MULTISCALE MODELING OF FLUID TRANSPORT IN HETEROGENEOUS MATERIALS USING DISCRETE BOLTZMANN METHODS

Nicos S. Martys
Building and Fire Research Laboratory
100 Bureau Drive Stop 8615
National Institute of Standards and Technology
Gaithersburg, MD 20899-8615 USA

John G. Hagedorn
Information Technology Laboratory
100 Bureau Drive Stop 8911
National Institute of Standards and Technology
Gaithersburg, MD 20899-8911 USA

ABSTRACT

The lattice Boltzmann method is a promising approach for modeling single and multicomponent fluid flow in complex geometries like porous materials. Here, we review some of our previous work and discuss some recent developments concerning fluid flow in multiple pore size materials. After presenting some simple test cases to validate the model, results from large scale simulations of single and multicomponent fluid flow through digitized Fountaine sandstone, generated by X-Ray microtomography, are given. Reasonably good agreement was found when compared to experimentally determined values of permeability for similar rocks. Finally, modification of the lattice Boltzmann equations, to describe flow in microporous materials, is described. The potential for modeling flows in other microstructures of interest to concrete technology will be discussed.

INTRODUCTION

Diffusive and moisture transport in porous materials like ceramics, concrete, soils, and rocks plays an important role in many environmental and technological processes [1]. For example, the service life and durability of concrete can depend on the rate of ingress of chloride ions while the diffusion of carbon dioxide controls the rate of carbonation of the cementitious matrix. Further, such processes depend on the degree of saturation of the porous medium. The detailed simulation of such transport phenomena, subject to varying environmental conditions or saturation, is a great challenge because of the difficulty of modeling fluid flow in random pore

geometries and the proper accounting of the interfacial boundary conditions. In this paper, we will review [2] some recent advances in the modeling of fluid flow in complex geometries using the discrete Boltzmann methods. Discrete or lattice Boltzmann methods (LB) have emerged as a powerful technique for the computational modeling of a wide variety of complex fluid flow problems including single and multiphase flow in complex geometries. These methods naturally accommodate 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. Indeed, discrete Boltzmann methods serve as an ideal mesoscopic approach that bridges microscopic phenomena with the continuum macroscopic equations. Further, it can be directly implemented as a numerical method to model the time evolution of such systems. Finally, the LB method generally needs nearest neighbor information, at most, so is well suited to take advantage of parallel computers.

An outline of the paper goes as follows. After a brief review of the theory of the LB method, results are presented to validate predictions of fluid flow through a few simple pore geometries. Large scale simulations of fluid flow through a Fontainebleau sandstone microstructure, generated by X-ray microtomography, will then be presented. Single phase flow calculations were carried out on systems containing 510^3 computational elements. We also calculate relative permeability curves as a function of fluid saturation and driving force. The next section describes solution of the Brinkman equation using a lattice Boltzmann based approach. Finally, a comparison of the performance of such codes on different computational platforms is presented.

A LATTICE BOLTZMANN MODEL OF MULTICOMPONENT FLUIDS

The LB method of modeling fluid dynamics is actually a family [3] 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 are corrected for various deficiencies. 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, with 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 paper we use the D3Q19 (3 Dimensional lattice with $b = 19$) lattice[4,5]. 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:

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

where Ω_a^i is the collision operator representing the rate of change of the particle distribution due to collisions and g_a^i is body forcing term. The collision operator is greatly simplified by use of the single time relaxation approximation[6,7]

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

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 [5,8]:

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

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

where

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

S is the number of fluid component, m^i is the molecular mass of the ith component, $t_a = 1/36$ for $1 \le a \le 12$, $t_a = 1/18$ for $13 \le a \le 18$ 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 $k_b = 1$).

The above formalism leads to a velocity field that is a solution of the Navier-Stokes [7] 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 [8].

Phase Separation of Fluids

There are a variety of approaches to modeling the phase separation of fluids [3,9]. In the Shan-Chen model, a force, $\frac{d\mathbf{p}^i}{dt}(\mathbf{x})$, between the two fluids is introduced that effectively perturbs the equilibrium velocity [4,5] for each fluid so that they have a tendency to phase separate:

$$n^i(\mathbf{x})\mathbf{v}'(\mathbf{x}) = n^i\mathbf{v}(\mathbf{x}) + \tau_i\frac{d\mathbf{p}^i}{dt}(\mathbf{x}) \tag{7}$$

where $\mathbf{v}'$ is the new velocity used in Eqs. [3] and [4]. A simple forcing that depends on the density of each fluid, is as follows [4,5]:

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

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. Clearly, the forcing term is related to the density gradient of the fluid. It has been shown that the above forcing term can drive the phase separation process and naturally produce an interfacial surface tension effect consistent with the Laplace law boundary condition [5].

Phase separation of fluid can also be modeled by directly incorporating the force, $\frac{d\mathbf{p}^i}{dt}(\mathbf{x})$, into the body force term. First note that in the continuum Boltzmann equation, the body force term is written $\mathbf{a}\cdot\nabla_e n(\mathbf{x},\mathbf{e})$, where $\mathbf{a}$ is an acceleration field due to a body force. A representation [10] of this body force term to second order in Hermite polynomials, in the discrete velocity space of the D3Q19 lattice is

$$g_a = -3t_a n(\mathbf{x})\left[(\mathbf{e_a}-\mathbf{v})\cdot\mathbf{a} + 3(\mathbf{e_a}\cdot\mathbf{v})(\mathbf{e_a}\cdot\mathbf{a})\right] \tag{9}$$

To first order, the body force term is written as $g_a = -3t_i n(\mathbf{x})\mathbf{e_a}\cdot\mathbf{a}$.

In both models, phase separation takes place when the mutual diffusivity of the binary mixture becomes negative. An analytical expression for the mutual diffusivity has been determined in a previous work [8]. For the case of a critical composition, the condition for the system studied to undergo phase separation is $G \geq \frac{T}{12(n^1+n^2)}$.

IMPLEMENTATION

The approach to implementation of the algorithm is relatively straightforward. At each active site is 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 (n^i_a) 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.

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.

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) [11]. 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.

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 [1], $\langle \vec{v} \rangle = -\frac{k}{\mu} \langle \nabla P \rangle$, where $\langle \vec{v} \rangle$ is the average flow rate, ∇P is the average 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 the pore space around a simple cubic array of solid spheres that are allowed to overlap for large enough radius (i.e. when the solid fraction, c, exceeds $c \approx 0.5236$). In Figure 2, we compare our simulation data with that of Chapman and Higdon[12], 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 we used digitized spheres, while Chapman and Higdon used smooth spheres.

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 5.72 μm per lattice spacing and data sets were 510^3 voxels in size. A mirror image boundary condition was applied along directions perpendicular to the applied force. 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

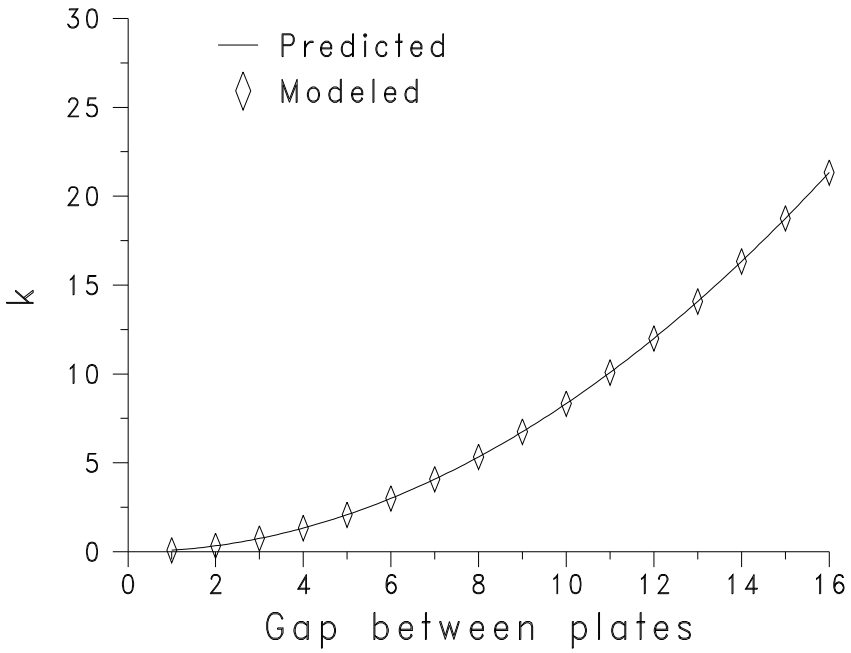

Figure 1: Flow through parallel plates. The permeability, k, is in units of lattice spacing squared, while the gap between plates is in units of lattice spacing.

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 4 shows the computed permeability compared to experimental data [13]. Clearly there is good agreement, especially at the higher porosities.

RELATIVE PERMEABILITY

We next present a sample calculation of the relative permeability for the 22 % porosity Fontainebleau sandstone. In this case, the pore space is filled with two fluids. One fluid preferentially wets the solid surface and the second fluid is non-wetting. The degree of saturation, Θ_w is V_w/V_p, where the V_w is the volume of the wetting phase in the pore space and V_p is the volume of the pore space. Although there is debate as to the correct formulation of the macroscopic two phase flow equations [14], we use the following empirical relation to describe the response of a multiphase fluid system to an external driving force:

$$\vec{v}_1 = -\frac{K_{12}}{\mu_2}\nabla P_2 - \frac{K_{11}}{\mu_1}\nabla P_1 \tag{10}$$

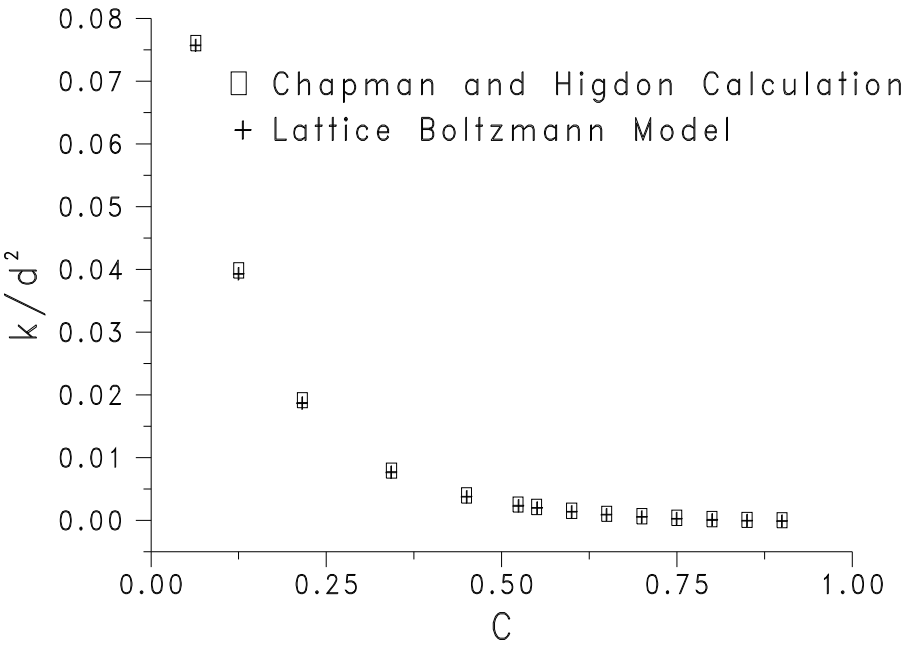

Figure 2: Flow through spheres centered on a simple cubic lattice. The permeability is normalized by the square of the distance, d, between the sphere centers.

$$\vec{v}_2 = -\frac{K_{21}}{\mu_1}\nabla P_1 - \frac{K_{22}}{\mu_2}\nabla P_2 \tag{11}$$

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 average velocity of each fluid component is given by $\vec{v}_1$ and $\vec{v}_2$. 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 $C_a^* = \frac{\mu <v>}{\gamma} = \frac{k\rho g}{\gamma}$. Figure 5 shows the relative permeability of the $\phi = 22$ % rock for the cases of $C_a^* = 7.5 \times 10^{-4}$ and 7.5×10^{-5}.

APPLICATION TO CEMENT BASED MATERIALS

LB algorithms are applicable to a wide variety of microstructures associated with cement based materials including cement paste, mortar, and concrete. For example, we have studied fluid flow in fractured mortars. A series of X-ray microtomography based images of fractured mortars as a function of applied strain have been

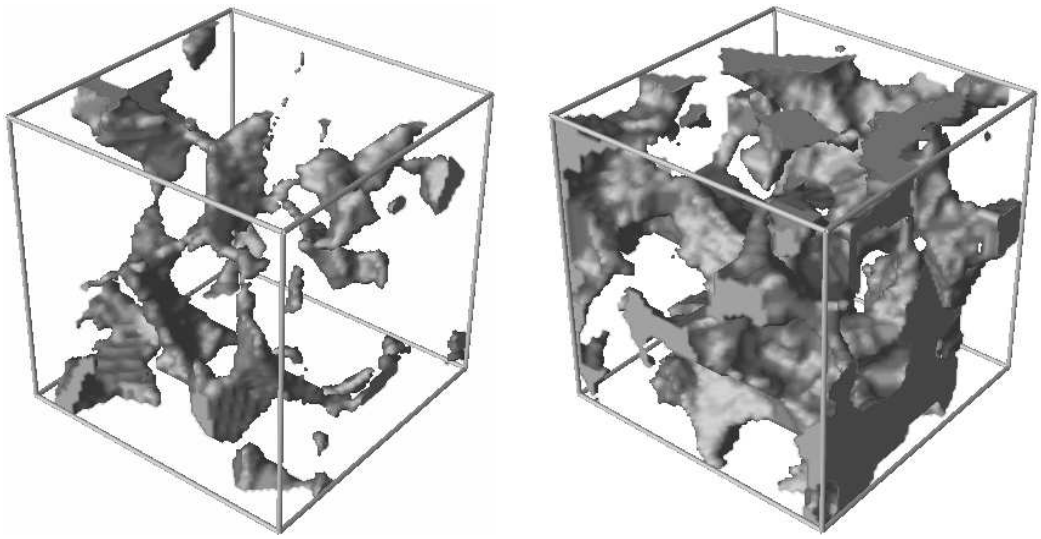

Figure 3: 64×64 portions of the Fontainebleau sandstone media. On the left is the 7.5 % porosity medium, on the right is the 22 % porosity medium. The solid matrix is made transparent to reveal the pore space (grey shaded region).

obtained by Landis and Keane [15]. Figure 6 shows a typical fracture image. The image used in the flow simulation was 200^3 voxels in size with lattice units equal to $6.0 \cdot 10^{-6}$ m. For this system, we found $k = 1.35 \cdot 10^{-11}$ m^2. Clearly, given the very low permeability of the uncracked mortar ($k \approx 10^{-18}$ m^2), nearly all fluid flow should be through the crack.

Solution of the Brinkman Equation: Multiple Scale Porous Media

Modeling fluid flow in porous heterogeneous materials with more than one typical pore size (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:

$$\nabla p = \mu \nabla^2 \mathbf{v} \tag{12}$$

$$\nabla \cdot \mathbf{v} = 0 \tag{13}$$

where p is the pressure, $\mathbf{v}$ is the fluid velocity and μ is the fluid viscosity. Regions

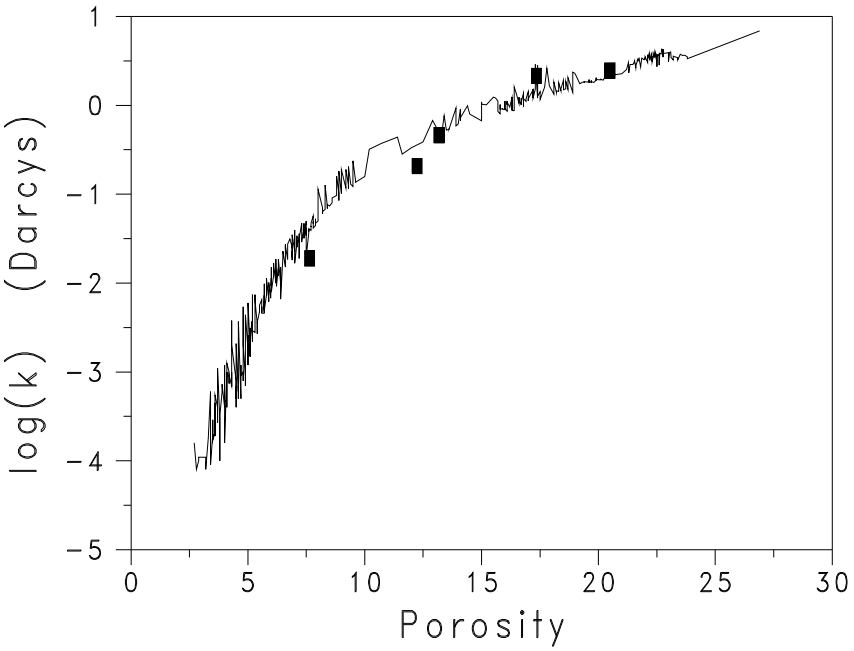

Figure 4: Measured (line) and modeled (diamonds) permeabilities of Fontainebleau sandstone medium.

with the smaller pores are treated as a permeable medium and flow is described by Darcy's law. The two boundary conditions to be satisfied at the pore/permeable medium interface are continuity of the fluid velocity and the shear stress [16,17]. Darcy's law alone is not sufficient to satisfy these boundary conditions. The Brinkman equation [17] 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

$$\langle \nabla p \rangle = -\frac{\mu}{k}\mathbf{v} + \mu_e \nabla^2 \langle \mathbf{v} \rangle \tag{14}$$

where $\mathbf{v}$ is the fluid velocity, μ is the fluid viscosity, and μ_e is an effective viscosity parameter. The so-called effective viscosity should not be thought of as the viscosity of the fluid but only a parameter that allows for matching of the shear stress boundary condition 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 for this example. The $+$ and $-$ refer to regions in the free-fluid and porous medium, respectively.

Although the Brinkman equation is semi-empirical in nature, it has been validated by detailed numerical solution of the Stokes' equations in regions near the interface between dissimilar regions [18]. Numerical solution of the Brinkman equa-

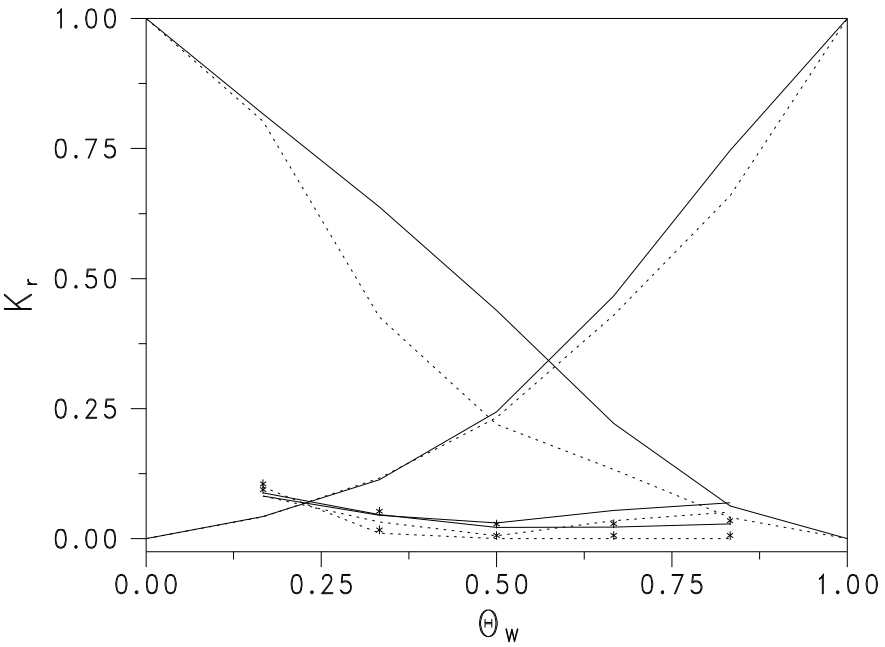

Figure 5: Relative permeabilities of 22 % porosity Fontainebleau sandstone versus wetting fluid saturation, Θ_W. The solid and dashed lines correspond to $C_a^* = 7.5 \times 10^{-4}$ and $C_a^* = 7.5 \times 10^{-5}$ respectively. The lower curves correspond to the off-diagonal elements of the permeability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

tion by more traditional computational methods (e.g. finite difference and finite element) is certainly possible. However, a recent lattice Boltzmann (LB) based model by Spaid and Phelan [19], along with recent improvements [20], has proven to be a simple and computationally efficient method to numerically approximate fluid flow described by the Brinkman equation.

To produce flow consistent with the Brinkman equation, a dissipative forcing $\mathbf{F} = -\frac{\mu \mathbf{v}}{k}$ is used. Originally, this forcing was incorporated into a LB model, normally used to approximate the Navier Stokes equations, by introducing a velocity shift, $\Delta \mathbf{v} = \tau \mathbf{F}/n$, ($\tau$ is a relaxation parameter and n is the density) in the Boltzmann equilibrium distribution according to the method of Shan and Chen [4]. However it is well known that this approach will produce errors of order $\tau^2 F^2$ in the pressure tensor [20]. Such errors can have a significant impact on the fluid dynamics of such systems. Hence, it can be advantageous to instead apply the force in the body force term.

To first validate this model, a simple Couette flow geometry was used (see Figure 7). Starting with a parallel plate geometry, a permeable medium is positioned

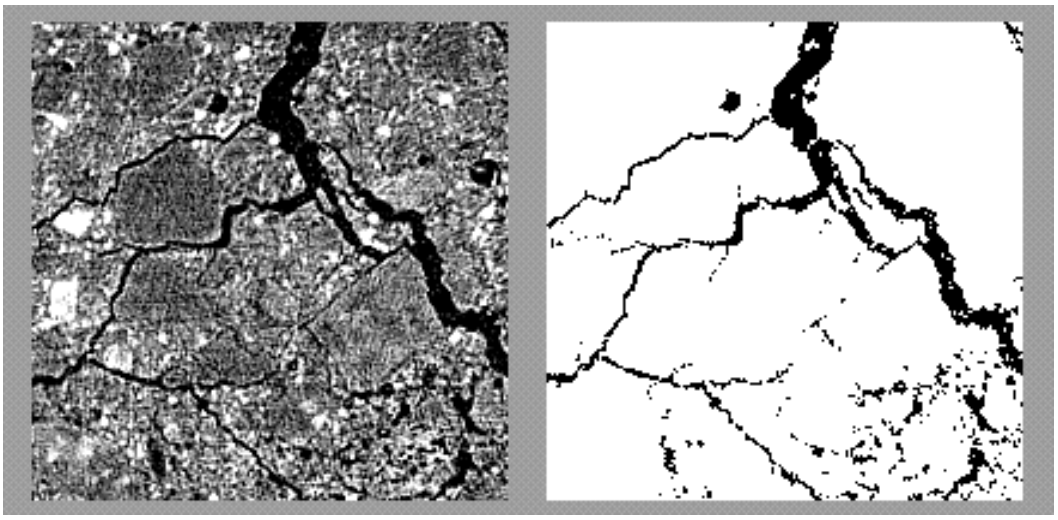

Figure 6: Single slice from an X-Ray microtomography based image of a fractured mortar. The image on the left is based on the original data set. The image on the right was produced by processing the original data set to distinguish between pore (black) and solid (white). The fluid flow was calculated throughout the pore region. The X-ray microtomography image was obtained by Eric Landis using the National Synchrotron Light Source, Brookhaven National Laboratory.

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 $\sqrt{\frac{\mu_e}{\mu}}$ [16]. In Figure 7, 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 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.

For this test case, $k = 1/11$ in units of lattice spacing squared. Such a choice of k, ignoring tortuosity effects, corresponds to a porous medium with a typical pore size of order a lattice spacing as can be seen by noting that the permeability associated with a cylindrical tube is $k = r^2/8$ where r is the tube radius.

We next consider solution of the Brinkman equation using the Fontainebleau sandstone, described earlier, as a porous medium where the solid phase is now

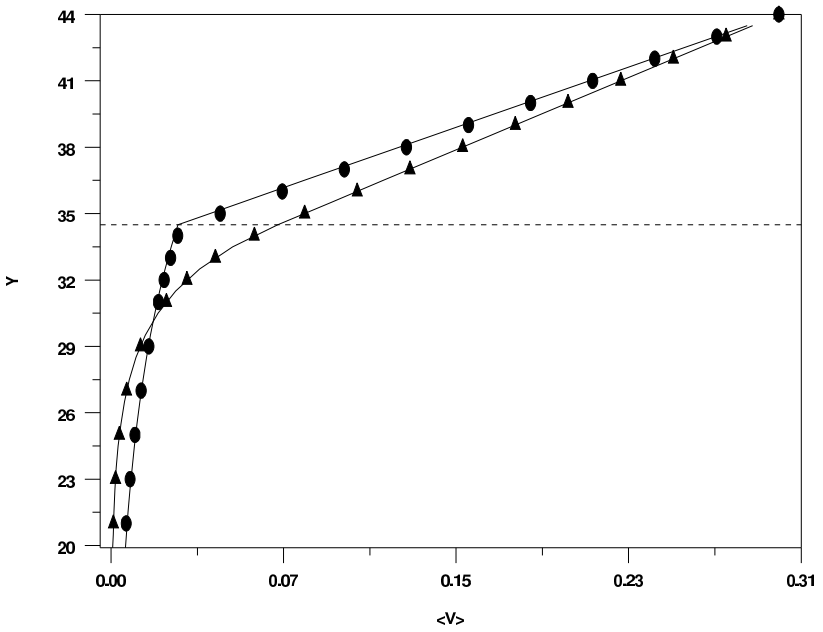

Figure 7: Velocity field of a sheared system next to a porous medium. The filled triangles and circles represent data from the lattice Boltzmann simulation using ($\mu_e/\mu = 1$ and $\mu_e/\mu = 4$, respectively). The solid lines are analytic solutions of the Brinkman equation. The region below the dashed line $y = 34.5$ (in units of lattice spacing) corresponds to a porous medium. The moving wall is at $y = 44$.

assigned a permeability k_s. Although this may not be the case for the original rock, the sandstone image serves as a convenient "random" pore structure to use. Four different permeable media were used with porosity ranging from about 7 % to 40 %. Here, the porosity refers to the original pore structure. In Figure 8, we plot the bulk permeability, k_b, of the overall system vs k_s. Clearly, k_b increases with k_s. At lower values of k_s, the higher porosity system appears to be less sensitive to k_s as a result of the the larger pores carrying most of the flow. Fluid flow in the lower porosity systems are much more influenced by k_s, as a larger fraction of the system is composed of the permeable medium. It should also be pointed out that an alternate version [21] of the lattice Boltzmann method was used to determine results for $k_s < 0.1$ as the above described model is unstable in this regime. The instability is a result of the fixed time step used in the LB method. The alternate version allows for introducing smaller time steps so that the instability is avoided. The reader is referred to [21] for more details.

PERFORMANCE RESULTS

We ran a series of timing tests in an effort to understand how performance of our

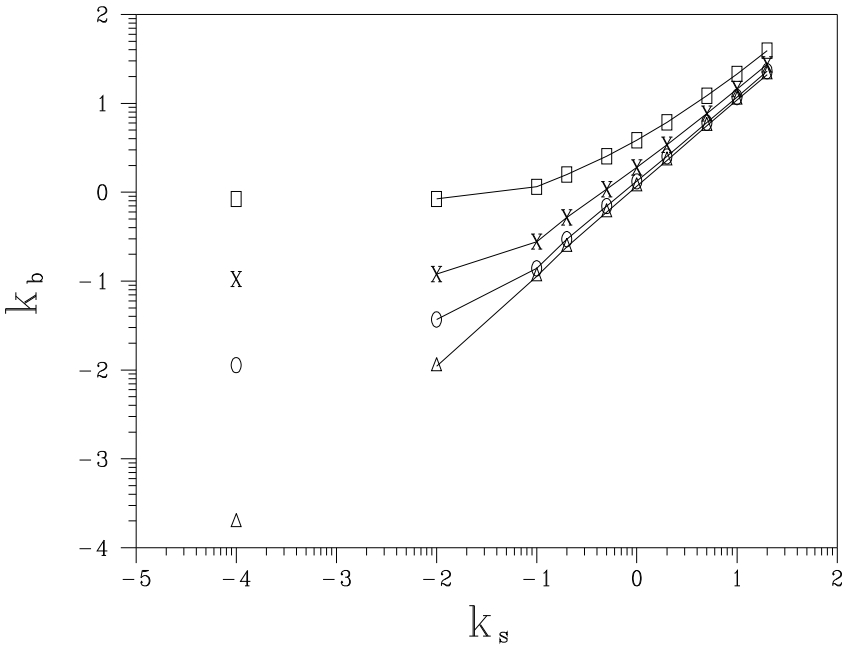

Figure 8: Bulk permeability, k_b vs permeability assigned to the normally solid portion of the microstructure. The curves (top to bottom) correspond to microstructres whose initial porosity was 40 % (squares), 22.5 % (Xs), 13.0 % (circles), and 7.5 % (triangles). The isolated data points on the left represent the case where $k_s = 0$.

implementation scales on different computer architectures. We have tested on an SGI Onyx with 12 R10000 processors running at 196 MHz, and an IBM SP2 with 37 RS/6000 processors, most running at 66 MHz. The same code and the same cases were run on the two systems. The results are presented in Tables I and II. 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 this effect.

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

# 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 II. Execution time in seconds for one iteration on the IBM SP2.

# 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

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: $T = 4.78 + 487.26/N$ s, where N is the number of processors. Similarly, the timings for the two component runs on the IBM SP2 closely match: $T = 41.67 + 1198.45/N$ s. Formulae for the other cases are easily derived. Figures 9 and 10 present these results graphically.

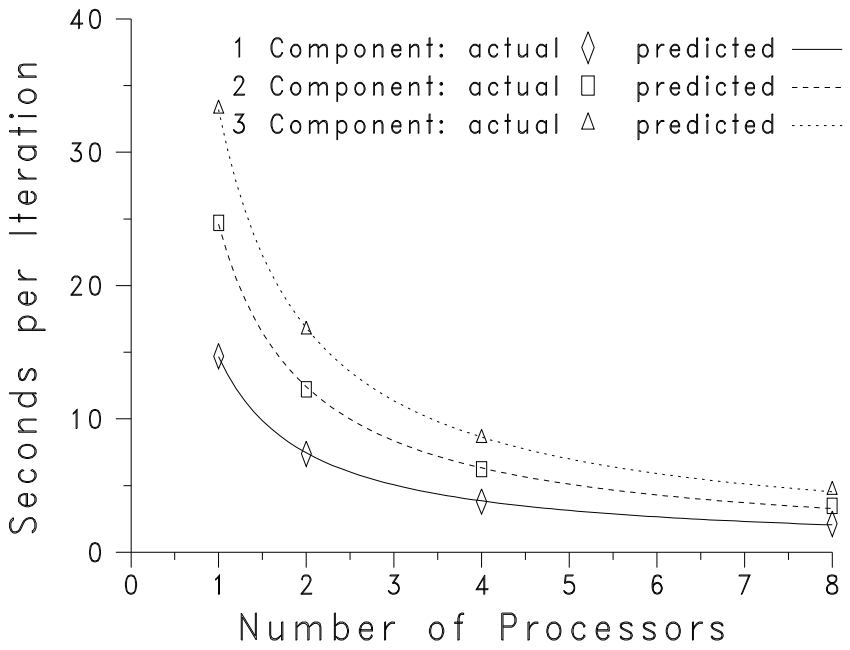

Figure 9: Time in seconds for one iteration on the SGI Onyx.

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 s on the SGI versus 41.67 s 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

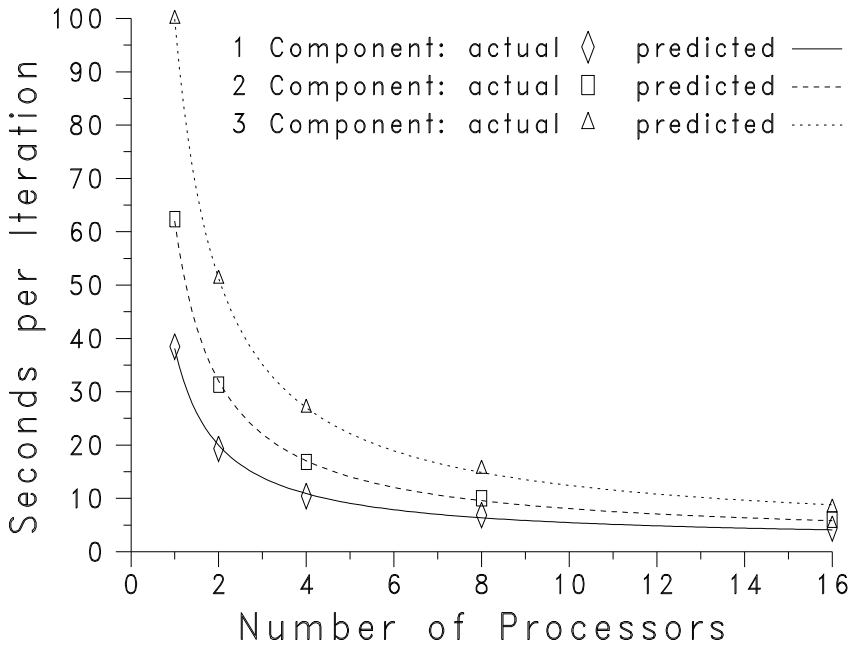

Figure 10: Time in seconds for one iteration on the IBM SP2.

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.

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.

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 h on one processor on an SGI Onyx. On four

processors, the same run will take approximately 10.6 h. Approximate times for other sizes and porosities are easily calculated from the data above.

CONCLUSIONS

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 Linux 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. We are clearly in a good position to study fluid flow in a variety of microstructures relevant to concrete technology.

ACKNOWLEDGEMENTS

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.

REFERENCES

[1] F.A.L. Dullien, *Porous Media Fluid Transport and Pore Structure,* 2nd ed. (Academic Press, Inc., San Diego 1992).

[2] Most of the results presented in this paper originally appeared in the following paper: N. S. Martys, J. G. Hagedorn and J. E. Devaney, "Pore scale modeling of fluid transport using discrete boltzmann methods," Materials Science of Concrete, Ed. R. D. Hooten, M.D.A. Thomas, Jacques Marchand, James J. Beaudoin (2001).

[3] D. H. Rothman and S. Zaleski, "Lattice-gas model of phase separation: interfaces, phase transitions, and multiphase flow", *Rev. Mod. Phys.*, **66**[4], 1417-1479 and references, 1998.

[4] X. Shan and H. Chen, "A lattice Boltzmann model for simulating flows with multiple phases and components", *Phys. Rev. E*, **47**, 1815-1819, 1993.

[5] N. S. Martys and H. Chen, "Simulation of multicomponent fluids in complex three-dimensional geometries by the lattice Boltzmann method", *Phys. Rev. E*, **53**, 743-750, 1996.

[6] Y. H. Qian and D. d'Humières and P. Lallemand, "Lattice BGK models for Navier-Stokes equation", *Europhys. Lett.*, **17**, 479-484, 1992.

[7] H. Chen and S. Y. Chen and W. H. Matthaeus, "Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method", *Phys. Rev. A*, **45**, R5339-R5342, 1992.

[8] X. Shan and G. Doolen, "Diffusion in a multicomponent lattice Boltzmann equation model", *Phys. Rev. E*, **54**, 3616-3620, 1996.

[9] N. S. Martys and J. F. Douglas, "Critical properties and phase separation in lattice Boltzmann fluid mixtures", Phys. Rev. E, **63**, 31205 (2001).

[10] N. S. Martys, X. Shan, and H. Chen, "Evaluation of the external force term in the discrete Boltzmann equation," Phys. Rev. E, **58**, 6855 (1998).

[11] Message Passing Interface Forum, "MPI: A Message-Passing Interface Standard", *International Journal of Supercomputing Applications*, **8**[3/4], 1994.

[12] A. M. Chapman and J. J. L. Higdon, "Oscillatory Stokes Flow in Periodic Porous Media", *Phys. Fluids A*, **4**[10], 2099-2116, 1992.

[13] T. Bourbie and B. Zinszner, "Hydraulic and Acoustic Properties as a function of Porosity in Fontainebleau Sandstone", *J. Geophys. Res.*, **90**[B13], 11,524-11,532, 1985.

[14] P. A. Goode and T.S. Ramakrishnan, "Momentum Transfer Across Fluid-Fluid Interfaces in Porous Media: a Network Model", *AIChE Journal*, **39**[7], 1124-1134, 1993.

[15] E. N. Landis and D. T. Keane, "X-ray microtomography for fracture studies in cement-based materials," Proceedings of SPIE, Developments in X-Ray Tomography II, ed. U. Bonse, 3772, 1999.

[16] J. Koplik, H. Levine, and A Zee, "Viscosity renormalization in the Brinkman equation," Phys. Fluids **26**, 2864 (1983).

[17] H. C. Brinkman, "A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles," Appl Sci. Res. A **1**, 27 (1947).

[18] N. Martys, D.P. Bentz, and E.J. Garboczi "Computer simulation study of the effective viscosity in Brinkman's equation," Phys. Fluids **6**, 1434 (1994).

[19] M. A. A. Spaid and F. R. Phelan, "Lattice Boltzmann methods for modeling microscale flow in fibrous porous media," Phys. Fluids **9**, 2468 (1997).

[20] N.S. Martys, "Improved approximation of the Brinkman equation using a lattice Boltzmann method," Physics of Fluids, Vol. 13, No. 6, 1807-1810, (2001).

[21] N. S. Martys, N.S., "A Classical Kinetic Theory Approach to Lattice Boltzmann Simulation," Int. J. Mod. Phys. C., 12 (8), 1169-1178 (2001).