Lattice Boltzmann Simulations of Single and Multi-Component Flow in Porous Media

Nicos S. Martys[a], John G. Hagedorn[b], and Judith E. Devaney[b]

[a]National Institute of Standards and Technology
100 Bureau Drive, Stop 8621
Gaithersburg, MD 20899-8621, USA

[b]National Institute of Standards and Technology
100 Bureau Drive, Stop 8951
Gaithersburg, MD 20899-8951, USA

ABSTRACT

We examine the utility of the lattice Boltzmann method for modeling fluid flow in porous media. First the lattice Boltzmann method will be validated for the case of single component flow in several idealized flow geometries. Large scale simulations of fluid flow through digitized images of Fontainebleau sandstone, generated by X-ray microtomography, will then be presented. Reasonably good agreement was found when compared to experimentally determined values of permeability for similar rocks. In order to model flow in porous media well described by Darcy's law or large pore size variation, an extension of the lattice Boltzmann method for approximating the Brinkman equation will be described. We will then discuss the modeling of multicomponent fluid systems. The critical properties of the binary mixture used in the fluid simulation will be shown to be consistent with mean-field theory. Relative permeability curves as a function of fluid saturation and driving force will then be presented. Values of permeability from three phase flows are compared to corresponding two phase values. Performance on several computing platforms is given.

Keywords: Brinkman equation, critical phenomena, lattice Boltzmann, microtomography, parallel computing, permeability, porous media

1. INTRODUCTION

The lattice Boltzmann (LB) method has evolved into a powerful computational method for the modeling of fluid flow in complex geometries like porous media. It naturally accommodates a variety of boundary conditions such as the pressure drop across the interface between two fluids and wetting effects at a fluid-solid interface. Since the LB method can be derived from the Boltzmann equation, its physical underpinnings can be understood from a fundamental point of view. In addition, the LB method generally needs nearest neighbor information at most so that it is ideally suited for parallel computers. While LB methods are developing rapidly in response to recent theoretical advances and the availability of resources for large scale computation, there is still a lack of critical comparisons between theoretical and experimental results with simulation data. Such comparisons are crucial, not only to validate LB methods, but to advance their development. Key to modeling multicomponent systems is the correct discription of the fluid/fluid interaction and its proper incorporation into the lattice Boltzmann model. In this chapter we examine two approaches for the modeling of multicomponent fluids in complex geometries. The first is the model of Shan and Chen[?] where a simple effective interaction potential is proposed to describe the fluid/fluid interaction. The Shan Chen model then incorporates the forcing due to the potential by shifting the velocity in the equilibrim distribution. The second approach is based on the direct incorporation of the fluids mutual forcing into the body force term of the Boltzamann equation. We also describe an approximation of the use of the Brinkman eqaution which can link regions treated as a simple homogeneous porous medium with larger pore regions. While, both the method of Shan and Chen and the direct body forcing approach may be used to model the multicomponent

Further author information:
N.S.M.: E-mail: nicos.martys@nist.gov
J.G.H.: E-mail: john.hagedorn@nist.gov
J.E.D.: E-mail: judith.devaney@nist.gov

fluid sysmtems and Brinkman equation there may be some advantage to using the body forcing. After a brief review of the theory of the LB methods used, results are presented to validate and compare methods, including predictions of fluid flow through a few simple pore geometries. Large scale simulations of fluid flow through a Fontainebleau sandstone microstructure, which was generated by X-ray microtomography, will then be presented. Single phase flow calculations were carried out on 510^3 systems. We also calculate relative permeability curves as a function of fluid saturation and driving force. Values of relative permeability from three phase flows were compared to corresponding two phase values. Finally, a comparison of the performance of such codes on different computing platforms is given.

2. LATTICE BOLTZMANN MODEL WITH FLUID PHASE SEPARATION

The LB method of modeling fluid dynamics is actually a family[?] of models with varying degrees of faithfulness to the properties of real liquids. These methods are currently in a state of evolution as the models become better understood and corrected for various deficiencies. In this article we utilize two version of LB. The first was proposed by Shan and Chen[?,?] that is particularly simple in form and adaptable to complex flow conditions like the presence of solid-fluid and fluid-fluid boundaries. The second model, is based on the correct implementation of the body force term in the Boltzmann equation. The introduction of the fluid/fluid interactions can be thought of as a mean field produced by neighboring fluid components. This forcing can be incorporated into the body force term. This second approach removes second order artifacts in the pressure tensor and is found to produce more realistic flow in certain situations.

The approach of LB is to consider a typical volume element of fluid to be composed of a collection of particles that are represented in terms of a particle velocity distribution function at each point in space. The particle velocity distribution, $n_a^i(\mathbf{x}, t)$, is the number density of particles at node $\mathbf{x}$, time t, and velocity, $\mathbf{e}_a$, where $(a = 1, ..., b)$ indicates the velocity direction and superscript i labels the fluid component. The time is counted in discrete time steps, and the fluid particles can collide with each other as they move under applied forces.

For this study we use the D3Q19 (3 Dimensional lattice with $b = 19$)[?] lattice.[?] The microscopic velocity, $\mathbf{e}_a$, equals all permutations of $(\pm 1, \pm 1, 0)$ for $1 \leq a \leq 12$, $(\pm 1, 0, 0)$ for $13 \leq a \leq 18$, and $(0, 0, 0)$ for $a = 19$. The units of $\mathbf{e}_a$ are the lattice constant divided by the time step. Macroscopic quantities such as the density, $n^i(\mathbf{x}, t)$, and the fluid velocity, $\mathbf{u^i}$, of each fluid component, i, are obtained by taking suitable moment sums of $n_a^i(\mathbf{x}, t)$. Note that while the velocity distribution function is defined only over a discrete set of velocities, the actual macroscopic velocity field of the fluid is continuous.

The time evolution of the particle velocity distribution function satisfies the following LB equation:

$$n_a^i(\mathbf{x} + \mathbf{e}_a, t + 1) - n_a^i(\mathbf{x}, t) = \Omega_a^i(\mathbf{x}, t) - B_a^i, \tag{1}$$

where Ω_a^i is the collision operator representing the rate of change of the particle distribution due to collisions and B_a^i is a body force term. The collision operator is greatly simplified by use of the single time relaxation approximation[?,?]

$$\Omega_a^i(\mathbf{x}, t) = -\frac{1}{\tau_i}\left[n_a^i(\mathbf{x}, t) - n_a^{i(eq)}(\mathbf{x}, t)\right], \tag{2}$$

where $n_a^{i(eq)}(\mathbf{x}, t)$ is the equilibrium distribution at $(\mathbf{x}, t)$ and τ_i is the relaxation time that controls the rate of approach to equilibrium. The equilibrium distribution can be represented in the following form for particles of each type[?,?]:

$$n_a^{i(eq)}(\mathbf{x}) = t_a n^i(\mathbf{x})\left[\frac{3}{2}(1 - d_o) + 3\mathbf{e}_a \cdot \mathbf{v} + \frac{3}{2}(3\mathbf{e}_a\mathbf{e}_a : \mathbf{vv} - \mathbf{v}^2)\right] \tag{3}$$

$$n_{19}^{i(eq)}(\mathbf{x}) = t_{19} n^i(\mathbf{x})\left[3d_o - \frac{3}{2}\mathbf{v}^2\right], \tag{4}$$

where

$$\mathbf{v} = \frac{\sum_i^S m^i \sum_a n_a^i \mathbf{e}_a/\tau_i}{\sum_i^S m^i n^i(\mathbf{x})/\tau_i}, \tag{5}$$

and where m^i is the molecular mass of the ith component, and $t_a = 1/36$ for $1 \leq a \leq 12$, $t_a = 1/18$ for $13 \leq a \leq 18$ and $t_{19} = 1/3$. The free parameter d_o can be related to an effective temperature, T, for the system by the following moment of the equilibrium distribution:

$$T(\mathbf{x}, t) = \frac{\sum_a n_a^{i(eq)}(\mathbf{x}, t)(\mathbf{e}_a - \mathbf{v})^2}{3n^i(\mathbf{x}, t)}, \tag{6}$$

which results in $T = (1 - d_o)/2$ (we take units such that the Boltzmann constant $k_b = 1$).

A representation [8] of this body force term, to second order in Hermite polynomials, in the discrete velocity space of the D3Q19 lattice is

$$B_a^i = -3t_a n^i(\mathbf{x}) \left[(\mathbf{e_a} - \mathbf{v}) \cdot \frac{\mathbf{F_b}^i}{\rho^i} + 3(\mathbf{e_a} \cdot \mathbf{v})(\mathbf{e_a} \cdot \frac{\mathbf{F_b}^i}{\rho^i}) \right] \tag{7}$$

where $\mathbf{F_b}^i$ is the body force on fluid i.

It has been shown that the above formalism leads to a velocity field that is a solution of the Navier-Stokes[?] equation with the kinematic viscosity, $\nu = \frac{c^2}{6}(\sum_i^S c_i \tau_i - \frac{1}{2})$ where c_i is the concentration of each component.[?]

2.1. Interaction Potential

In order to model the phase separation of fluids, an interaction between the fluids is needed to drive them apart. Shan and Chen proposed a simple form force, $\mathbf{F}^i$, acting on fluid i:

$$\mathbf{F}^i(\mathbf{x}) = -n^i(\mathbf{x}) \sum_{i'}^{S} \sum_{a} G^a_{ii'} n^{i'}(\mathbf{x} + \mathbf{e}_a)\mathbf{e}_a \tag{8}$$

with $G^a_{ii'} = 2G$ for $|\mathbf{e}^a| = 1; G^a_{ii'} = G$ for $|\mathbf{e}^a| = \sqrt{2}$; and $G^a_{ii'} = 0$ for $i = i'$. G is a constant that controls the strength of the interaction. It has been shown that the above forcing term produces a surface tension effect consistent with Laplaces law.

In the LB model of Shan and Chen, phase separation takes place when the mutual diffusivity of the binary mixture becomes negative and this provides a condition determining the critical coupling G_c for phase separation. An analytical expression for the mutual diffusivity has been determined in a previous work.[?] For a viscosity matched binary mixture, phase separation occurs when

$$G_c = \frac{T}{48} \frac{-c_1 n^1 - c_2 n^2 + \sqrt{(c_1 n^1 + c_2 n^2)^2 + 8n^1 n^2}}{n^1 n^2}. \tag{9}$$

At the time it was not understood how to properly include the body forcing term of the Boltzmann equation because it was not known how to properly evaulate the microscopic velocity derivative of the Boltzmann equation. Shan and Chen suggested a clever way to incorporate forcing into the BGK collision operator by shifting the velocity in the equilibrium distribution[?,?]:

$$n^i(\mathbf{x})\mathbf{v}'(\mathbf{x}) = n^i\mathbf{v}(\mathbf{x}) + \tau_i \frac{d\mathbf{p}^i}{dt}(\mathbf{x}) \tag{10}$$

where $\mathbf{v}'$ is the new velocity used in Eqs. [3] and [4].

This method automatically satisfies conservation of mass and since the forcing is based on a simple gradient term it will conserve momentum globally. Note, unlike the usual BGK operator or a competing model binary mixtures due to Yeomans et.al., momentum is not conserved locally. Not conserving momentum locally may be physically more appealing since it indicates that the forcing results in a momentum transfer between molecules in neighboring regions.

Instead of shifting the velocity as in the Shan Chen model the fluid/fluid interaction can be incorperated into the Body force term of the Boltzmann eqaution by setting ${\mathbf{F_b}}^i = \mathbf{F}^i$.

To give the interaction potenial of Shan and Chen a more physical basis, it can be shown, starting from the BBGKY formalism that the correct form of forcing for a "mean-field" fluid mixture can be derived when condsidering long range interactions and invoking a molecular chaos assumption.

$\mathbf{F}^i = \rho^i \nabla V_m^{ij}$ where $V_m^{ij} = 2a_{ij}\rho^j(\vec{r}_1) + \kappa_{ij}\nabla^2\rho^j(\vec{r}_1)$ with $a_{ij} = \frac{1}{2}\int_d d^3 r V_{ij}(r)$ and $\kappa_{ij} = \frac{1}{6}\int_d d^3 r r^2 V_{ij}(r)$. V_m^{ij} can be thought of as a mean field potential produced by neighboring particles.

The pressure tensor can be determined for this systems and is given by

$$\begin{aligned}
\overleftrightarrow{P} &= (-a_{11}{\rho_1}^2 - a_{22}{\rho_2}^2 - 2a_{12}\rho_1\rho_2 + \kappa_{11}(\frac{1}{2}|\nabla\rho_1|^2 + \rho_1\nabla^2\rho_1) \\
&\quad + \kappa_{22}(\frac{1}{2}|\nabla\rho_2|^2 + \rho_2\nabla^2\rho_2) + \kappa_{12}(\nabla\rho_1\cdot\nabla\rho_2 + \rho_1\nabla^2\rho_2 + \rho_2\nabla^2\rho_1))\ \overleftrightarrow{I} \\
&\quad - \kappa_{11}\nabla\rho_1\nabla\rho_1 - \kappa_{22}\nabla\rho_2\nabla\rho_2 - 2\kappa_{12}(\nabla\rho_1\nabla\rho_2 + \nabla\rho_2\nabla\rho_1).
\end{aligned}$$

The forcing suggested by Shan and Chen corresponds to the case where all terms in the above pressure tensor are zero accept that with the coefficient a_{12} which is proportional to the coupling constant, G, described earlier in the paper. Surprisingly there is no explict surface tension term in their model! Actually, the effective surface tension force is an artifact from the discretization of the forcing. This can be seen by noting that

$$F \sim \rho(x+\Delta x) - \rho(x-\Delta x) = 2\Delta x\frac{\partial\rho}{\partial x} + \frac{2}{3\cdot 2}(\Delta x)^3\frac{\partial^3\rho}{\partial x^3} + \cdots + . \tag{11}$$

The second term on the right produces the effective surface tension effect.

Below is a phase diagram for the Shan-Chen model for the case of a critical mixure ($c_1 = c_2 = \frac{1}{2}$ globally). Data is also included for the body forcing model.

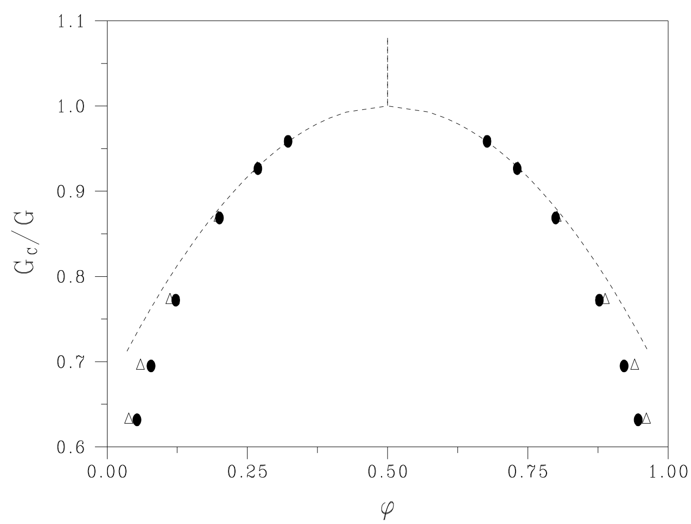

Figure 1. Phase diagram of LB fluid mixture for Shan-Chen (filled circles) and body-forcing based model (triangles). The dashed line coresponds to the mean-field theory prediction. The coexisting phase fraction of a fluid is given by ϕ.

3. IMPLEMENTATION

The approach to implementation of the algorithm is relatively straightforward. At each active site we hold the necessary velocity and mass data for each fluid component. Over the course of an iteration we visit each cell in the data volume and calculate the distribution of each fluid component to be streamed to neighboring cells. New mass and velocity values are accumulated at each cell as its neighbors make their contributions. The most notable aspects of the implementation were our tactics for managing the large amounts of memory required by the algorithm, and the adaptation of the code for use in parallel computing environments.

3.1. Memory Optimizations

Experience with the implementation of related algorithms indicated that the memory required for modeling large systems would be prohibitive. We therefore looked for ways to conserve and reduce memory usage. There are several tactics that we used in this implementation:

- Store data only at the active sites.
  This is accomplished in the C implementation by representing the medium as a three dimensional array of pointers. At each active site the pointer references a data structure with the necessary velocity and mass data. At the inactive sites the pointer is NULL; no additional storage is required at the inactive sites. For a low porosity medium the memory savings are very large.
- Assume that $\tau = 1$.
  This assumption simplifies evaluation of equations 1-5 such that at each active site we need only store the density of each fluid component, and a single velocity vector. Without this assumption, we must store all 19 values associated with the velocity distribution, n_i, at each site.
- Only one copy of the data volume is stored.
  Rather than keeping an entire second data volume in which to accumulate the newly calculated data, we exploit the fact that the algorithm only uses nearest neighbors at each site. Thus we only need an additional buffer of three planes of data at any one time.

Assuming that floating point numbers and C pointers each take four bytes, these memory optimizations yield savings of over 94 % of memory usage in the one component case for systems of useful sizes. The memory savings are even greater when more fluid components are used or when larger floating point representations are used.

3.2. Parallelization

The amount of computation and memory required for a large system suggested that it would be advantageous to adapt the implementation so that a single problem could be run in parallel across a collection of processors. The nearest-neighbor dependence of the algorithm also suggested that parallelization would be straightforward and would yield substantial benefits. Parallelization enables us to run larger systems by distributing the memory requirements across many machines, and gives us faster performance by distributing the computation.

We implemented the parallel version of the algorithm using the Message Passing Interface[?] (MPI). This is an industry-standard library of routines for coordinating execution and communicating between processes in a parallel computing environment. The parallelization was accomplished within a simple Single Program Multiple Data (SPMD) model. The data volume is divided into spatially contiguous blocks along the Z axis; multiple copies of the same program run simultaneously, each operating on its block of data. Each copy of the program runs as an independent process and typically each process runs on its own processor. At the end of each iteration, data for the planes that lie on the boundaries between blocks are passed between the appropriate processes and the iteration is completed. The periodic boundary condition is handled transparently; the process handling the “top” plane of data volume simply exchanges data with the process handling the “bottom” plane of the data volume.

4. NUMERICAL TESTS

Several numerical tests were carried out to verify our algorithm. Results from two cases, fluid flow between parallel plates and through an overlapping sphere model, are given below. For both cases we determined the fluid permeability, k, as defined by Darcy's law, $\langle \vec{v} \rangle = -\frac{k}{\mu} \nabla P$, where $\langle \vec{v} \rangle$ is the average flow rate, ∇P is the pressure gradient and μ is the fluid viscosity. Figure 1 shows the permeability, in units of the lattice spacing squared, as a function of the distance between parallel plates. Clearly, there is excellent agreement between the simulation and theoretical prediction. Surprisingly, very accurate results were obtained even for the case of a one node wide channel. Since permeability depends on the average flow or net flux rate of fluid, we conclude that the LB method accurately determines the net flux across a voxel surface, not the velocity at a point. Hence, resolving the actual local flow field at a point would require more nodes. We next consider the permeability of a simple cubic array of spheres that are allowed to overlap for large enough radius (i.e. when the solid fraction, c, exceeds $c \approx .5236$). In Fig. 2 we compare our simulation data with that of Chapman and Higdon,[?] which is based on the numerical solution of coefficients of a harmonic expansion that satisfies the Stokes equations. Note that our calculations were performed on a relatively small 64^3 system. Again, agreement is very good, especially given that the solid inclusion is a digitized sphere.

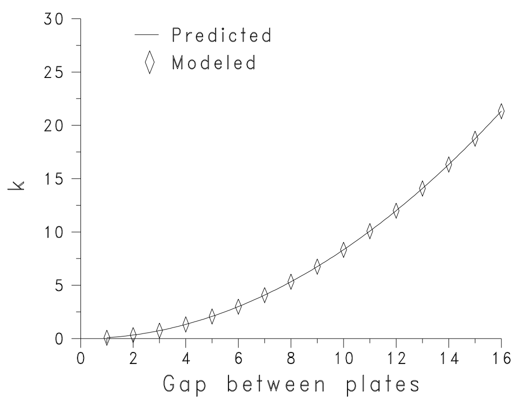

Figure 1. Flow through parallel plates.

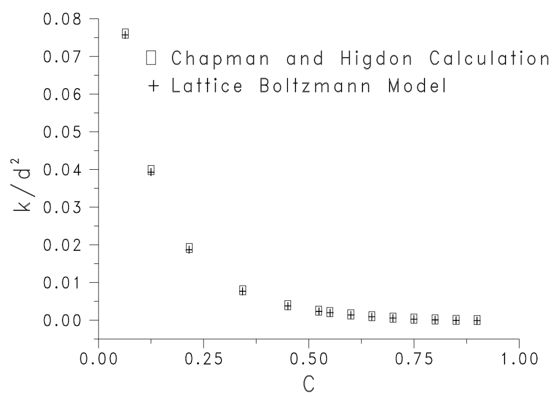

Figure 2. Flow through spheres centered on a simple cubic lattice. The permeability is normalized by the square of the distance, d, between sphere centers.

5. COMPARISON WITH EXPERIMENTAL DATA

We next determined the permeability of several microtomography-based images of Fontainebleau sandstone. Figure 3 depicts portions of two of these sandstone images. The resolution was 5.72μm per lattice spacing and data sets were 510^3 voxels. A mirror image boundary condition was applied along directions perpendicular to the applied forcing. The porous medium was made periodic in the flow direction by creating its mirror image at the inlet. The numerical calculations were carried out on a $1020 \times 510 \times 510$ system for all but the lowest porosity system. We found that at the lowest porosity (7.5 %) there were not enough nodes across the pores to produce a reliable flow field. So for this case the permeability was determined from a 256^3 piece of the sandstone image that was mapped to a 512^3 image, and calculations were performed on a $1024 \times 512 \times 512$ system. In addition to requiring sufficient resolution, another potential source of error is not having precise knowledge of the location of the pore/solid interface. For example, an error of half a lattice spacing could be significant when modeling flow in narrow channels like that in the low porosity system. Figure 6 shows the computed permeability compared to experimental data.[?] Clearly there is good agreement, especially at the higher porosities.

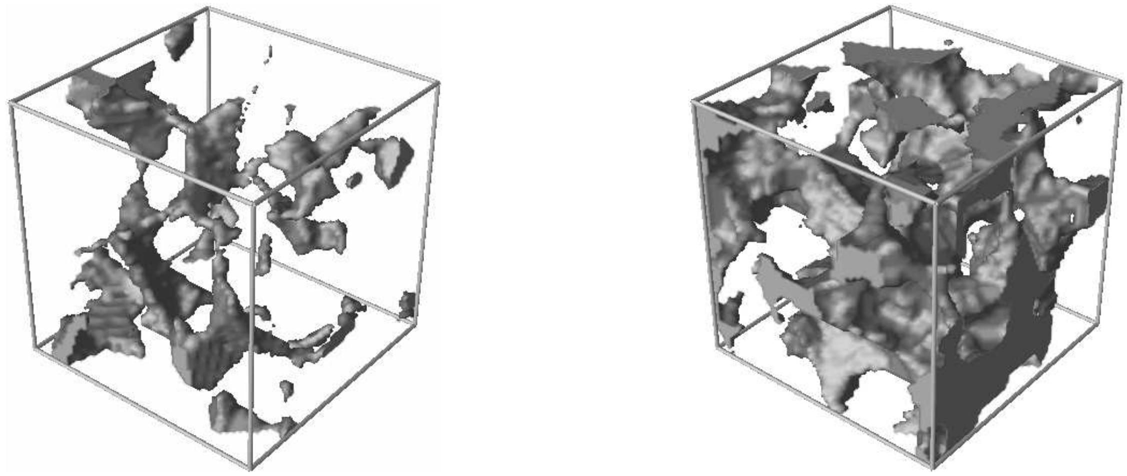

Figure 3. 64×64 portions of the Fontainebleau sandstone media. On the left is the 7.5 % porosity medium, on the right is the 22 % porosity medium. The solid matrix is made transparent to reveal the pore space (grey shaded region).

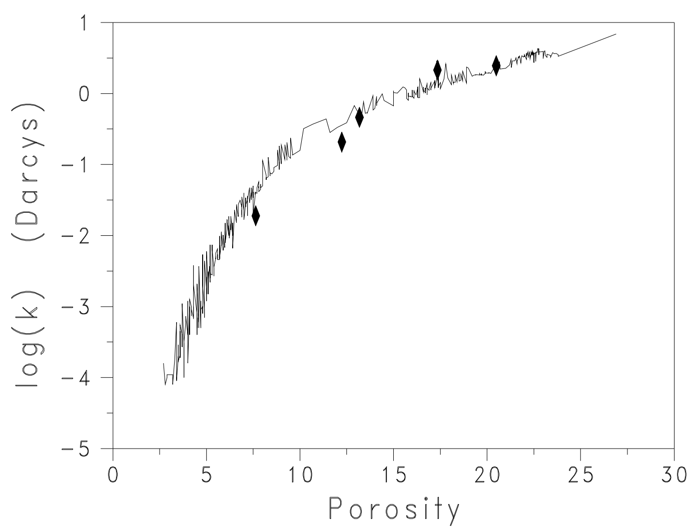

Figure 4. Measured and modeled permeabilities of Fontainebleau sandstone medium.

6. RELATIVE PERMEABILITY

We next present a sample calculation of the relative permeability for the 22 % porosity Fontainebleau sandstone. Although there is debate as to the correct formulation of the macroscopic two phase flow equations,[?] we use the following empirical relation to describe the response of a multiphase fluid system to an external driving force:

$$\vec{v}_1 = -\frac{K_{12}}{\mu_2}\nabla P_2 - \frac{K_{11}}{\mu_1}\nabla P_1 \tag{12}$$

$$\vec{v}_2 = -\frac{K_{21}}{\mu_1}\nabla P_1 - \frac{K_{22}}{\mu_2}\nabla P_2 \tag{13}$$

Here the K_{ij} are the components of a permeability tensor and the applied pressure gradient on each fluid component ∇P_i is from a simple body force, $\nabla P = \rho g$, where g is an acceleration constant. The forcing can be applied to each phase separately allowing determination of the off-diagonal terms in the permeability tensor. The viscosity

μ_i is the same for both fluids. Relative permeability data is usually presented in terms of constant capillary number, $C_a = \frac{\mu v}{\gamma}$, where γ is the interfacial surface tension. For our body force driven fluids, we can define an effective capillary number, C_a^*, by replacing v with the Darcy velocity so that $C_a^* = \frac{\mu < v >}{\gamma} = \frac{k \rho g}{\gamma}$. Below is a plot of the relative permeability of the $\phi = 22$ % rock for the cases of $C_a^* = 7.5 \times 10^{-4}$ and 7.5×10^{-5}.

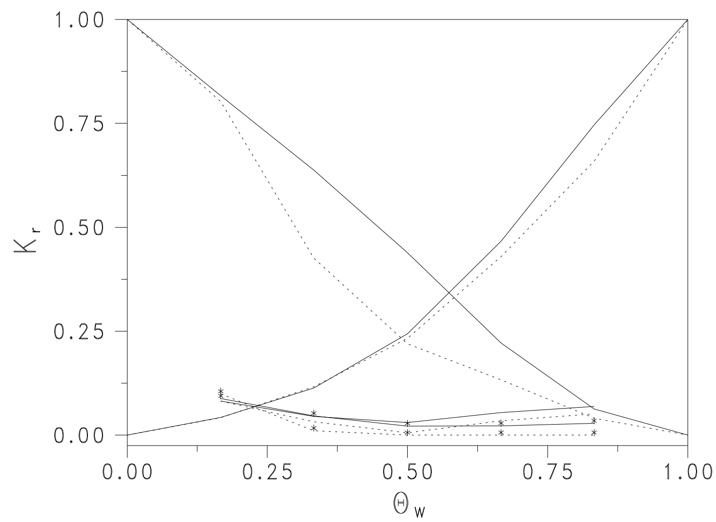

Figure 5. Relative permeabilities of 22 % porosity Fontainebleau sandstone versus wetting fluid saturation, Θ_W. The solid and dashed lines correspond to $C_a^* = 7.5 \times 10^{-4}$ and $C_a^* = 7.5 \times 10^{-5}$ respectively. The lower curves correspond to the off-diagonal elements of the permeability tensor with the $*$ denoting the case where the nonwetting fluid is driven.

Clearly, as the forcing decreases the relative permeability decreases. Also note that at lower wetting saturation the relation $K_{12} = K_{21}$ holds fairly well. This is the well known Onsager relation. However, as the wetting phase increases, this relation appears to break down. In this regime, the nonwetting phase is beginning to form disconnected blobs that do not respond in a linear fashion to the applied force due to pinning effects as the nonwetting blobs are pushed through the smaller pores.

The LB code can be easily extended to model three or more fluid components. As a simple test case we considered a three component system with each component having a pore volume fraction of 1/3. In addition, two phases were made non-wetting while the third was wetting. A forcing was applied to one of the nonwetting phases. We found, over the range $7.5 \times 10^{-4} < C_a^* < 3.75 \times 10^{-3}$, there was an approximately 30 % to 35 % decrease in flow of the driven nonwetting fluid as compared to the case where 1/3 fluid was non wetting and 2/3 was wetting. Presumably, this decrease can be understood as the non-driven nonwetting phase interfering with the flow of the driven nonwetting phase as the fluids move through narrow channels.

7. APPROXIMATION OF BRINKMAN EQUATION

Modeling fluid flow in porous heterogeneous materials [1] (e.g. concrete, microporous rocks and fractured materials) presents a challenge because it is difficult to simultaneously resolve all the microstructural features of the porous medium that are at different length scales. One possible approach is to divide the porous medium into two regions: (1) the larger pores and (2) homogeneous regions of smaller pores. In the larger pores, the Stokes' equations for incompressible flow hold:

$$\nabla p = \mu \nabla^2 \mathbf{v} \tag{14}$$

$$\nabla \cdot \mathbf{v} = 0 \tag{15}$$

where p is the pressure, $\mathbf{v}$ is the fluid velocity and μ is the fluid viscosity. Regions with the smaller pores are treated as a permeable medium and flow is described by Darcy's law:

$$\langle \nabla p \rangle = -\frac{\mu}{k}\langle \mathbf{v} \rangle \tag{16}$$

where k is the permeability of the porous medium and $\langle\rangle$ denotes the volume average.

The two boundary conditions to be satisfied at the pore/permeable medium interface are continuity of the fluid velocity and the shear stress [2]. Darcy's law alone is not sufficient to satisfy these boundary conditions. The Brinkman equation [3] is a generalization of Darcy's law that facilitates the matching of boundary conditions at an interface between the larger pores and the permeable medium. Brinkman's equation is

$$\langle \nabla p \rangle = -\frac{\mu}{k}\mathbf{v} + \mu_e \nabla^2 \langle \mathbf{v} \rangle \tag{17}$$

where $\mathbf{v}$ is the fluid velocity, μ is the fluid viscosity, and μ_e an effective viscosity parameter. The so called effective viscosity should not be thought of as the viscosity of the fluid but allows for matching of the shear stress across the free-fluid/porous medium interface. That is, $(\mu d\langle v \rangle/dy \ (y = 0^+) = \mu_e d\langle v \rangle/dy \ (y = 0^-)$ where $y = 0$ specifies the location of the interface. The $+$ and $-$ refer to regions in the free-fluid and porous medium respectively.

Although the Brinkman equation is semiempirical in nature it has been validated by detailed numerical solution of the Stokes equations in regions near the interface between dissimilar regions [4]. While numerical solution of the Brinkman equation by more traditional computational methods (e.g. finite difference and finite element) is certainly possible, a recent lattice Boltzmann (LB) based model by Spaid and Phelan [5] has proven to be a simple and computationally efficient method to numerically approximate fluid flow described by the Brinkman equation. While their formalism is capable of describing the more general case of $\frac{\mu_e}{\mu} \neq 1$, their treatment assumed $\frac{\mu_e}{\mu} = 1$ in the absence of any definitive knowledge about this ratio. However, theoretical studies [6] and numerical simulations [4] have demonstrated that this limiting case is only true when the porosity, $\phi \to 1$ and that $\frac{\mu_e}{\mu}$ increases with solid fraction [4,6]. For example, when $\phi = 50\%$, it was found that $\frac{\mu_e}{\mu} \approx 4$ for an overlapping sphere model of porous media [4]. Indeed, $\frac{\mu_e}{\mu}$ appeared to be independent of how the fluid flow was driven indicating that $\frac{\mu_e}{\mu}$ can be thought of as a material parameter that depends on the pore geometry. Therefore, it is important to consider this more general case when constructing and validating a numerical method to approximate the Brinkman equation.

To model the momentum loss in the Brinkman equation we take, as in the previous model [5], $\mathbf{F_b} = -\frac{\mu \mathbf{v}}{k}$. In addition, the relaxation time to be used in the permeable medium is taken to be $\tau_e = 3\frac{\mu_e}{n} + \frac{1}{2}$. In the limit of low Reynolds number, these modifications will recover the Brinkman equation with the option of $\frac{\mu_e}{\mu} \neq 1$. The previous model of Spaid and Phelan introduced the body forcing by shifting the velocity in the equilibrium distribution according to the prescription of Shan and Chen. Unfortunately, we have found that this implimentation can lead to large errors for many cases tested (in particular, lower porousity systems of interest). Also, when lumping other forces due to other fluids or fields together (i.e. $F \approx F_1 + F_2 ...$ where the F_i represnet different sources of forcing) and shifting the velocity in the equilibrium distribution further obfusticates the correct physics as the method of Shan-Chen produces errors of order F^2. Instead, the dissipative forcing is implimented in the body-force term so that its linearity is maintained.

To validate this model, a simple Couette flow geometry was used (see Fig. 1). Starting with a parallel plate geometry, a permeable medium is positioned such that there is a gap between the permeable medium and the upper plate. The upper plate is given a velocity V_w to the right. Analytic solution of the Brinkman equation predicts a linear velocity profile in the gap and an exponentially decaying velocity profile in the porous medium. The rate of decay depends on the value of $\sqrt{\frac{\mu_e}{\mu}}$ [2,4]. In Fig. 1 velocity profiles are compared for the case of $\frac{\mu_e}{\mu}$=4 and the assumption of $\frac{\mu_e}{\mu} = 1$. The solid line is the analytic solution of the Brinkman solution. Clearly, there is excellent agreement between simulation and theory and there is excellent agreement between simulation and theory and there can be a considerable change in the velocity profile when $\frac{\mu_e}{\mu} \neq 1$. In addition, the lattice Boltzmann method also does a reasonably good job capturing the discontinuity of the gradient of the velocity field at the free-fluid/porous medium interface for the case of $\frac{\mu_e}{\mu}$=4. Note that this is achieved without direct incorporation of the stress boundary condition in the simulation model.

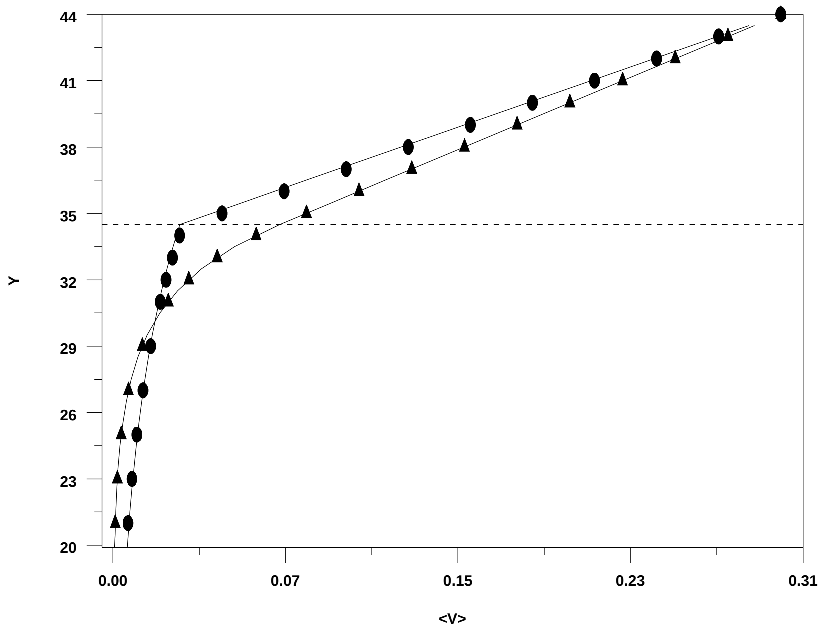

Figure 6. Velocity field of a sheared system next to a porous medium. The filled triangles and circles represent data from the lattice Boltzmann simulation ($\mu_e/\mu = 1$ and $\mu_e/\mu = 4$, respectively). The solid lines are analytic solutions of the Brinkman equation. The region below the dashed line $y = 34.5$ (in units of lattice spacing) corresponds to the porous medium. The moving wall is at $y = 44$.

8. PERFORMANCE RESULTS

We ran a series of timing tests in an effort to understand how performance of our implementation scales on different computer architectures. We have tested on an SGI Onyx with 12 R10000 processors running at 196MHz, an IBM SP2 with 37 RS/6000 processors, most running at 66MHz. The same code and the same cases were run on the two systems. The results are presented in Tables 1 and 2. The performance reported was somewhat affected by other jobs that were running at the same time that the tests were being run, although efforts were made to minimize the effect.

# Processors	# Fluid Components 1	2	3
1	14.70	24.70	33.27
2	7.39	12.22	16.69
4	3.80	6.23	8.57
8	2.14	3.48	4.68

Table 1. Execution times in seconds for one iteration on the SGI Onyx.

# Processors	# Fluid Components 1	2	3
1	38.48	62.36	99.93
2	19.30	31.32	51.16
4	10.44	16.83	26.97
8	6.86	10.01	15.54
16	4.37	6.00	8.30

Table 2. Execution time in seconds for one iteration on the IBM SP2.

These data closely agree with a very simple model describing performance: $T = P/N + S$, where T is the total time for a single iteration, P is the time for the parallelizable computation, S is the time for the non-parallelizable computation, and N is the number of processors. The parallelizable computation is that portion of the processing that can be effectively distributed across the processors. The non-parallelizable computation includes processing that cannot be distributed; this includes time for inter-process communication as well as computation that must be performed either on a single processor, or must be done identically on all processors.

For example, the two-component fluid performance data for the SGI Onyx, closely match this formula: $T = 4.78 + 487.26/N$ seconds, where N is the number of processors. Similarly, the timings for the two component runs on the IBM SP2 closely match: $T = 41.67 + 1198.45/N$ seconds. Formulae for the other cases are easily derived. Figures 6 and 7 present these results graphically.

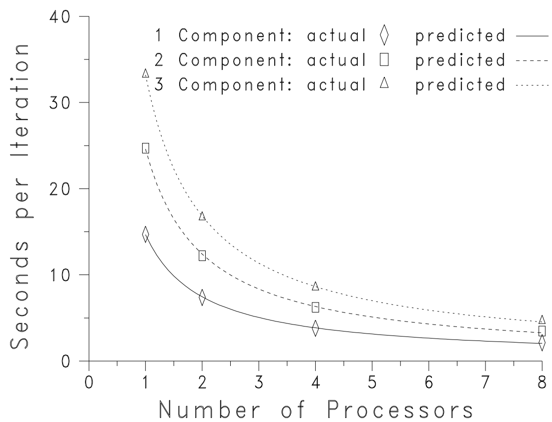

Figure 6. Time in seconds for one iteration on the SGI Onyx.

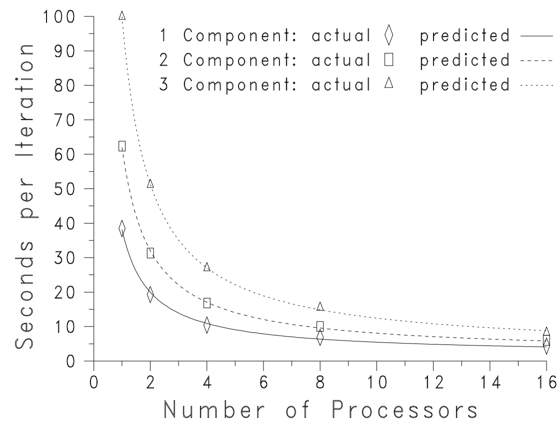

Figure 7. Time in seconds for one iteration on the IBM SP2.

Much of the difference between the performance of these two systems is likely due simply to the relative computational speeds of each processor. But the difference in the serial overhead (4.78 seconds on the SGI versus 41.67 seconds on the IBM), is most likely due to the different memory architectures of the two systems. The SGI Onyx uses a Non-Uniform Memory Access (NUMA) architecture that enables processes to pass data to one another through shared memory, However, on the IBM SP2 no memory is shared and data must be transferred over an external high-speed network. Thus the overhead for message passing on the SGI Onyx is considerably lower than that on the IBM SP2. We intend to run timing tests to measure the difference in message passing overhead.

The time for the parallelizable portion of the code is expected to be in proportion to the number of active sites, which depends on the porosity and the size of the volume. But the time for the non-parallelizable portion of the code is likely to be dominated by the inter-process communication. Assuming that communication time is roughly proportional to the amount of data transferred, the communication time should be proportional to the number of active sites on an XY plane.

So as we process larger systems, the time for the parallelizable portion of the code should increase proportionally with the cube of the linear size of the system, while the non-parallelizable portion should increase with the square of the linear size of the system. This means that for larger systems, a larger proportion of the time is in the parallelizable computation, and greater benefits can be derived from running on multiple processors. We are still in the process of investigating the scaling of the software's performance with system size.

These performance data give us a general idea of how long it takes to get practical results for real-world problems on the computing platforms tested. For example, a typical case requires about 10000 iterations to converge. So from the performance described above, a one-component run of the sample size and porosity (22 %) described above will take about 41 hours on one processor on an SGI Onyx. On four processors, the same run will take approximately 10.6 hours. Approximate times for other sizes and porosities are easily calculated from the data above.

9. CONCLUSION

Lattice Boltzmann methods for simulating fluid flow in complex geometries have developed rapidly in recent years. The LB method produces accurate flows and can accommodate a variety of boundary conditions associated with fluid-fluid and fluid-solid interactions. With the advent of large memory/parallel workstations (or PC clusters), computations on fairly large systems that were considered beyond the reach of even some "super" computers from a decade ago can now be considered routine. Further work is needed to model fluids with large viscosity and density mismatches, which is the subject of current research.

ACKNOWLEDGMENTS

Fontainebleau sandstone images were prepared by John Dunsmuir of Exxon Research & Engineering Co. in collaboration with Brent Lindquist and Teng-Fong Wong (SUNYSB) at the National Synchrotron Light Source, Brookhaven National Laboratory, which is supported by the U.S. Department of Energy, Division of Materials Sciences and Division of Chemical Sciences under contract number DE-AC02-98CH10886.

DISCLAIMER

Certain commercial equipment and software may be identified in order to adequately specify or describe the subject matter of this work. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment or software is necessarily the best available for the purpose.