CHAPTER 13

LATTICE BOLTZMANN METHOD FOR CALCULATING FLUID FLOW AND DISPERSION IN POROUS AND FRACTURED MEDIA

HARLAN W. STOCKMAN
Sandia National Laboratories, P.O.Box 5800, Albuquerque, NM, 87185, USA

13.1 INTRODUCTION

The lattice Boltzmann (LB) method is a numerical technique suited for modeling flow of fluids, particularly those obeying the Navier-Stokes and advection-dispersion equations (Rothman and Zaleski 1997; Martys and Chen, 1996). The strength of the method lies in its ability to calculate transport in the presence of geometrically complex solid boundaries and initial conditions. Added effects may include chemical reaction, dissolution and precipitation, immiscibility, and buoyancy.

In the past ten years, the LB method has been applied increasingly to problems involving flow through porous and fractured rocks and soils (Stockman et al., 1997; Stockman et al., 1998; Zhang et al., 2002; Pan et al., 2004; Sukop and Or, 2004, and references therein). While the technique can be applied to most fluids, there are some special considerations for LB models of gases, and in particular, for assemblages of many pores, or macroscopic fractures. These special considerations, along with illustrative examples and benchmarks, are the subject of this chapter. The focus of the chapter is on pore-scale models, where fluid behavior is approximated by the Navier-Stokes equations. These pore-scale models are used to obtain permeability and dispersion coefficients applicable to a larger scale, and the inherent and practical limitations of the method are illustrated with examples.

13.2 METHOD

13.2.1 Basic Method

In the LB method, physical space is broken up into a set of nodes, usually on a Cartesian grid; the set of these nodes is called the automaton. The term *lu* is used for

C. Ho and S. Webb (eds.), Gas Transport in Porous Media, 221–242.

lattice unit, the fundamental unit of distance in the automaton (the distance between adjacent nodes is one *lu*); *ts* refers to one time step in the automaton. Mass is allowed to move about the grid, but via a fixed set of n velocity vectors $\vec{\mathrm{e}}_i$ (n is typically 15 or 19 for 3D LB). At each node, there is set of n floating-point (non-integral) particle distribution functions f_i, corresponding to each ith velocity. The mass density at each node is $\rho = \sum_{i=0}^{n-1} f_i$. The evolution of the particle distribution function f_i with time t,is calculated according to

$$f_i(\vec{\mathrm{r}} + \vec{\mathrm{e}}_i,\, t+1) = f_i(\vec{\mathrm{r}}, t) \; - \; \omega \cdot (f_i(\vec{\mathrm{r}}, t) \; - f_i^{\mathrm{eq}}(\vec{\mathrm{r}}, t)) \tag{13.1}$$

where $\vec{\mathrm{r}}$ is a node position (x, y, z) in the lattice, f_i^{eq} is the Boltzmann equilibrium distribution, and ω is the collision operator, which determines the viscosity and diffusion coefficients. With a carefully chosen form for f_i^{eq}, the $\rho(x, y, z)$ will obey the incompressible Navier-Stokes equation for fluid flow. There are several ways to add momentum "kicks" with each step, so the fluid will accelerate, perhaps reaching a viscous equilibrium. In this chapter, it is assumed that all LB calculations use some variant of the BGK (Bhatnagar-Gross-Krook) method, where the collision operator is simply a constant $1/\tau$ (Qian et al., 1992). It is important to understand that LB does not presume to be a model of molecular behavior; while one refers to distribution functions, and there are allusions to translations of particles, the distributions are merely convenient artifices for tracking the movement of mass and momentum.

In typical LB computer programs, a translation or streaming function performs the left side of Eq. (13.1), and a collision function performs the right side. Thus, Eq. (13.1) is really carried out in two steps:

1. At each node, an equilibrium distribution f_i^{eq} is calculated for each vector direction. The distribution of f_i at that node is then replaced with a new distribution $f_i - (f_i - f_i^{\mathrm{eq}})/\tau$.
2. The new f_i are then translated, or "streamed," along the vector directions to the adjacent nodes. This process is handled by the computer in such a way that none of the new values are accidentally overwritten.

A velocity field is calculated at each x, y, z site as

$$\vec{\mathrm{u}} = (u_x, u_y, u_z) = \sum_{i=0}^{n-1} f_i \cdot \vec{\mathrm{e}}_{\mathrm{i}} \tag{13.2}$$

$$u^2 = \vec{\mathrm{u}} \cdot \vec{\mathrm{u}} \quad \text{(a scalar)} \tag{13.3}$$

In this chapter, we use the 19-vector D3Q19 lattice (Qian et al., 1992). This lattice has one "rest" particle with a velocity vector $\mathrm{e}_0 = (0, 0, 0)$, six velocity vectors along the Cartesian axes (permutations of $(\pm 1, 0, 0)$) and twelve vectors that are permutations of $(\pm 1, \pm 1, 0)$. For this lattice, the equilibrium distribution is calculated as (Martys and Chen, 1996):

$$f_0^{\text{eq}} = \frac{\rho}{3}\left[1 \;-\; \frac{3}{2}u^2\right] \tag{13.4}$$

$$f_i^{\text{eq}} = q_i\,\rho\;\left[1 \;+\; 3\,\vec{\text{e}}_i \cdot \vec{\text{u}} \;+\; \frac{3}{2}((3\vec{\text{e}}_i \cdot \vec{\text{u}})^2 \;-\; u^2)\right] \tag{13.5}$$

where $q_i = 1/18$ for the six vectors along the Cartesian axes, and $q_i = 1/36$ for the remaining vectors. This distribution, when used in Eq. (13.1), will give back a very good approximation of incompressible Navier-Stokes behavior for $\vec{\text{u}}$.

13.2.2 Dispersion

To model dispersion of dilute tracers, the simplest approach is to use one lattice for a "carrier fluid" that obeys the Navier-Stokes equation, and calculate $\vec{\text{u}}$ for that carrier fluid only. A separate lattice exists for each tracer, and changes in the tracer distributions are propagated according to Eqs. (13.1), (13.2), (13.4) and (13.5), using the $\vec{\text{u}}$ determined from the carrier.

Unlike the Navier-Stokes equations, the advection-dispersion equation (which governs tracer movement) is linear in $\vec{\text{u}}$. Flekkøy et al. (1995) and Noble (1996) reasoned that the tracer equilibrium distributions should be linear in $\vec{\text{u}}$ as well. Thus for a tracer s, the equilibrium distribution is of the form:

$$f_{s,i}^{\text{eq}} = A + B \cdot (\vec{\text{e}}_{s,i} \cdot \vec{u}) \tag{13.6}$$

and the A and B are fixed by the requirement that solute is conserved ($\rho_s = \sum f_{s,i}^{\text{eq}}$), and the requirement that the solute flux at equilibrium is due entirely to advection ($\sum f_{s,i}^{\text{eq}}\vec{\text{e}}_{s,\,i} = \rho_s \cdot \vec{\text{u}}$). Compared to Eq. (13.5), Eq. (13.6) requires fewer floating point operations to evaluate the equilibrium distribution, hence is faster to compute.

The real bottleneck in the LB calculation, is the time taken to read all the vectors f_{i} from memory (for both carrier and tracers), and to write them back after they have been updated. Thus a great deal of memory traffic can be avoided by using lattices with fewer vectors for the tracers. In 2D, only four Cartesian vectors are needed, and in 3D, the six Cartesian vectors will suffice. Wolf-Gladrow (1994) suggested low-vector lattices for modeling diffusion, and Noble (1996) performed a Chapman-Enskog expansion for Cartesian lattices, recovering the advection-dispersion equation and estimating the error. The present author compared the 6-vector and 19-vector methods for tracer problems in extreme conditions and found the agreement was extremely close. (It must be emphasized that low-vector lattices, and Eq. (13.6), are used *only* for the dilute tracers, not the carrier.)

In many dispersion problems, the flow field $\vec{\text{u}}$ is at steady state, and need not be recalculated at each timestep. Furthermore, periodic boundary conditions on $\vec{\text{u}}$ are often appropriate, even when tracer dispersion is not periodic, as in the SC dispersion problem described in Section 13.3.2. For such problems, one can use a 19-vector carrier fluid in a single repeat unit, $N_x \times N_y \times N_z\, lu^3$, then save the equilibrium

$\vec{u}(x, y, z)$. Typically, the flow reaches steady-state at time $> L^2/\nu$, where ν is the kinematic viscosity of the carrier fluid, and L is a characteristic length scale perpendicular to the flow direction (e.g., the channel width). Then the memory used for the 19-vector carrier fluid can be reclaimed, and reallocated for additional repeat units of solid geometry (and the lattice used for the tracers) in the x direction. The velocity field at x,y,z is then $\vec{u}((x\,\mathrm{MOD}\,N_x),y,z)$. The lattice Boltzmann updates for the tracers are still carried out at each step, but rather than recalculate the velocity at each point, the saved velocity field is used. We refer to these additional repeat units as "clones."

As mentioned, the viscosity of the carrier fluid, and the diffusion coefficients of the tracers in the fluid, are entirely dependent on the collision parameter τ. In the 3D model above,

$$\nu = \frac{\left(\tau_0 - \frac{1}{2}\right)}{3} \quad \text{and} \quad D_{\mathrm{ms}} = \frac{\left(\tau_{\mathrm{s}} - \frac{1}{2}\right)}{3} \tag{13.7}$$

where D_{ms} is the molecular diffusion coefficient of tracer s, in units of lu^2/ts.

In one important way, the LB method is better suited to model dispersion in gas transport, than dispersion in liquids such as water. It is difficult to set up a dispersion calculation wherein the τ_0 of the bulk fluid, or carrier, is vastly different than the τ_{s} of the tracers, either for reasons of numerical stability or convenient scaling. In water, the $\nu \sim 0.01$ cm^2/s, but $D_{\mathrm{m}} \sim 10^{-5}$ cm^2/s, so there is a three-order of magnitude difference that must be spanned. But in gases, kinetic theory leads us to expect the kinematic viscosity and diffusion coefficients (for similar molecules) will be approximately the same. Thus ν for air is ~0.14 cm^2/s at 18 C and 1 atm, and D_{m} for CO_2, O_2 and H_2O in air are approximately ~0.14, 0.18 and 0.24 cm^2/s, respectively, at the same temperature and pressure. Hence, one is more likely to find a suitable ν/D_{m} for gases.

13.2.3 Boundaries

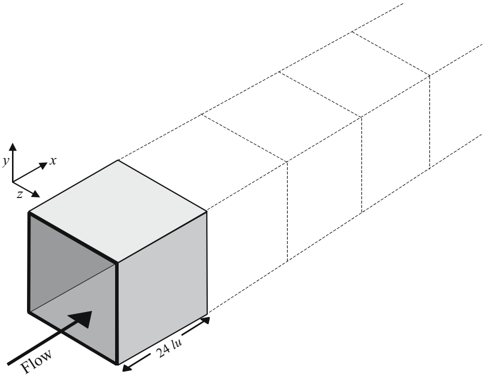
The equations and calculation method above describe the update of the free fluid. The processing at calculation boundaries and solid walls require special attention, to ensure mathematical "closure;" i.e., that the total mass of each component is conserved in the absence of intentional sinks. The calculation domain is finite in size; thus during the translation stage, the distributions that point out of the domain would be lost, unless a special effort were made to preserve that mass. A simple way to solve this dilemma is to use "wrap" boundaries, so that a vector leaving the right side of the automaton at $x_{\max}$ is reinjected at 0. Similarly, a vector that leaves the left side of the automaton at $x = 0$, is reinjected at $x_{\max}$, the last site in the x-direction. This wrap is performed automatically in the translation step, and applies to the y- and z-directions as well.

One must also decide how to treat solids during the collision step. The basic problem is this: before a collision step, the preceding translation step moves vectors from free fluid to solid sites, and these vectors point into the solid. Yet there may be no vectors pointing out of the wall into the fluid, since there is no fluid inside the solid before translation. A simple solution is to invert the directions of all vectors on the wall (for both the carrier fluid and each tracer); this approach is called the "bounce-back" boundary, and it assures that there is a no-slip velocity boundary near the wall (Chen et al., 1996). Section 13.4 will discuss the consequences of bounce-back in more detail.

If a solid boundary is placed at the ends of the automaton, it automatically supercedes the wrap condition, because the solid wall deflects all outward-pointing vectors back into the automaton.

Self Study Example:

We will simulate dispersion in a long duct, with square cross-section, by the method described in Section 13.2.2. The initial duct contains 24 nodes in the x-direction, and 26 in each of the y- and z-directions, and is open in the x-direction, with the other 4 sides closed off with solid walls, as shown below:



A x-direction body force is placed on the fluid, causing the fluid to accelerate along the x-axis. The viscosity ν is 0.01 lu^2/ts, and the average speed is limited to 0.01 lu/ts. The initial calculation begins with one 19-vector carrier fluid, to calculate the fluid velocity field. After 60000 ts, the memory from the flow calculation is reclaimed, the flow field is "frozen in," and the channel is cloned 47 additional times in the x-direction. The resultant (48 × 24) × 26 × 26 channel is used for a dispersion study with a single 6-vector tracer. A slug of tracer is injected into the left end of the channel, and is followed as it disperses in the flow; this part of the calculation proceeds for additional 60000 ts.

(1a) Are 60000 ts adequate to reach viscous equilibrium?

(1b) On your computer, the LB program processes a single 19-vector component at a rate of 9 million site-updates per second (MUPS), and processes a single 6-vector component (in dispersion-only mode) at 35 MUPS. How many seconds will be needed to complete the first (19-vector) and last (6-vector) parts of the calculation?

(1c) The calculation is performed in single precision, and each vector is represented by a 32-bit (4 byte) floating-point number. How much memory is required for the first (19-vector) and second (6-vector) parts of the calculation?

13.2.4 Efficient Algorithms

The LB method can be quite calculation-intensive, and is simply not efficient for most simple geometries. However, many of the problems experienced by newcomers to the technique probably stem from coding inefficiencies. The calculations reported herein have benefited from many speed optimizations in the source code. Some of these optimizations, such as disambiguation of pointers, have been discussed previously (Stockman et al., 1998). Since that publication, the code has been modified to use SIMD (single-instruction, multiple data) programming, along with explicit prefetch; these improvements resulted in ~3-fold improvement in performance. The compiler used for producing the executable code can make a great deal of difference. For example, we found a factor 3.5 speedup in going from the most efficient version of code compiled under gcc (the public domain compiler available through the GNU project), to code compiled with Intel's version 5.01 C. We found that the choice of operating systems (OS; e.g., Linux vs. Microsoft Windows) made no difference on the program speed, so long as the same version compiler was used on each OS. Currently, the code is limited by the speed with which it can read and write vectors to main memory, and performance is greatest on desktop computers with high-speed, multiple-channel memory. It is notable that we have seen little improvement in the code speed in 3.5 years, simply because the speed of memory access has not improved greatly in that time.

13.3 BENCHMARKS AND ACCURACY

The following benchmarks illustrate the bounds of accuracy of the method, and also provide background for the discussions in Section 13.5. Where necessary for clarification, *lu* and *ts* units will be converted into centimeters and seconds; however, most comparisons are made in dimensionless parameters. Of particular interest are the Reynolds number (Re), Peclet number (Pe), and Schmidt number (Sc), defined as:

$$\mathrm{Re} = U \cdot \frac{L}{\nu}, \qquad \mathrm{Pe} = U \cdot \frac{L}{D_{\mathrm{m}}}, \text{ and } \mathrm{Sc} = \frac{\mathrm{Pe}}{\mathrm{Re}} \tag{13.8}$$

where U is a characteristic speed and L is a characteristic length scale, defined for the particular problem at hand. Usually L is the width or a particle or a pore.

(2) The D_{m} for the self-study problem is 0.01 lu^2/ts. What are the Re, Pe, and Sc for the channel calculation in Part 1?

13.3.1 Permeability of an Array of Spheres

The first test is calculation of permeability in a non-trivial geometry; in this case, in an infinite array of spheres in a simple cubic (SC) geometry (Zick and Homsy, 1982).

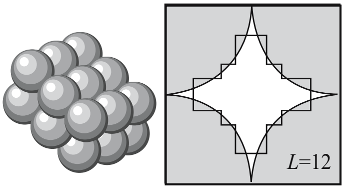


Figure 13.1. Left: part of an array of simple cubic (SC) spheres. Right: the ideal shape of the pore space between spheres (thin smooth curves) and the actual pixelation (black solid line) for a calculation with $L = 2R = 12$ lattice units

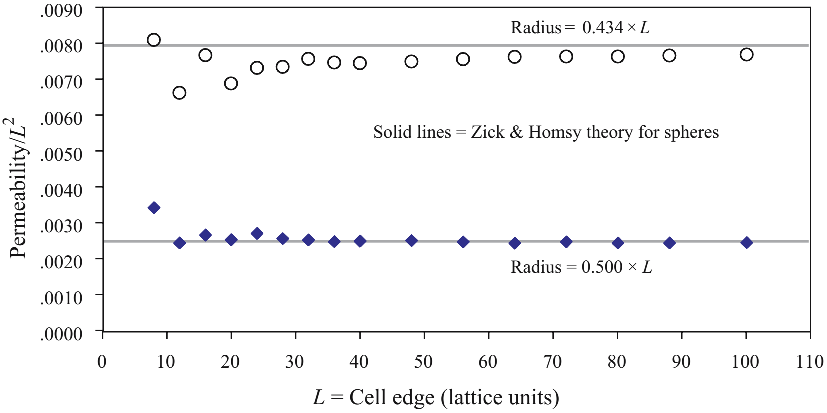


Figure 13.2. LB estimates of permeability for infinite SC array of spheres (Figure 13.1) as symbols, and Zick and Homsy (1982) theoretical values as solid horizontal lines

A portion of the geometry is shown in Figure 13.1. The LB simulation uses wrap boundaries on the faces of the automaton, so only one sphere need be considered in the calculation, as the wrap boundaries effectively produce an infinite array of spheres in the x, y, and z-directions. Figure 13.1 also shows a potential problem; in the relatively small LB simulations, the narrow channel between adjacent spheres is "pixelated" by the limited number of nodes ($12 \times 12 \times 12$ in this case), and is not well-resolved.

Figure 13.2 shows the LB calculated permeability, plotted against the separation L between the spheres (in *lu* on a side), for the case where the sphere radius $R = 0.5 \cdot L$ (touching) and the case $R = 0.434 \cdot L$ (slightly separated spheres), and Re ~ 0. The Zick and Homsy values are shown as solid lines; the agreement is fairly good, and the measured permeabilities converge on the "true" values as the number of nodes in the simulation increases. Figure 13.3 shows the correlation of the porosity with the permeability, for the LB simulations with $R = 0.5 \cdot L$; the target porosity is 0.476. The deviations from the target porosity result from the pixelation of the LB model, and are reflected in the errors in estimated permeability.

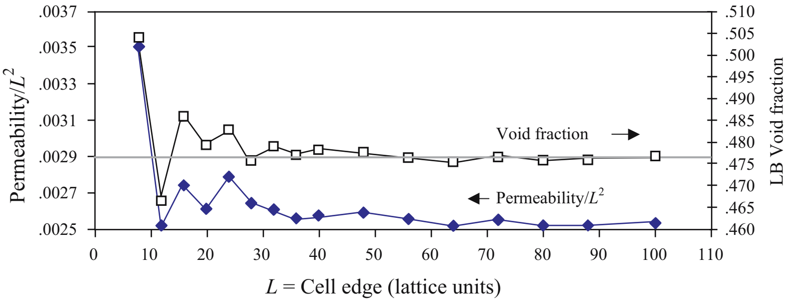


Figure 13.3. Correlation of void fraction in LB (ideal value = 0.476) with estimated permeability

13.3.2 Dispersion in an Array of Spheres

Dispersion coefficients can be estimated for the same SC array. This example also illustrates the "cloning" process described above, and is a necessary precursor to the interpretations in Section 13.5.1; it also shows the significant differences in modeling dispersion in gases, vs. liquids. The technique for measuring dispersion is the method of moments; that is, a slug of solute is injected into a flow field, and the spread of the solute 2nd moment is measured as it disperses downstream, as illustrated in Figure 13.4.

The dispersion coefficient D^* is then defined by:

$$D^* = \frac{1}{2}\,\mathrm{d}m_2/\mathrm{d}t \tag{13.9}$$

where m_2 is the second moment, or variance of the solute distribution, projected onto the x-axis (more detail about the method of moments can be found in Stockman, 1997). For the SC Geometry, $\mathrm{Pe} \equiv L \cdot U/(D_\mathrm{m} \cdot (1-\varepsilon))$, and $\mathrm{Re} \equiv L \cdot U/(\nu \cdot (1-\varepsilon))$ and where ε is the porosity and U is the average Darcy flow speed.

Figure 13.4 shows the LB-calculated dispersion patterns for the tracer for two cases. The upper figure shows dispersion for a case of $\mathrm{Sc} = 0.88$, which matches the Sc of argon in air; that gas mixture was used for an experimental investigation of gaseous dispersion in an SC array (Gunn and Pryce, 1969). The bottom of the figure shows the pattern for a hypothetical case with the same Pe, but the Re lowered so $\mathrm{Sc} = 32 = \mathrm{Pe}/\mathrm{Re} = \nu/D_\mathrm{m}$. The latter case does not represent any real tracer gas, but shows the hypothetical dispersion in near-Stokesian flow, with little inertia. At these early stages in the calculation, the lower Sc case appears to show greater dispersion. Figure 13.5 shows the flow patterns for the $\mathrm{Sc} = 0.88$ case ($\mathrm{Re} = 50$), with those for the $\mathrm{Sc} = 32$ case ($\mathrm{Re} = 4$). The flow patterns have two significant differences; the higher Re flow has (1) recirculation zones, and (2) greater maximum fluid speed in the channels between the spheres, at the same Darcy speed. Either difference may help account for the enhanced dispersion.

Figure 13.6 compares the measured D^*/D_m for LB calculations, with a set of experiments by Gunn and Pryce (1969), and with two other numerical estimates.

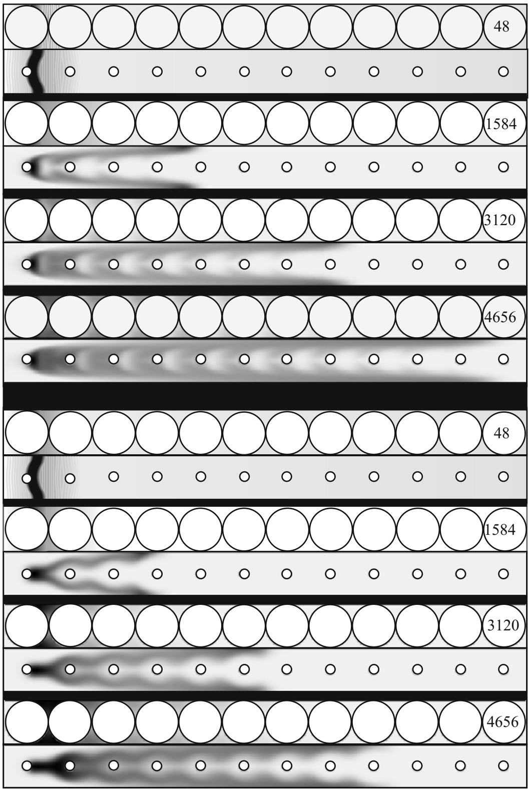


Figure 13.4. Solute patterns in the early stages of the LB calculations, for Re/Pe = 0.88 (top) and Re/Pe = 32 (bottom); both are at Pe = 132. The step number is indicated on the upper right of each frame. Each frame shows cuts at depths = 18 and 0 *lu* (out of a total depth of 36 *lu*). The solute is indicated in black

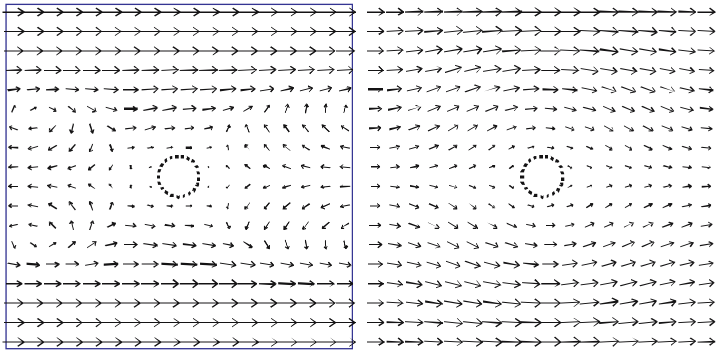


Figure 13.5. Vector flow fields around one sphere, for Re = 150 (left) and Re = 4 (right). The $x - y$ cut is at depth where the two spheres just touch. Vectors are 4th-root scaled to emphasize smaller vectors (i.e., a vector that appears half as long as another, is really 1/16 as long)

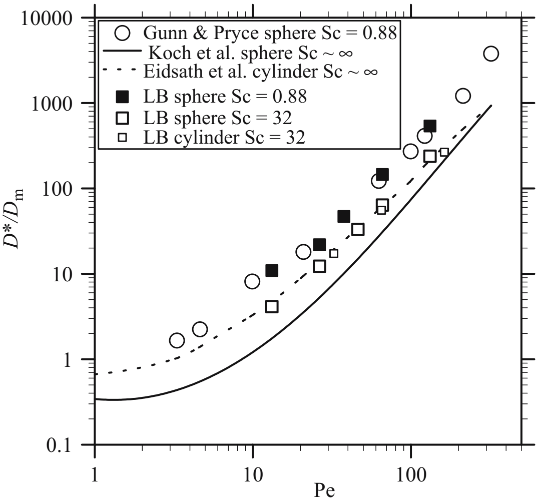


Figure 13.6. Comparison of experimental results (Gunn and Pryce, 1969), LB calculations (this study) and Stokesian theory (Koch et al., 1989). Pe defined relative to particle diameter and averaged Darcy flow speed

First, one sees that the agreement between the LB calculations and the experiments (both at Sc = 0.88) is rather good. Second, one sees that the LB calculations for the Sc = 32 produce significantly lower dispersion than for Sc = 0.88 (as might be expected from Figure 13.4). For Sc = 32, the inertial effects should be much smaller; but it is surprising to see that the deviation at even Pe ~ 15, corresponding to Re ~ 17 and 0.5 (for Sc = 0.88 and 32 respectively). Third, Figure 13.6 also contains a comparison of LB calculations of dispersion in an array of cylinders, with the 2D numerical results of Eidsath et al. (1983) for the same geometry. The latter comparison is included to show that the LB technique is capable of generating agreement with other numerical techniques, in non-trivial geometries, in case the near-agreement of the experimental results is not deemed sufficient. And fourth, the figure also plots the "Stokesian" theory of Koch et al. (1989) for a 3D SC array. The latter involved a Stokesian flow field (i.e., no inertial effects), but also included several other simplifications; e.g., effective diffusion within the spheres. In any case, it is clear that assuming a Stokesian model – widely used for dispersion in liquids – may not be appropriate for modeling dispersion in gases.

13.3.3 Unsteady Flow

Unsteady LB can be used to model high frequency wind-oscillation pumping of near surface soils (Neeper, 2001), mixing at fractures junctions, and transport in periodic dispersion experiments. Reynolds et al. (2000) used LB for flow, combined with a particle-tracking method for diffusion, to study oscillatory dispersion of

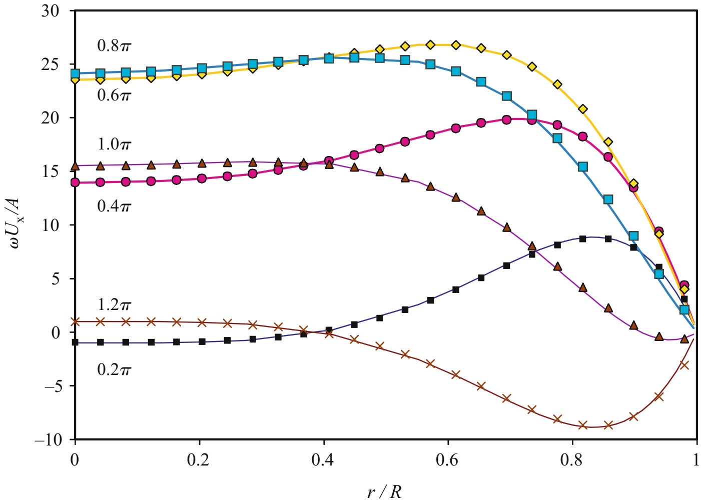


Figure 13.7. Comparison of theoretical normalized flow speed (unbroken lines) with LB calculations (symbols) at different times (where the fractions of π indicate the time offsets in fractions of one period). The abscissa is the radial position, $R = 25$ *lu*, and A is the axial pressure gradient

gases in closest-packed porous beds. However, models of deeper barometric pumping (e.g., Nilson et al., 1991) are generally out-of-reach for pore-scale LB, since the required time and length-scales are simply too large.

The following benchmark shows the ability of the model to handle oscillatory flow; in Section 13.4 we will use this example to illustrate scaling. We reproduce Lambossy's (1952) analytical solution for oscillatory flow in a cylindrical tube at a "Womersly" number $\text{Wo} = 7.237 = R \cdot (\omega/\nu)^{1/2}, \omega = 2\pi/T$, where T is the oscillation period. A similar 2D LB benchmark was recently published by Cosgrove et al. (2003), and a 2D and 3D LB comparison was published by Artoli et al. (2002).

Figure 13.7 shows the comparison of the theoretical Lambossy solution, and an LB calculation done for this study, for a cylinder with radius $= 25$ *lu*. No special effort was made to prevent the pixelation (stair-stepping) of the walls. The agreement is quite good, with the LB capturing the antegrade flow that continues after a reversal of the forcing (there is no "normalization" in this figure; the good fit is despite the fact that the $R = 25$ channel was pixelated).

13.4 FUNDAMENTAL AND PRACTICAL LIMITATIONS

There are several limitations of the LB method that give it insufficient flexibility for all arbitrary problems, or make the method far too computation-intensive. Below we explore the sources of some of these limitations, and use some illustrations to show the constraints on calculation time.

13.4.1 Bounce-Back Interpretation

During the development of lattice gas automata, it was noted (Cornubert, 1991) that the traditional bounce-back condition did not put the zero-velocity boundary on the solid nodes, but rather, at a position roughly halfway between a solid node and the adjacent fluid node. The accuracy of the reinterpreted bounce-back is a function of τ (thus the viscosity; Chen et al., 1996), but the nature of the variation is not made clear in most publications.

Figure 13.8 shows the position of the solid-fluid boundary in a small 2-D model, 8 columns wide by 8 rows high. The "nodes" of the model are shown as black circles. A solid at node (x, y) is considered a square "block" extending from $x-0.5$ to $x+0.5$, and from $y-0.5$ to $y+0.5$; solids are represented in the figure as gray blocks. This interpretation assigns a constant volume to each solid block, hence it is very easy to account for volume changes when (e.g.) solids dissolve or precipitate. In addition, when a geometry is supplied as a "voxel map" (volumetric pixels) from an X-ray tomographic scan, this interpretation allows a 1:1 map from voxel volume to LB volume.

There are several problems with the bounce-back method, as regards estimation of permeability and dispersion. Some of the LB fluid is resident on the solid nodes. Presumably, the fluid resident on the nodes must be considered in an account of applied body force, and it is not straightforward to apply forcing at the solid site. However, tests performed by the author suggest that forcing on the solid can be ignored, and forcing may be applied only at the fluid sites. Some tracers are resident on the nodes, behind the 1/2 boundary implied by the reinterpreted bounce-back; thus the effective width of a channel for dispersion may be slightly greater than the width for flow. The effective lateral diffusion coefficient of tracer at the solid nodes is debatable; with

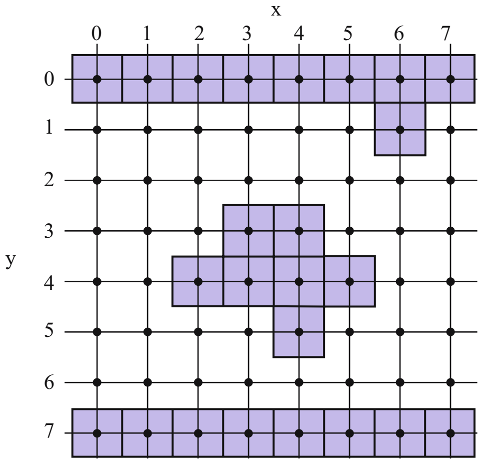


Figure 13.8. The black dots represent the node positions in the LB automaton; the gray blocks show the solid volume associated with each node in the "reinterpreted bounce-back" approach

bounce-back, this fluid does not experience the free-fluid collision step, and appears to have a lateral diffusion coefficient of zero. Again, tests suggest the overall error (for measured vs. theoretical dispersion coefficients) is small, even for narrow channels (Stockman et al., 1998). Obviously, the effect of the fluid at the interface decreases as the system is scaled to use more nodes to represent the same solid.

13.4.2 Error in Permeability for Small Channels

Figure 13.9 shows the ratio of measured to true permeability for flow between two flat plates, as a function of the width of the plate separation W (measured in nodes or lattice units, *lu*), for LB calculations with the bounce-back condition. The "measured" permeability is taken as $\mathrm{U_{meas}} \cdot \nu \cdot \rho/(\mathrm{d}P/\mathrm{d}x)_{\mathrm{meas}}$ (with $\rho \equiv 1$), while the true permeability is taken as $W^2/8$, where ν is the kinematic viscosity, U_{meas} and $(\mathrm{d}P/\mathrm{d}x)_{\mathrm{meas}}$ are the measured average flow speed and pressure gradient. The upper curve is for $\tau = 0.998$ ($\nu = 0.166\, lu^2/ts$), and the lower curve is for $\tau = 0.530$ ($\nu = 0.01\, lu^2/ts$). Clearly, there is not a monotonic increase in accuracy from high to low τ. However, with both τ choices, there is a rapid increase in accuracy, so that by $W = 10$, the error is only 1 to 2%.

13.4.3 Instability at Irregular Surfaces

With the bounce-back method, instabilities in tracer distributions may arise at irregular solid walls. Under steady-state flow at higher Pe (ranging from 25 to 500, depending on the geometry), anomalous concentrations form adjacent to protruding solids. Abnormally high concentrations form on the windward side of solids, and lows form

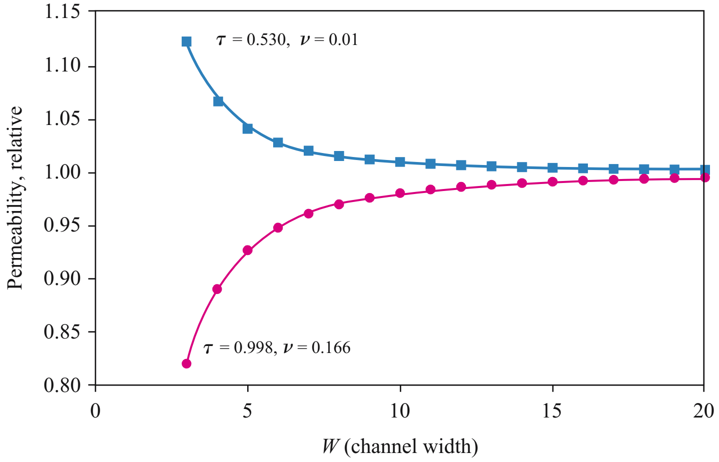


Figure 13.9. LB calculation of permeability for flow between two parallel plates, against the separation of the plates in *lu*, for two different viscosities

on the leeward side. A single node may have both a very high concentration on one side, and a very low on the other, resulting in very asymmetrical non-equilibrium distributions in the adjacent nodes. At some point, an oscillatory instability sweeps across the entire automaton, producing nonsensical concentrations.

Two conditions seem to stabilize systems against this form of instability, without causing perceptible changes in the measurement of quantities such as the dispersion coefficient. The first employs a function that sweeps through the automaton, searching for boundary solid nodes with asymmetrical concentration distributions. Any nodes which have vectors with an $(f_{\max}/f_{\min}) >$ constant, have the concentration on the node equally redistributed among the vectors. Typically, there are few such nodes; constant is currently set to 1.5.

The second stability-enhancing condition is oscillatory flow, when it is appropriate. Reversal of flow also reverses the build-up of anomalous concentrations, as the "leeward" and "windward" directions are reversed from time to time.

13.4.4 Limits Imposed by Time and Space Scaling

The LB method is limited by the Courant-Friedrichs-Lewy (CFL) criterion. Methods have been proposed for overcoming CFL criterion (Nourgaliev et al., 2003); but to date, all these methods have been of limited application, and may have severe drawbacks (e.g., calculational intensity). In the classic LB, $c \cdot ts/lu = \mathrm{CFL} \sim 1$, where c is the speed of sound in the automaton, and is typically ~1. Thus the LB method is expected to be marginally stable, and becomes unstable as the $\tau \rightarrow 0.5$ (Sterling et al., 1996).

The average fluid speed in the spaces of the LB automaton cannot exceed c. However, experience shows us that the practical upper limit is more like $U = 0.02\ lu/ts$ (where U is the average flow speed). The wide variation in pore diameter ensures that local fluid speeds may be close to 0.1 lu/ts, even when the average speed in just 0.01. This restriction puts a severe limit on the applicability of the method.

Similarly, the τ cannot be varied arbitrarily. To avoid significant error in the reinterpreted bounce-back (Chen et al., 1996), $\tau < 1$, hence (from Eq. 13.7) we must have ν and $D_{\mathrm{m}} < 0.1667$. From experience, we know that we must have ν and $D_{\mathrm{m}} > 10^{-6}$ for numerical stability.

Given these limits, there is an inexorable tie between time and length scales. One often finds that LB is quite capable of modeling a phenomenon of interest; but only for seconds. Thus LB has typically been used to model pore-scale phenomena that are expected to reach steady-state in a short time. Before deciding to apply LB to any problem, it is wise to calculate the number of steps and physical nodes required.

For example, suppose one wished to model a process that involved transfer between a minor condensed phase and flowing air, in a matrix of sand. To be concrete, we assume sand with 0.1 cm diameter grains and ~40% porosity. Such a process might be the transfer of CO_2 between a water film and air, the evaporation of an organic phase in remediation, or the sorption of a tracer gas on the solid grains. We take a pore flow speed $U \sim 2$ cm/s as an example, implying $\mathrm{Re} \sim \mathrm{Pe} \sim 10$ (the length scale is taken as the grain diameter). Such flow would be fast enough to justify using a Navier-Stokes solver. We pick 10 *lu* for the average grain size; from Figure 13.2, it can be seen that this choice of grain size may still leave substantial (10–20%) permeability errors. If near-spherical, the average grain will then have a volume of $\sim$524 lu^3. If we aim for $\sim$40% porosity, and we choose a modest LB model with $100 \times 50 \times 50$ nodes (the longest axis in the direction of flow), the simulation will contain $\sim$ 500 grains. The long distance (100 *lu*) will correspond to just 1 cm. At the maximum average LB speed of 0.02 lu/ts, it would take 5000 *ts* for air to travel the length of the automaton, which would correspond to 0.5 s of real time; that is, 10000 *ts* correspond to 1 s. To model 1 minute of the process, $6 \cdot 10^5$ *ts* would be required. Using the fast dispersion-only modes, a personal computer with fast memory (e.g., 6 GByte/s transfer while streaming) would update at $\sim 4 \cdot 10^7$ sites/s, so it would take $100 \times 50 \times 50\,\text{nodes} \times 6 \cdot 10^5\ ts/(4 \cdot 10^7\ \text{node-}ts/s) = 3750$ seconds of computer clock time, or about 1 hour – to model just one minute of the real process – and then, over a flow path of just 1 cm. If we opted instead for an average grain diameter of 20 *lu*, but still wanted $\sim$ 500 grains in the model, the clock time would increase by a factor of 16.

Now suppose one wishes to measure the dispersion coefficient in a similar porous medium, as a function of $\mathrm{Pe} \sim \mathrm{Re}$, with $1 \leq \mathrm{Pe} \leq 100$, by injecting a slug of solute and measuring the second moment of the tracer (Stockman, 1997) as it disperses downstream along the x-axis. We begin by assuming that the $100 \times 50 \times 50$ segment is sufficient to define a representative volume; however, we expect that this volume must be cloned in the x-direction, as described in Section 13.2.2. First, continuity of flow must be provided at the boundaries of the cloned units; the easiest approach is to mirror the representative volume along the x-axis, so our 100 x-direction *lu* might really contain a $50 \times 50 \times 50$ volume and its mirror image. Then how many of these cloned units would we need, to measure D^* with reasonable confidence?

Typically, at least one characteristic diffusion time $\tau_D = L^2/D_\mathrm{m}$, is needed to reach a steady-state, linear growth of the second moment, where L is a characteristic length scale. We take $L = 10\,lu =$ the grain diameter. To achieve $\mathrm{Pe} = U \cdot L/D_\mathrm{m} = 100$, with a maximum U of 0.02 lu/ts, D_m must be $2 \cdot 10^{-3}$, and thus $\tau_D = 5 \cdot 10^4$ *ts*. To use the second moment method, the slug must stay entirely within the solid domain – it can't fall off the edge. Thus the automaton must be at least as long as the distance the slug would travel in τ_D, or $\sim U \cdot \tau_D = 0.02 \cdot 5 \cdot 10^4 = 10^3$ *lu*. Thus the 100-long geometry must be cloned 10 times in the x-direction. The number of cells in the total cloned automaton would be $50 \times 50 \times 10^3 = 2.5 \cdot 10^6$. In the code described by Stockman (1999), $\sim$25 bytes per node are required in diffusion-only mode. Thus the simulation would require 62.5 Mbytes. Again assuming the update rate (that is, the real computer calculation rate) of $\sim 4 \cdot 10^7$ sites/s, the $\mathrm{Pe} = 100$ run would take $(2.5 \cdot 10^6 \text{sites}) \times (5 \cdot 10^4\ ts)/(4 \cdot 10^7\ \text{site} \cdot ts/s) = 3125\ s$, or less than an hour. However, note that the calculation time would go as Pe^2, so a substantially higher Pe might be impractical. At $\mathrm{Pe} \sim 500$, the calculation would take nearly a day.

(**3**) Are the *ts* allotted, and the length of the cloned channel in question 1 of the example, sufficient to reach a dispersive steady-state?

Finally, consider what the oscillatory flow simulation in Figure 13.7 represents, in terms of both real gaseous flow through a pore or fracture, and the automaton needed to carry out the calculation. The results in Figure 13.7 were carried out with an $8 \times 53 \times 53$ lu^3 automaton, the radius of the cylinder (parallel to the x-axis) was $R = 25$ *lu* (nodes on the x-axis could have been 1, but the vectorization built into the program kept the simulation to $x = 8$). The simulation was carried out to match the $Wo = 7.237$ used in Lambossy's original benchmark (Lambossy, 1952). This *Wo* was achieved by setting the period of oscillation to 750 *ts*, with a $\nu = 0.100$; the corresponding $\text{Re} = 4.47$, defined relative to the diameter and the maximum flow speed. This simulation was taken through 400 oscillation periods ($3 \cdot 10^5$ *ts* total), to ensure a steady state was reached ($3 \cdot 10^5$ *ts*), and took $\sim$2 hours of computer time.

Suppose then that the simulation in Figure 13.7 were to represent gaseous oscillations, first at the diameter of a large pore (1 mm), then at the diameter of a macropore in soil (1 cm). The period of oscillation, $T = \pi(d/Wo)^2/(2\nu)$, where d is the diameter, and $\nu = 0.14$ cm^2/s, would be 0.00214 s for the 1 mm pore, and 0.214 sec for the 1 cm pore. Obviously, these periods are probably unrealistic for surface wind driven phenomena; but the power two dependence on *Wo* means that by merely dropping *Wo* to 2, the period corresponds to $\sim$3 seconds for the 1 cm macropore. However, for modeling dispersion, a greater period of oscillation means that a physically longer simulation is required (both in *ts* and in axial *lu*), to keep a slug of solute from running off the end of the automaton. Thus pore-scale LB is not practical for modeling phenomena that have longer periods, such as deep barometric pumping (Nilson et al., 1991).

13.5 REPRESENTATIVE VOLUME

When one uses pore-scale LB to estimate permeability or dispersion coefficients in porous media, it is necessary to determine a representative volume with confidence. For reasons of computational efficiency, the volume should contain as few nodes as practical; yet the more nodes, and the better the resolution of pores (i.e., the higher the number of nodes/cm), the more likely the calculation is to be accurate.

13.5.1 Porous Filter Cake

To illustrate the choice of representative volume with a practical example, consider the determination of permeability in a porous filter cake, as might be formed during minerals processing. The geometry for this example is a digital $256 \times 256 \times 100$ voxel, X-ray microtomagraphic image (provided by C.L. Lin; see Lin and Miller, 2004 for methods).

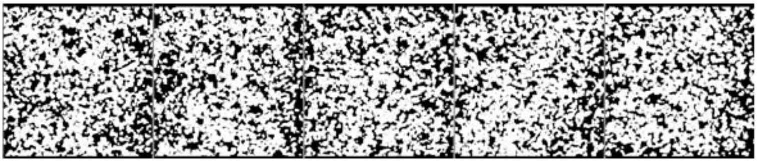


Figure 13.10. Five cuts through the porous filter cake, at different depths into the sample.

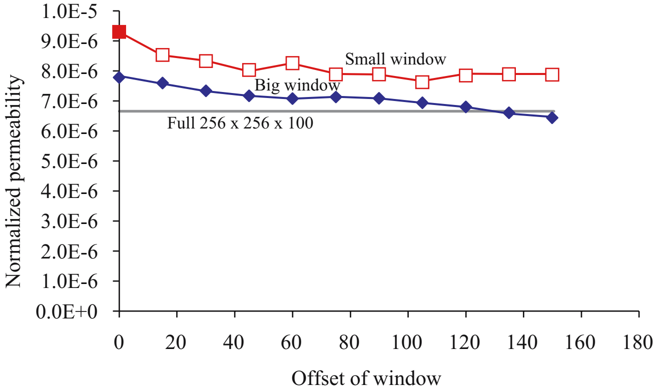


Figure 13.11. Measured LB permeability, for 50-node-wide and 100-node-wide windows moved left-to-right across the 256 × 256 × 100 geometry

Figure 13.10 shows five cuts through a 256 × 256 × 100 geometry. The particles have apparently not compacted evenly, and there are variations in the density of each. To test the effects of these density variations, we take successive subsamples of the full geometry, marching from left to right across the geometry, and determine the LB permeability on each slice, using windows of two sizes.

Figure 13.11 shows the variation of the LB-measured permeability (Re ∼ 0), for subsamples within a 50-wide pixel window ("small window") and a 100-window, as the windows are moved from left to right across the 256-pixel-wide sample. The horizontal line marked "full" represents the 256 × 256 × 100 pixel LB calculation. Clearly, both windows yield a slight underestimation in the aggregate permeability (but never by more than a factor of 1.39), and the error varies with position across the sample; but the net variation for the larger window is much smaller. Actually, the distribution of higher speed "jets" among the pores in quite inhomogeneous; in a few channels, the fluid moves up to 20 times faster than the net Darcy flow speed, so the small amount of variation is actually rather surprising.

To estimate the dispersion coefficient in this same sample, a 1500-node long porous medium was constructed by allowing the flow to reach steady state, then the solids were cloned in the x-direction, as described in Section 13.2.2. A solute slug was then released at one end of the automaton, and evolution the 2nd moment of the slug

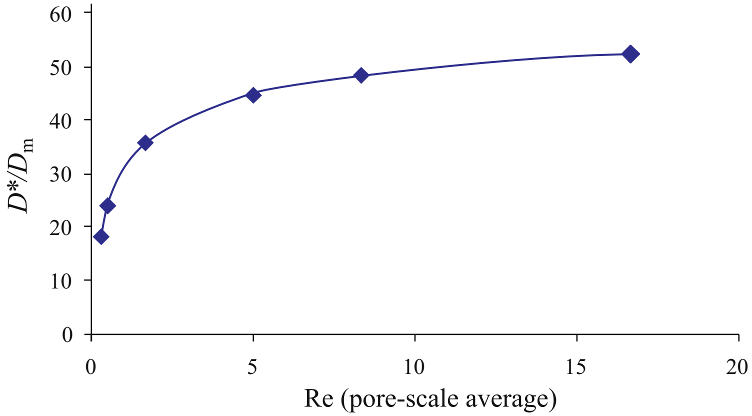


Figure 13.12. The LB measured dispersion coefficient D^*, as a friction of the speed of increasing inertial effects (Re), for a fixed Pe = 5

was used to calculate the dispersion coefficient (per Section 13.3.2). Figure 13.12 shows the variation of measured dispersion coefficient, as a function of Re (defined relative to the average diameter of the grains in the aggregate). One would not expect to see this type of plot with a real gaseous system, as the ratio of $D_{\mathrm{m(tracer)}}/\nu_{\mathrm{(bulkfluid)}}$ would be fixed. Here, however, this numerical exercise allows us to see the surprising variation of D^*, particularly at modest Re. The source of this variation is probably the inhomogeneous distribution of the flow, with a few channels through the automaton having speeds much greater than the Darcy speed; in effect, the local Re (in a fast-moving channel) may exceed 20, even for average Re $\sim$ 1.

13.5.2 Fracture Flow

The selection of representative volume also affects modeling of flow in fractures, even synthetic fractures that are intentionally kept simple. Consider the dispersion of a water-borne solute between a flat and an irregular plate; while this is clearly not a "vapor" application, it does illustrate some limits of the method at high Pe. The experimental setup is described in detail by Detwiler et al. (2000). In brief: two 14.8 by 30.2 cm sheets of textured glass plate ("shower glass") were placed together, with an average separation (aperture) of 0.0221 cm. The full plate images were captured on a digital camera in a 958 × 1958 pixel image. The pattern on the glass was small-scale, so the average correlation length (λ) of the aperture (defined in Detwiler et al.) was only $\sim$0.044 cm. A steady-state flow of water was induced between the transparent plates, then a slug of solute was introduced and the dispersion of solute was determined by the method of moments. Initially, the slug of solute was $\sim$1.5 to 2 cm wide, and experienced little transverse dispersion on its path through between the plates; hence the actual geometry sampled by the slug was actually $\sim$2 cm × 28 cm. In the original paper, the experiment was modeled by approximating the flow field with the Reynolds equation, with particle tracking to determine the dispersion in that flow field. The aperture field between the two plates was determined by filling the gap with a constant concentration of dye.

For the LB simulation of the Detwiler et al. experiment, a small subset of the geometry was chosen, mirrored, and replicated in the x-direction after a steady-state flow field was achieved. Then a slug source was introduced at one end of the automaton, allowed to disperse downstream, and the method of moments was used to determine the D^*. Two 1.54 cm by 1.54 cm subsamples of the real system, corresponding to 100 × 100 pixels, were used for the LB simulation. This LB size was picked to contain 35λ, since Detwiler et al. reasoned that at least 20λ was needed to overcome ergodic effects. This subsample also approximated the width of the initial solute pulse in the real experiments. Since only the aperture field was known (not the distribution of porosity along the axis perpendicular to the plates), LB runs were performed with the aperture symmetrically disposed between the plates, and all on one side (Figure 13.13 inset).

There were several difficult constraints on the LB model, which deserve mention as illustrations of limits. First, the experiment was designed to allow reliable determination of the solute distribution, and to be suitable for Reynolds equation modeling; for those reasons, the slopes on the textured surface were gentle. Unfortunately, the LB method for this study used uniform gridding in the x-, y-, and z-directions, so nodes

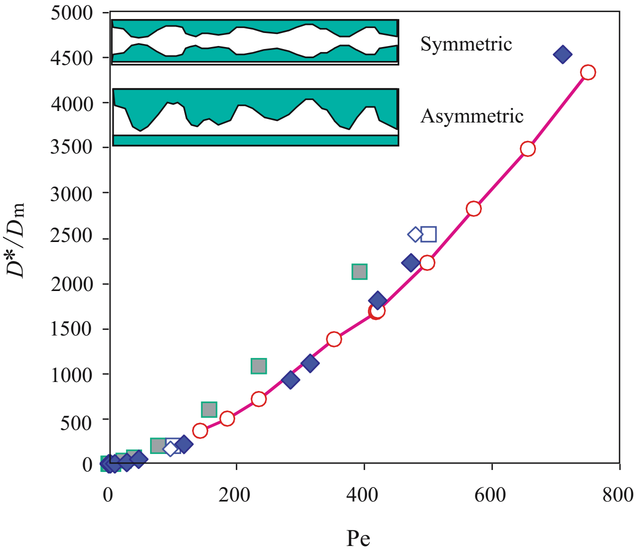


Figure 13.13. Comparison of Detwiler et al. (2000) experimental results (circles connected by line segments) and LB calculations. The diamonds are symmetric LB calculations (see inset, upper left), and the squares are for asymmetric calculations. The filled diamonds and squares are for LB subsample 1, and the open squares and diamonds (two each) are for subsample 2

were not distributed efficiently, and the 100 × 100 pixel image had to be stretched into a 400 × 400 × 20 LB automaton to achieve the correct aspect ratio. The geometry was then reflected in the flow direction, to assure continuity of flow fields. Finally, after the automaton reached a steady-state flow, the geometry was cloned up to 16 times in the flow direction (to provide a long path for method-of-moments dispersion, as described in Section 13.2), and a slug of solute introduced. Second, the irregular surfaces induced the instability described in Section 13.4.3, which limited the maximum Pe to ∼700. (This study was completed before the technique described in Section 13.4.3 was available.)

Figure 13.13 compares the LB estimated dispersion coefficient, with the experimental dye dispersion measurements. Despite all the limitations, the LB technique gave reasonable agreement. The two subsamples give significantly different results, but the differences are comparable to the differences created by other assumptions – e.g., the assumption of asymmetric vs. symmetric disposition of the apertures, which is an inherent uncertainty in the Reynolds equation modeling used by Detwiler et al. (2000).

13.6 CONCLUSIONS

Lattice Boltzmann modeling provides a powerful technique for investigating gas permeability and dispersion in complex geometries. Results averaged over many pore scales can be used to estimate macroscopic transport properties. However, the method is not a panacea, and for many geologic problems, pore-scale modeling is far too calculation-intensive. Significant problems include the need to define a representative volume; the fixed CWL number, often implying that only hours of physical time can be modeled with days of computer computation; and the restriction to "modest" Pe and Re (<1000 in a typical problem). Nonetheless, LB can model many coupled phenomena that are difficult to approach with traditional methods, such as finite difference and finite element techniques.

Answers to Self-Study Examples:

(1a) Yes. From Section 13.2.2, viscous equilibrium requires $\sim L^2/\nu = (26-2)^2/0.01 = 57600$. (The right and left, and top and bottom sites of the channel are solid, so the channel is 24 lu wide.)

(1b) First part: (sites $= 26 \times 26 \times 24) \times (60000\ ts)/(9 \cdot 10^6$ ts·sites/s$) = 108$ s.

Second part: (sites $= 26 \times 26 \times 24 \times 48) \times (60000\ ts)/(35 \cdot 10^6$ ts·sites/s$) =$ 1335 s.

(1c) First part: memory to store vectors: (sites $= 26 \times 26 \times 24) \cdot$ (19 vectors) × (4 bytes/vector) = 1233024 bytes = 1.18 MB. There is a smaller amount of memory required to store pointers, $26 \cdot 26 \cdot 24$ bytes required to store solid site information, and (sites $= 26 \times 26 \times 24) \times$ (3 speed components) × (4 bytes/component) = 0.185 MB required to store the x, y, z components of the flow velocity at each site.

Second part: memory to store vectors: (sites $= 26 \times 26 \times 24 \times 48$) $\times$ (6 vectors) $\times$ (4 bytes/vector) = 1233024 bytes= 17.8 MB. The memory to store the flow speed is the same as in (1b).

(2) From Eq. (13.8), Pe = Re = 24 = ($L = 24\,lu$) $\times$ (U = 0.01 *lu/ts*)/(ν or $D_{\mathrm{m}} = 0.01\ lu^2/ts$); Sc = 1.

(3) From Section 13.4.4, a dispersive steady-state is reached after a slug travels for a characteristic diffusion time $\sim L^2/D_{\mathrm{m}} = 24^2\ lu^2/0.01\ (lu^2/ts) = 57600\ ts$. The channel must be long enough to contain the entire slug during this time, else part of the slug will "fall off" the right end of the channel, and will wrap back if there is no concentration boundary at the channel end. While the average speed across the channel is 0.01 *lu/ts*, the solute in the middle of the channel will travel faster. In a round pipe, the middle fluid travels twice the average speed; this speed would be reached by the solute only at very high Pe, but to be on the safe side, we assume this factor 2. Thus the distance traveled by the leading edge of the solute, in the characteristic diffusion time, is (57600 *ts*)$\times$ (2 $\times$ 0.01 *lu/ts*)=1152 *lu*. The cloned channel is 48 $\times$ 24 = 1152 *lu* long, so the channel length is adequate.

REFERENCES

Artoli, A.M., Hoekstra, A.G. and Sloot, P.M.A. (2002) 3D pulsatile flow with the lattice Boltzmann BGK method, *Internat. J. Modern Phys. C* 13(8), 1119–34.

Chen, S.J., Martínez, D. and Mei, R. (1996) On boundary conditions in lattice Boltzmann methods, *Phys. Fluids* 8, 2527–2535.

Cornubert, R., d'Humières, D. and Levermore, D. (1991) A Knudsen layer theory for lattice gases, *Physica D* 47, 241–259.

Cosgrove, J.A., Buick, J.M., Tonge, S.J., Munro, C.G., Greated, C.E. and Campbell, D.M. (2003) Application of the lattice Boltzmann method to transition in oscillatory channel flow, *J. Phys. A: Math. Gen.* 36, 2609–2620.

Detwiler, R.L., Rajaram, H. and Glass, R.J. (2000) Solute transport in variable-aperture fractures: An investigation of the relative importance of Taylor dispersion and macrodispersion , *Water Resources Res.* 36, 1611–1625.

Eidsath, A., Carbonell, R.G., Whitaker, S. and Herrmann, L.R. (1983) Dispersion in Pulsed Systems – III Comparison between Theory and Experiments for Packed Beds, *Chem. Engng. Sci.* 38, 1803–1816.

Flekkøy, E.G., Oxaal, U., Feder, T. and Jøssang, T (1995) Hydrodynamic dispersion at stagnation points: simulations and experiments, *Phys. Rev.* E 52, 4952–4962.

Gunn, D.J. and Pryce, C. (1969) Dispersion in Packed Beds, *Trans. Inst. Chem. Engrs.* 47, T341-T350.

Koch, D.L., Cox, R.G., Brenner, H. and Brady, J.F. (1989) The effect of order on dispersion in porous media, *J. Fluid Mech.* 200, 173–188.

Lambossy, P (1952) Oscillations forcées d'un liquide incompressible et visqueaux dans un tube rigide et horizontal. Calcul de la force frottement, *Helv. Phys. Acta* XXV, 371–386.

Lin, C.L. and Miller, J.D. (2004) Pore structure analysis of particle beds for fluid transport simulation during filtration, *Inter. J. Mineral Process* 73(2–4), 281–294.

Martys, N. and Chen, H. (1996) Simulation of multicomponent fluids in complex three-dimensional geometries by the lattice Boltzmann method, *Phys. Rev. E* 53, 743–750.

Neeper, D.H. (2001) A model of oscillatory transport in granular soils, with application to barometric pumping and earth tides, *J. Contam. Hydrology* 48, 237–252.

Nilson, R.H., Peterson, E.W., Lie, K.H., Burkhard, N.R. and Hearst, J.R. (1991) Atmospheric pumping: a mechanism causing vertical transport of contaminated gases through fractured permeable media, *J. Geophys. Res.* 96(B13), 21933–21948.

Noble, D.R. (1996) Lattice Boltzmann Study of the Interstitial Hydrodynamics and Dispersion in Steady Inertial Flows in Large Randomly Packed Beds. Ph.D. thesis, University of Illinois Urbana-Champaign.

Nourgaliev, R.R., Dinh, T.N., Theofanous, T.G. and Joseph, D. (2003) The lattice Boltzmann equation method: theoretical interpretation, numerics and implications, *International Journal of Multiphase Flow* 29, 117–169. (CFL, methods for dissociating grids and getting CFL < 1).

Pan, C., Hilpert, M. and Miller, C.T. (2004) Lattice-Boltzmann simulation of two-phase flow in porous media, *Water Resources Res.* 40(1), W01501 14pp.

Qian, Y.H., d'Humières, D. and Lallemand, P. (1992) Lattice BGK models for Navier-Stokes equation, *Europhys. Lett*. 7(6 BIS), 479–484.

Reynolds, A.M., Reavell, S.V. and Harral, B.B. (2000) Flow and dispersion through a close-packed fixed bed of spheres, *Phys. Rev. E* 62(3), 3632–3639.

Rothman, D.H. and Zaleski, S. (1997) *Lattice-Gas Cellular Automata: Simple Models of Complex Hydrodynamics.* Cambridge University Press, Cambridge, 297p.

Sterling, J.D. and Chen, S.Y. (1996) Stability analysis of Lattice Boltzmann methods, *J. Computat. Phys.* 165(1), 288–306.

Stockman, H.W. (1997) A lattice-gas study of retardation and dispersion in fractures: Assessment of errors from desorption kinetics and buoyancy, *Water Resources Res.* 33 (8), 1823–1832.

Stockman, H.W., Li, C. and Wilson, J.L. (1997) A lattice-gas and lattice Boltzmann study of mixing at continuous fracture junctions: importance of boundary conditions, *Geophys. Res. Lett.* 24(12*)*, 1515–1518.

Stockman, H.W., Glass, R.J., Cooper, C. and Rajaram, H. (1998) Accuracy and computational efficiency in 3D dispersion via lattice-Boltzmann: models for dispersion in rough fractures and double-diffusive fingering, *International Journal of Modern Physics* C 9(8), 1545–1557.

Stockman, H.W. (1999) *A 3D Lattice Boltzmann Code for Modeling Flow and Multi-Component Dispersion*. SAND 99–0162, Sandia National Laboratories, Albuquerque, NM. 165 pp.

Sukop, M.C. and Or, D. (2004) Lattice Boltzmann method for modeling liquid-vapor interface configurations in porous media, *Water Resources Res.* 40(1), W01509 11pp.

Wolf-Gladrow, D. (1994) A lattice Boltzmann equation for diffusion, *J. Stat. Phys.* 79, 1023–1032.

Zhang, X., Bengough, A.G. and Deeks, L.K. (2002) A novel three-dimensional lattice Boltzmann model for solute transport in variably saturated porous media, *Water Resources Res.* 38(9), 6–1 to 6–10.

Zick, A.A. and Homsy, G.M. (1982) Stokes Flow through Periodic Arrays of Spheres, *J. Fluid Mech.* 115, 13–26.