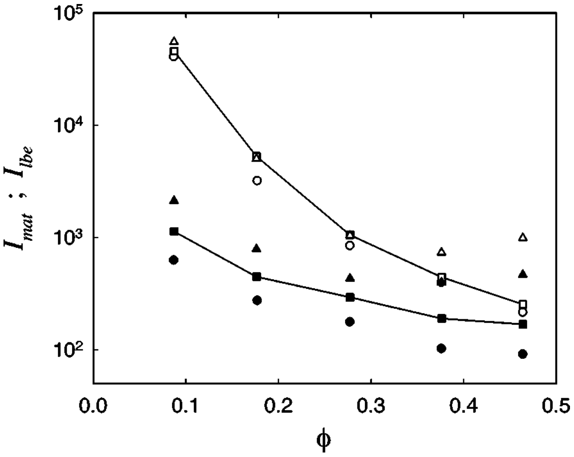

FIG. 4. Number of iterations to convergence, $\varepsilon_c < 10^{-6}$, in a random array of spheres as a function of porosity ϕ. The solid symbols are the results of the matrix method ($I_{\rm mat}$) and the open symbols those of the time-dependent method ($I_{\rm lbe}$). Results are shown for three mesh resolutions: $L=30$ (circles), $L=60$ (squares), and $L=120$ (triangles).

spheres, using the 18-speed model discussed in Ref. [20]. This model has velocities in the {100} and the {110} directions and does not include rest particles; it is isotropic and Galilean invariant. The coefficients a_0^i are equal to 1/12 for the {100} directions and 1/24 for the {110} directions, while $a_1^i = 2a_0^i$. All configurations use a cubic unit cell with periodic boundary conditions in all directions. The linear dimension of the lattice in each direction is L (i.e., $N=L^3$) and the external pressure gradient is again applied in the y direction.

Five different random configurations of 54 nonoverlapping spheres, with radius $R=2$, were obtained by melting a bcc lattice in a box with a linear dimension $L=15$, giving a porosity $\phi=0.464$. To obtain lower porosities, four new configurations were created from each of the original configurations, with porosities of approximately 0.4, 0.3, 0.2, and 0.1, again by increasing the radius. In addition, each configuration was replicated at higher mesh resolutions by increasing all linear dimensions (sphere coordinates, sphere radius, and box length) by factors of 2, 4, and 8. We emphasize that the arrangement of spheres in a particular configuration was the same for each mesh resolution and porosity.

In Fig. 4, we show the rate of convergence as a function of porosity and mesh resolution. We compare the number of iterations to convergence ($\varepsilon_c < 10^{-6}$) for the matrix method ($I_{\rm mat}$) and the conventional time-dependent lattice-Boltzmann method ($I_{\rm lbe}$). The results are averages over the five different random configurations. Just as for the disks, the matrix method becomes increasingly more efficient with decreasing porosity; at the lowest porosity it requires a factor of 20–65 fewer iterations.

The macroscopic flow of fluid in a porous medium is characterized by the permeability tensor **K**, which relates the volume-averaged momentum flow $\langle \mathbf{j} \rangle$ to the external pressure gradient $\nabla_p^{\rm ext}$ (Darcy's law),

$$\nu \langle \mathbf{j} \rangle = \mathbf{K} \cdot \mathbf{g}. \tag{24}$$

TABLE II. Reduced permeability K/R^2 and Kozeny coefficient c_0 [see Eq. (25)] for a single configuration of random spheres.

L	R	ϕ	K_{yx}/R^2 (10^{-3})	K_{yy}/R^2 (10^{-3})	K_{yz}/R^2 (10^{-3})	c_0
15	2.00	0.463	0.843	12.5	0.252	3.08
	2.11	0.376	0.482	6.00	0.240	2.52
	2.22	0.280	0.249	2.21	0.0922	2.12
	2.35	0.179	0.143	0.573	0.107	1.64
30	4.00	0.464	0.491	8.80	0.0702	4.28
	4.21	0.376	0.257	4.07	0.0584	3.67
	4.44	0.277	0.140	1.48	0.0565	2.75
	4.71	0.177	0.0735	0.379	0.0377	2.33
	5.00	0.087	0.0083	0.0295	0.0046	2.62
60	8.00	0.464	0.399	7.59	0.0409	5.08
	8.42	0.376	0.262	3.46	0.0499	4.38
	8.89	0.277	0.133	1.25	0.0434	3.61
	9.41	0.176	0.0514	0.297	0.0268	2.93
	10.0	0.087	0.0082	0.0322	0.0063	2.53
120	16.0	0.464	0.400	7.36	0.0502	5.11
	16.8	0.376	0.256	3.33	0.0503	4.49
	17.8	0.277	0.134	1.18	0.0426	3.82
	18.8	0.176	0.0474	0.275	0.0242	3.21
	20.0	0.087	0.0075	0.0284	0.0057	2.73
240	40.0	0.085	0.0074	0.0277	0.0055	2.94

We can compare our calculations of permeability with the Carman-Kozeny equation for nonoverlapping spheres (neglecting the dependence of specific surface area on microstructure) [36]:

$$\frac{K_{yy}}{R^2} = \frac{\phi^3}{9c_0(1-\phi)^2}. \tag{25}$$

Our results for c_0 at the highest porosity ($\phi=0.464$) converge to a value slightly larger than 5 (see Table II), consistent with values found in the literature for nonoverlapping spheres [37–39]. The decrease in c_0 with decreasing porosity is consistent with the expected reduction in specific surface area as the spheres overlap. However, a quantitative analysis of the dependence of c_0 on porosity is outside the scope of this paper.

The effect of mesh size on the permeability of random arrays of overlapping spheres has been analyzed as a function of porosity for one of the five initial configurations (Table II); the sphere coordinates, relative to the box length, are the same in each case. The configuration with the lowest porosity ($\phi \approx 0.1$) and the lowest resolution ($L=15$) turned out to have no percolating clusters and was discarded. It was replaced by a higher mesh resolution structure ($L=240$) at the same porosity. To compare the different mesh resolutions we normalized the permeability **K** by the square of the sphere radius R to make it dimensionless. The reduced permeabilities $\mathbf{K}/R^2$ are shown in Fig. 5 and are also summarized in Table II. It can be seen that the results for $L=60$ and $L=120$ are within 10% of one another, even at low porosity.

We have quantified the rate of convergence of the permeability with increasing mesh resolution by expressing the relative error

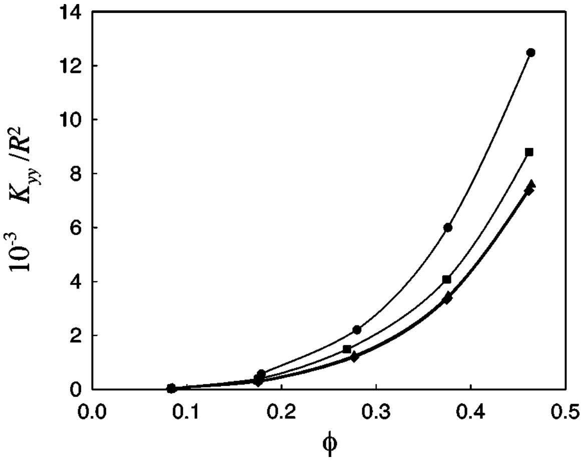

FIG. 5. Reduced permeability of a random array of spheres, K_{yy}/R^2, as a function of porosity ϕ: $L=15$ (circles), $L=30$ (squares), $L=60$ (triangles), and $L=120$ (diamonds) (cf., Table II).

$$\varepsilon=\frac{|K_{yy}(L)-K_{yy}(L=\infty)|}{|K_{yy}(L=\infty)|},\tag{26}$$

as a power of the box length, $\varepsilon\propto L^{-R}$, with

$$R(L)=\log_2\frac{|K_{yy}(2L)-K_{yy}(L)|}{|K_{yy}(4L)-K_{yy}(2L)|}.\tag{27}$$

Thus $R=1$ represents linear convergence, $R=2$, quadratic convergence, etc. For $R(15)$, the permeability of structures with linear dimensions $L=15$, 30, and 60 are used, for $R(30)$ those with linear dimensions $L=30$, 60, and 120; and for $R(60)$ those with linear dimensions $L=60$, 120, and 240. The results shown in Fig. 6 indicate that the convergence is approximately second order in the mesh resolution, as expected from our choice of bounce back rules. Deviations at smaller L are due to the fact that the convergence is not truly asymptotic. This is illustrated by the negative value of $R(30)$ at the lowest porosity, while the higher resolution simulations at the same porosity indicate quadratic convergence, $R(60)=2.5$.

VI. CONCLUSION

In this paper we have presented a numerical method to solve the Stokes equations for stationary low-Reynolds-number flow in porous media. The method gives the steady-state solution directly, utilizing the linearity of the velocity distribution function for low-Reynolds-number flows. It maintains the advantages of the lattice-Boltzmann method; in particular that arbitrarily complex geometries are easily incorporated. The new method is 1–2 orders of magnitude faster than the conventional time-dependent method for the model configurations discussed in this paper, although the actual time to update a single node is about 30% longer. The rate of convergence of the matrix method may be further increased by preconditioning [35]. A drawback of the present method is that it requires approximately 50% more memory than the conventional lattice-Boltzmann method.

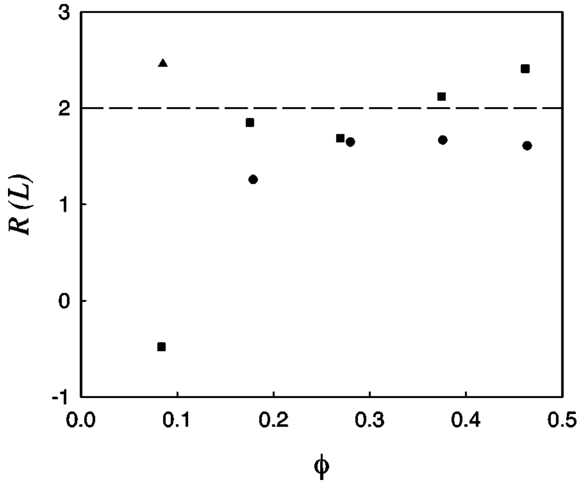

FIG. 6. Rate of convergence $R(L)$ of the permeability of a random array of spheres as a function of porosity, ϕ: $L=15$ (circles), $L=30$ (squares), and $L=60$ (triangles).

In addition, we have presented a detailed analysis of the convergence of the permeability as a function of mesh resolution for random arrays of spheres. Our results indicate second-order convergence in the mesh resolution for the link bounce-back method, in agreement with theoretical analysis for planar walls [33,34].

The matrix method can be extended to finite-Reynolds-number flows, although the lattice-Boltzmann equation is no longer linear in the velocity distribution function. However, it is impossible at this point to make any estimate of the gain in the convergence rate. This question remains open to further research.

ACKNOWLEDGMENT

This work was supported by the U.S. Department of Energy, Office of Basic Energy Sciences (Grant No. DE-FG02-98ER14853).

[1] S. Chen and G. D. Doolen, in *Annual Review of Fluid Mechanics*, edited by J. L. Lumley, M. Van Dyke, and H. L. Reed (Annual Reviews Inc., Palo Alto, California, 1998), Vol. 30, pp. 329–364.

[2] G. R. McNamara and G. Zanetti, Phys. Rev. Lett. **61**, 2332 (1988).

[3] U. Frisch, B. Hasslacher, and Y. Pomeau, Phys. Rev. Lett. **56**, 1505 (1986).

[4] F. J. Higuera, S. Succi, and R. Benzi, Europhys. Lett. **9**, 345 (1989).

[5] F. J. Higuera and J. Jiménez, Europhys. Lett. **9**, 663 (1989).

[6] X. He and L.-S. Luo, Phys. Rev. E **55**, R6333 (1997).

[7] X. He and L.-S. Luo, Phys. Rev. E **56**, 6811 (1997).

[8] T. Abe, J. Comput. Phys. **131**, 241 (1997).

[9] Y. H. Qian, D. d'Humières, and P. Lallemand, Europhys. Lett. **17**, 479 (1992).

[10] H. Chen, S. Chen, and W. H. Matthaeus, Phys. Rev. A **45**, R5339 (1992).

[11] D. H. Rothman, Geophysics **53**, 509 (1988).

[12] S. Succi, E. Foti, and F. J. Higuera, Europhys. Lett. **10**, 433 (1989).

[13] A. J. C. Ladd, J. Fluid Mech. **271**, 311 (1994).

[14] A. Koponen, Ph.D. thesis, University of Jyväskylä, 1998 (unpublished).

[15] S. Chen, Z. Wang, X. Shan, and G. D. Doolen, J. Stat. Phys. **68**, 379 (1992).

[16] D. O. Martinez, W. H. Mattaeus, S. Chen, and D. C. Montgomery, Phys. Fluids A **6**, 1285 (1994).

[17] D. R. Noble, J. G. Georgiadis, and R. O. Buckius, Int. J. Numer. Methods Fluids **23**, 1 (1996).

[18] J. Bernsdorf, F. Durst, and M. Schäfer, Int. J. Numer. Methods Fluids **29**, 251 (1999).

[19] A. Koponen *et al.*, Phys. Rev. Lett. **80**, 716 (1998).

[20] A. J. C. Ladd, J. Fluid Mech. **271**, 285 (1994).

[21] S. D. Chen, H. D. Chen, D. Martinez, and W. Matthaeus, Phys. Rev. Lett. **67**, 3776 (1991).

[22] U. Frisch *et al.*, Complex Syst. **1**, 649 (1987).

[23] R. Cornubert, D. d'Humières, and C. D. Levermore, Physica D **47**, 241 (1991).

[24] D. P. Ziegler, J. Stat. Phys. **71**, 1171 (1993).

[25] I. Ginzbourg and P. M. Adler, J. Phys. II **4**, 191 (1994).

[26] P. A. Skordos, Phys. Rev. E **48**, 4823 (1993).

[27] D. R. Noble, S. Y. Chen, J. G. Georgiadis, and R. O. Buckius, Phys. Fluids **7**, 203 (1995).

[28] T. Inamuro, M. Yoshino, and F. Ogino, Phys. Fluids **7**, 2928 (1995).

[29] R. S. Maier, R. S. Bernard, and D. W. Grunau, Phys. Fluids **8**, 1788 (1996).

[30] S. Chen, D. Martinez, and R. Mei, Phys. Fluids **8**, 2527 (1996).

[31] Q. Zou and X. He, Phys. Fluids **9**, 1591 (1997).

[32] A. J. C. Ladd and D. Frenkel, in *Cellular Automata and Modeling of Complex Physical Systems*, edited by P. Manneville, N. Boccara, G. Y. Vichniac, and R. Bidaux (Springer-Verlag, Berlin, 1989), pp. 242–245.

[33] X. He, Q. Zou, L.-S. Luo, and M. Dembo, J. Stat. Phys. **87**, 115 (1997).

[34] A. J. C. Ladd, in *Dynamics: Models and Kinetic Methods for Non-equilibrium Many Body Systems*, edited by H. van Beijeren and J. Karkheck (Kluwer Academic, Dordrecht, 1999).

[35] A. Greenbaum, *Iterative Methods for Solving Linear Systems* (Society for Industrial and Applied Mathematics, Philadelphia, 1997).

[36] P. C. Carman, Trans. Inst. Chem. Eng. **15**, 1550 (1937).

[37] A. P. Philipse and C. Pathmamanoharan, J. Colloid Interface Sci. **159**, 96 (1993).

[38] A. W. J. Heijs and C. P. Lowe, Phys. Rev. E **51**, 4346 (1995).

[39] R. S. Maier *et al.*, Phys. Fluids **10**, 60 (1998).