Anal. Chem. **2007,** *79,* 2965−2971

Lattice-Boltzmann Simulations of Three-Dimensional Fluid Flow on a Desktop Computer

Jeffrey D. Brewster*

Agricultural Research Service, Eastern Regional Research Center, United States Department of Agriculture, 600 East Mermaid Lane, Wyndmoor, Pennsylvania 19038

The lattice-Boltzmann (LB) method is a cellular automaton approach to simulating fluid flow with many advantages over conventional methods based on the Navier−Stokes equations. It is conceptually simple, amenable to a wide array of boundary conditions, and can be adapted to handle thermal, density, miscibility, and other effects. The LB approach has been used to model a number of fluid systems of interest to analytical chemists, including chromatography columns, micromixers, and electroosmotic pumps. However, widespread use of this tool has been limited, in part because virtually all large-scale 3D simulations in the literature have been executed on supercomputers. This work demonstrates that such simulations can be executed in reasonable periods of time (hours) on a desktop computer using a cross-platform software package that is easy to learn and use. This package incorporates several improvements that enhance the utility of the LB approach, including an algorithm for speeding common calculations by 2 orders of magnitude and a scheme for handling convection−diffusion equations of interest in electrochemical and surface reaction studies.

Computer simulations are extremely useful tools for investigating and optimizing analytical systems and devices. Simulations can be used to interpret experimental data and to replace tedious and lengthy experiments or device construction and testing, leading to more rapid progress. Electrochemists have made extensive use of simulations for reaction−diffusion systems,[1] but simulations of convective (flow) systems in electrochemistry[2] and other analytical disciplines has been much more limited. This is in part due to the greater complexity of computational fluid dynamics algorithms, the cost of commercial simulation packages, and the need (historically) for supercomputers to perform simulations of large-scale, three-dimensional systems. The time and expertise required to perform such simulations can consume much of the resources available in a research program. My interest in simulating processes for the capture and detection of pathogenic bacteria in food led to an investigation of methods for simulating fluid flow coupled to surface reactions. Initial efforts using finite difference methods suffered from poor stability and difficulty in establishing meshes and boundary conditions even for relatively simple geometries. The lattice-Boltzmann (LB) method proved far more effective. LB methods have been applied with increasing frequency for analytical simulations such as flow in chromatographic columns[3,4] and in microfluidic devices.[5] A simple two-dimensional LB simulation was coded in a few hours, and the method appeared robust and readily adaptable to realistic geometries. More complex three-dimensional geometries, faster computation, and appropriate treatment of boundary conditions required considerably more work, but steady progress resulted. However, it was not expected that large-scale simulations would be feasible, since virtually all large-scale, three-dimensional studies in the literature employed supercomputers,[4] supercomputer clusters,[5] or grid computing engines.[6] Not only is this a limitation in terms of access to computing resources, but programs must often be rewritten to operate efficiently on high-performance systems, raising an additional barrier to the nonexpert. This report describes the development and application of software capable of simulating large-scale (128 × 128 × 128 and larger) three-dimensional systems in reasonable times (hours) on a desktop computer. The code is written as an extension to the Mathematica technical computing package and takes advantage of the built-in programming language and extensive graphics capability of Mathematica for mesh generation and display of data. It is intended to be easily learned and used by the nonspecialist. A number of boundary conditions are handled, including stationary walls, moving walls, periodic boundaries, and pressure boundaries. Extensions of the basic algorithm are employed that can dramatically speed up common calculations and allow simulation of convection−diffusion systems.

BACKGROUND

Lattice-Boltzmann Equation. The starting point for conventional simulations of fluid flow are the Navier−Stokes equations, which describe the fluid from a macroscopic viewpoint as a

* Corresponding author. Tel.: 215-233-6447. Fax: 215-233-6559. E-mail: jeffrey.brewster@ars.usda.gov.

(1) Feldberg, S. In *Electroanalytical Chemistry*; Bard, A. J., Ed.; Marcel Dekker: New York. 1969; Vol. 3, pp 200−295.
(2) Moldoveanu, S.; Anderson, J. L. *J. Electroanal. Chem.* **1984**, *175*, 67−77.
(3) Schure, M. R.; Maier, R. S.; Kroll, D. M., Davis, H. T. *Anal. Chem.* **2002**, *74*, 6006−6016.
(4) Maier, R. S.; Kroll, D. M.; Kutsovsky, Y. E.; Davis, H. T.; Bernard, R. S. *Phys. Fluids* **1998**, *10*, 60−74.
(5) Li, C.; Chen, T. *Sens. Actuators, B* **2005**, *106*, 871−877.
(6) Harting, J.; Chin, J.; Venturoli, M.; Coveney, P. V. *Phil. Trans. R. Soc. A* **2005**, *363*, 1895−1915.

10.1021/ac062178v Not subject to U.S. Copyright. Publ. 2007 Am. Chem. Soc.
Published on Web 02/24/2007

continuous material. Applying this set of coupled, nonlinear equations to a discrete space and time lattice gives a set of simultaneous nonlinear finite difference or finite element equations, which may be transformed to a set of simultaneous, linear algebraic equations with appropriate approximations. Algorithms for building lattices and solving the equations are not trivial, and memory demands are relatively high. Solution of the equations is nonlocal; i.e., it requires information from all lattice sites (as well as initial and boundary conditions) and is not amenable to parallel computation. These limitations stimulated the development of an alternative approach to flow simulation based on a microscopic viewpoint, which can be efficiently computed in parallel.

The Boltzmann equation describes a fluid from a microscopic viewpoint as an ensemble of discrete particles having a distribution $f = f(\mathbf{v}, \mathbf{x}, t)$, where f is the probability of finding a particle with velocity (or momentum) in the range $(\mathbf{v}, \mathbf{v} + d\mathbf{v})$ and position in the range $(\mathbf{x}, \mathbf{x} + d\mathbf{x})$ at time t:

$$\partial_t f + \mathbf{v}\cdot\nabla f = Q(f)$$

The collision operator $Q(f)$ describes the effects of interparticle collisions, which result in relaxation of the probability distribution toward the equilibrium state given by the Maxwell distribution. The LB equation results from applying the Boltzmann equation to a discrete space and time lattice with intervals Δx and Δt. Particles are constrained to move along a finite set of velocity vectors $\mathbf{v}_i$ connecting each lattice site to its neighbors. The magnitude of each $\mathbf{v}_i$ is such that a particle moves to the next lattice site along the vector in exactly Δt time units. While the particles treated by the Boltzmann equation are generally understood to be of molecular dimensions, the LB method assumes that the dynamics of pseudoparticles of macroscopic dimensions are governed by the same equation. The behavior of the fluid is simply given by the collective behavior of pseudoparticles as they move from one lattice site to another, colliding and exchanging momentum with each other and the solid boundaries of the system. Using a discrete lattice and expressing the time derivative with an explicit finite difference approximation gives the lattice-Boltzmann equation:

$$f_i(x + \mathbf{v}_i\Delta t, t + \Delta t) - f_i(x,t) = \frac{1}{\tau}\left[f_i^{eq} - f_i(x,t)\right]$$

where f_i is the density carried by vector $\mathbf{v}_i$. $Q(f)$ has been approximated as a single relaxation process with time constant τ, with

$$f_i^{eq} = w_i\rho\left[1 + \frac{1}{c_s^2}(\mathbf{v}_i\cdot\mathbf{u}) + \frac{1}{2c_s^4}(\mathbf{v}_i\cdot\mathbf{u})^2 - \frac{1}{2c_s^2}\mathbf{u}^2\right],$$

$$\rho = \sum f_i, \quad \mathbf{u} = \frac{1}{\rho}\sum \mathbf{v}_i$$

Parameter w_i is a weighing factor calculated to provide conservation of mass and momentum, c_s the lattice speed of sound, ρ the density, and $\mathbf{u}$ the velocity. Members of the family of LB lattices and equations defined by different sets of $\mathbf{v}_i$ are usually designated DdQq, where d is the number of dimensions and q the number of velocity vectors (e.g. D2Q9). Only certain sets of $\mathbf{v}_i$ obey the conservation relations and provide rotational invariance. For computation, the LB equation is usually broken into two steps, a collision step that redistributes density within a cell,

$$f_i^*(x,t) = f_i(x,t) + \frac{1}{\tau}\left[f_i^{eq} - f_i(x,t)\right]$$

and a streaming step, which transports density from neighboring cells

$$f_i(x, t + \Delta t) = f_i^*(x - \mathbf{v}_i\Delta t, t)$$

These computations are implemented by simple operations or rules similar to those used to define cellular automata. Modified rules are used along boundaries to impose no-slip, fixed pressure or velocity, impermeability, or periodic conditions.[7]

Advantages and limitations of the Lattice-Boltzmann Method. LB calculations involve only neighboring lattice sites, and memory requirements are modest, making the method well suited to parallel computation on multiprocessor supercomputers. The simplicity and modularity of the algorithm make it easy to program and debug, and execution is fast. The basic LB equation is efficient, stable, and accurate for low Reynolds number flow and Mach number below 0.3, a regime that includes most applications of interest. There are limitations of the method, however. Lattice spacing, time spacing, fluid viscosity, and compressibility are not independent parameters, and care must be taken to constrain their values to obtain stable, accurate results. The fluid represented by typical values of Δx and Δt is much more compressible than water, so physically accurate pressure values cannot be obtained directly from a simulation. Some of these limitations can be alleviated by modification of the basic equation to enhance stability, improve performance at high Reynolds numbers, account for diffusion, etc.,[8,9] and references therein and research in the area is active. The philosophy, development, and applications of the LB method are well described in the literature, which includes a monograph,[8] numerous reviews,[10] and many publications, of which only a representative few are cited here. The reader is referred to these sources for further information on the method and its implementation.

METHODS

Code Development and User Interface. The technical computing system Mathematica (Wolfram Inc., Champaign, IL) was used for code development, simulation setup and analysis and presentation of data. Mathematica is widely available and relatively inexpensive, provides extensive graphics and computational capabilities, and is available in versions for several operating systems (including Unix, Windows, and OS X). Programs were initially written in Mathematica's built-in programming language and tested with small-scale simulations. This interpreted language

(7) Maier, R. S.; Bernard, R. S.; Grunau, D. W. *Phys. Fluids* **1996**, *8*, 1788-1801.
(8) Succi, S. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*; Clarendon Press: Oxford, 2001.
(9) Brenner, G. LRN-Workshop. 2004; http://www.hlrn.de/events/Ing2004/abstracts/a-brenner.pdf.
(10) Raabe, D. *Modell. Simul. Mater. Sci. Eng.* **2004**, *12*, R13−R46.

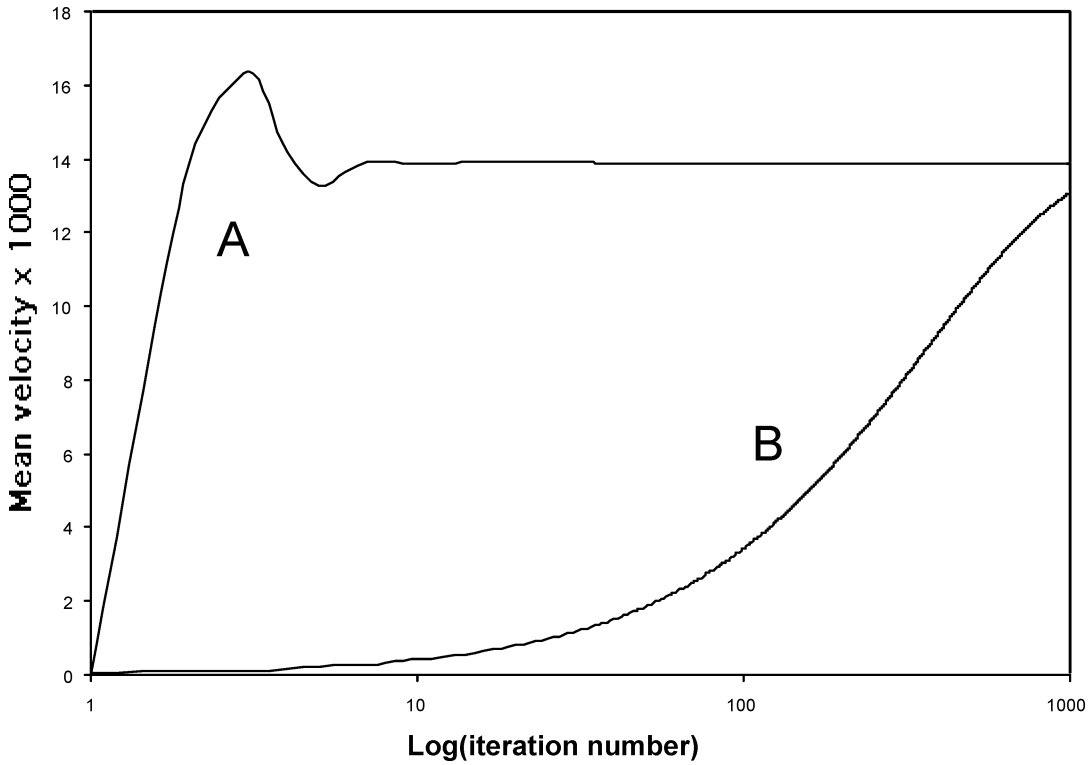

Figure 1. Relaxation of simulated flow velocity through a rectangular duct containing a sphere. Flow induced by body force with periodic boundary conditions at the duct walls. (A) With PID loop adjusting body force. (B) With constant body force. The duct was 25 × 25 × 25 cells, and the sphere radius was 6 cells.

allowed rapid development and debugging of code since no tedious compiling and linking was required, although execution speed was slow. The core calculation routines were then rewritten in the C language, a process facilitated by the C-like syntax of the Mathematica programming language. The C language program was compiled with an interface module (Mathlink) that allowed the C simulation "engine" to be called from, and exchange data with, Mathematica. Mathematica routines were used to establish boundary conditions and initialize the data arrays, which were then passed to the C program for simulation. The resulting data were returned to Mathematica for display and analysis. Other technical computing systems such as Maple or Matlab could be used in a similar fashion, and data could be passed to and from the C engine via files or pipes to facilitate other avenues for data visualization. All simulations were performed on an Apple PowerMac G5 (Apple, Inc., Cupertino, CA) containing a single PowerPC 970 (G5) processor running at 1.8 GHz, with 900 MHz bus speed, 512 KB L2 cache memory, and 2.5 GB system memory.

Lattice-Boltzmann Simulation Algorithm. The algorithm for a domain with dimensions n_x, n_y, n_z is based on three arrays. Each array includes an outer layer of boundary cells for a total of $(n_x + 2)(n_y + 2)(n_z + 2)$ sites or cells. The *cells* array contains the 19 floating point values per cell representing the densities f_i on the D3Q19 lattice.[7] The *types* array contains a single integer value per cell that encodes the type of cell (fluid, wall boundary, pressure boundary, etc.). The *concns* array contains one floating point value per diffusing species and is used only for simulations involving mass transport or reaction. During collision computations, the *types* array is interrogated to identify fluid cells for collision. During streaming, *types* is tested to determine the appropriate rule to apply to each neighboring cell. The current algorithm duplicates the entire cells array to avoid overwriting existing cells during streaming. Double precision (64 bit) floating point values were used throughout. Memory usage could be reduced significantly by employing the method of Argentini et al.[11] to duplicate only the neighboring layer of cells during each computational step.

The streaming algorithm employed no-slip boundary conditions at the walls and constant-pressure conditions at inlet and outlet boundaries as described by Maier et al.[7] All boundaries were along the orthogonal axes, and no effort was made to follow curved boundaries with higher order approximations.[12] The wall-jet simulation required a minor modification of the Maier algorithm to handle the curved pressure boundary at the outlet. The collision parameter, τ, was fixed at 1 and the Mach number was kept below 0.2 in all simulations shown here. Full source code is provided in the Supporting Information. Execution speed was 1.56 s per iteration for a 64 × 64 × 64 cell simulation, or ~6 μs per cell per iteration. An additional 530 s was required to transfer data between Mathematica and the C engine under these conditions. This overhead was independent of the number of iterations and scaled linearly with the number of cells used.

Accurate software implementation of the LB algorithm was tested by simulating body-force or pressure-driven flow in a cylindrical duct with no-slip conditions at the walls (data not shown). This produced a parabolic flow profile with zero velocity at the walls for all systems tested (up to 64 × 64 × 256). As with all simulations, attention must be paid to mesh size, boundary effects, and other simulation parameters in order to achieve accurate results.

(11) Argentini, R.; Bakker, A. F.; Lowe, C. P. *Future Generation Computer Systems* **2004**, *20*, 973−980.
(12) Mei, R.; Luo, L.-S.; Shyy, W. *J. Comput. Phy.* **1999**, *155*, 307−330.

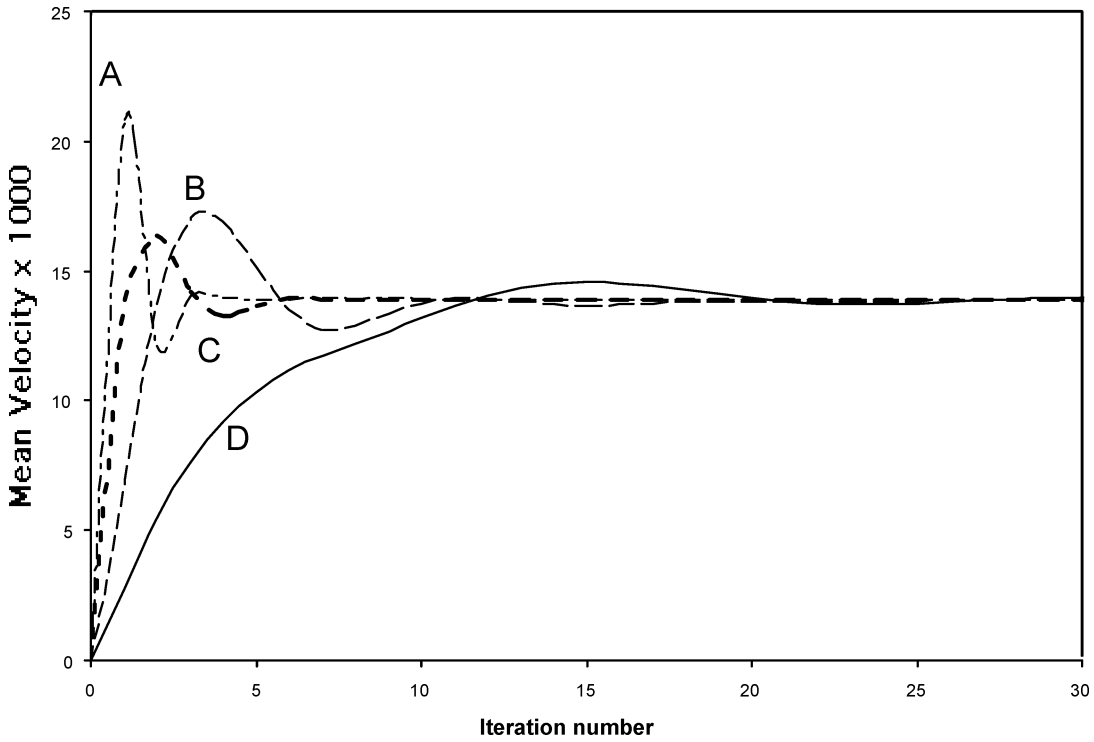

Figure 2. Relaxation of simulated flow velocity through a rectangular duct containing a sphere. Flow induced by body force with periodic boundaries. The proportional, integral, and differential gains for the PID loop were as follows: (A) 0.15, 0.01, 0.01; (B) 0.05, 0.001, 0.05; (C) 0.10, 0.02, 0.02; (D) 0.02, 0.01, 0.02; respectively. The duct was 25 × 25 × 25 cells, and the sphere radius was 6 cells.

Fast Body Force Relaxation. Flow can be induced by several means: applying pressure at the boundaries, moving the walls (e.g., Couette flow), or applying a constant force ("body force") to each fluid cell in the direction of flow. The latter approach is widely used due to its simplicity and because it facilitates periodic boundary conditions, which are essential in simulating a portion of a larger system such as a bead bed.[3,4] The body force accelerates the fluid during each iteration, an effect that is counteracted by frictional force at the stationary boundaries. Relaxation to a steady state occurs when the body force balances the frictional force and the fluid velocity and momentum become constant. The relaxation time is a function of the square of the mean distance between boundaries,[13] and it may take tens or hundreds of thousands of iterations to reach steady state in larger systems. This is a serious limitation of LB methods, and several authors have addressed it. Verberg and Ladd[14] reformulated the time-dependent LB equations at each node to produce a matrix of time-independent simultaneous equations. This matrix of equations was solved (once) by conventional techniques to give the steady-state solution. This "matrix method" reduced computation time by 10−100-fold, but at considerable cost in program complexity, and no use of the method has been reported since. Kandhai et al.[13,15] developed a method termed "iterative momentum relaxation" (IMR), which also accelerated relaxation 10−100-fold. The author's described their approach as "adjusting the applied body-force during the iteration depending on the change of fluid momentum at the iteration step considered." In practice, a body force much larger than that required for the steady state is initially applied, and the force is reduced in an ad hoc manner as the total fluid momentum approaches the steady-state value. In essence, the method is a sort of proportional control loop adjusting the body force to reach a specified momentum value. Minor deficiencies of the IMR method are the need for prior knowledge of the starting body force and the fact that the body force exhibits considerable oscillation about the set point. A major deficiency is the inability to specify a target velocity, which is generally far more useful in simulations than a specified momentum. Both deficiencies can be corrected by adding a software proportional-integral-derivative (PID) control loop[16] to the simulator, which rapidly relaxes the system to a specified mean or maximum velocity with minimal oscillation. The computational cost is negligible since velocity is normally calculated during the collision step of each iteration. It is necessary to tune the PID loop by performing a number of iterations with a constant body force in order to estimate the gain and response time of the system. Based on these values, PID parameters can be calculated to optimize relaxation. Additional simulations with different body forces can then be performed using the PID loop to reduce the simulation time by 2 orders of magnitude. Figure 1 shows the dramatic improvement in relaxation time obtained with the PID loop compared to the fixed body force approach. Figure 2 shows response curves obtained for several gain values during tuning of the loop.

Hybrid Scheme for Convection−Diffusion with Surface Reactions. Many important analytical processes involve surface

(13) Kandhai, D.; Koponen, A.; Hoekstra, A.; Kataja, M.; Timoonen, J.; Sloot, P. M.A. *J. Comput. Phys.* **1999**, *150*, 482−501.
(14) Verberg, R.; Ladd, A. J. C. *Phys. Rev. E.* **1999**, *60*, 3366−73.
(15) Kandhai, D.; Koponen, A.; Hoekstra, A.; Sloot, P. M.A. *Future Generation Comput. Syst.* **2001**, *18*, 89−96.
(16) Fraser, C. J.; Milne, J. S.; Noltingk, B. In *Mechanical Engineer's Reference Book*; Smith, E. H., Ed.; Butterworth-Heinemann: Oxford, UK, 1994; pp 3/40−3/75.

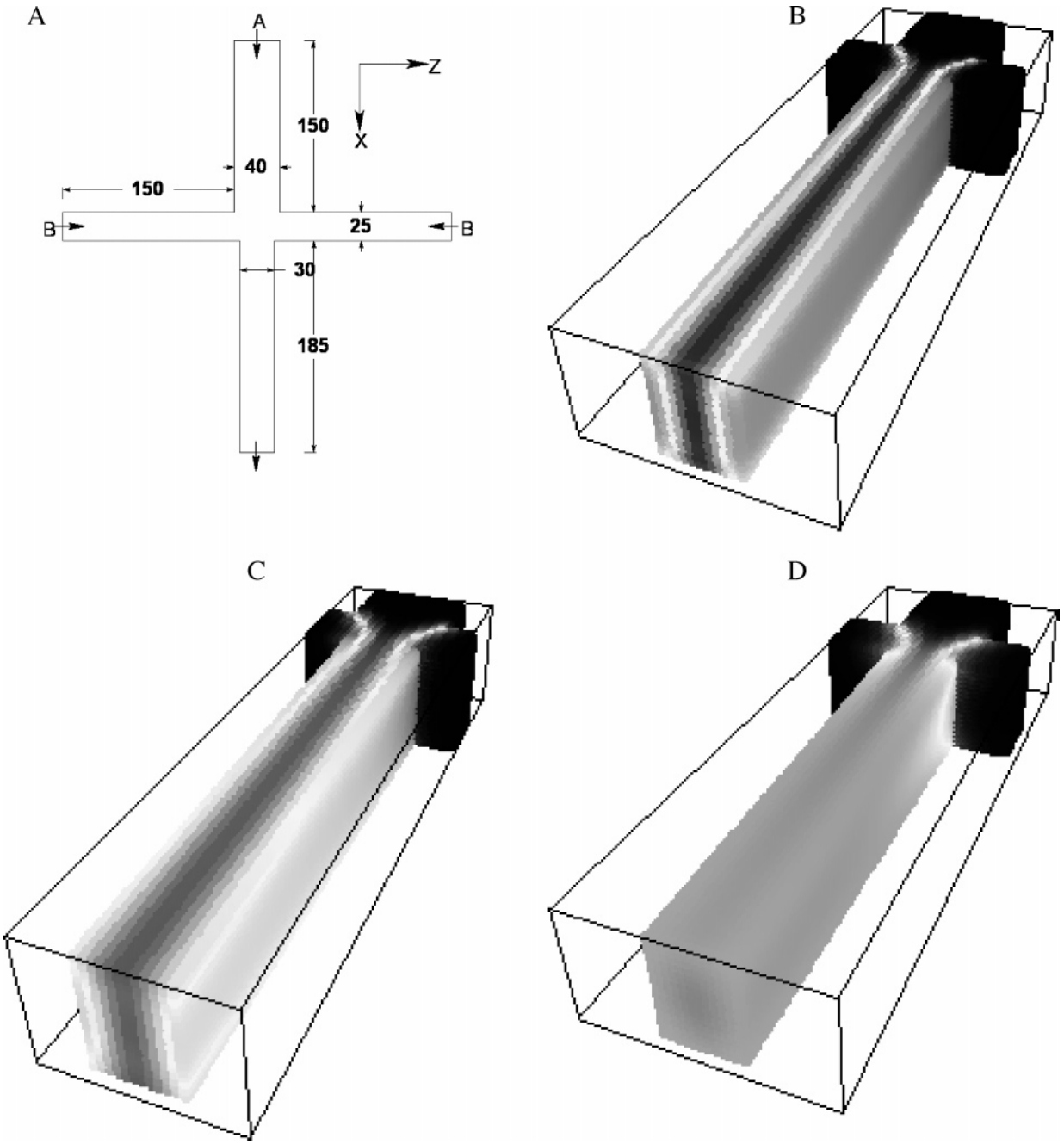

Figure 3. Simulation of a microfluidic cross-mixer[29] using hybrid convection−diffusion algorithm with varying diffusion coefficients. Overall model dimensions: 145 × 137 × 17 cells. Overall physical dimensions as shown. Pressure-driven flow with constant-pressure boundaries at inlet and outlet. Shading indicates the mole fraction of liquid A on a nonlinear scale, with black indicating a mole fraction of 1 and white indicating a mole fraction of 0.5. Only portions of the inlet arms near the mixing point are plotted. (A) Top view of micromachined cross-mixer. Lateral dimensions (μm) of the channels are shown. All channels were 40 μm deep. (B) Lattice diffusion coefficient 1×10^{-6}. (C) Lattice diffusion coefficient 3×10^{-3}. (D) Lattice diffusion coefficient 8×10^{-3}.

reactions coupled with convection−diffusion, including binding of DNA to microarrays, affinity capture of analytes on sensors, and electrochemical detectors. The ability of LB methods to accommodate complex boundaries makes them very attractive for modeling integrated systems containing mixers, reactors, separation devices, and detectors. While diffusion has been widely investigated in LB simulations of mixing and dispersion,[17,18] relatively few workers have investigated solution reactions,[19,20] and only a few reports discuss surface reactions.[21−24] For steady-state flows, the effect of diffusion has been determined by first calculating the flow field and then simulating the trajectory of a tracer undergoing a random walk while moving with the fluid.[3,25]

(17) Cali, A.; Succi, S.; Cancelliere, A.; Benzi, R. *Phys. Rev. A* **1992**, *45*, 5771−75.
(18) O'Brien, G. S.; Bean, C. J.; McDermott, F. *J. Hydrol.* **2002**, *268*, 143−157.
(19) Jorg, R. Weimar, J. R., Boon, J. P. *Physica A* **1996**, *224*, 207−215.
(20) Yu, H.; Luo, L-S.; Girimaji, S. S. *Int. J. Comput. Eng. Sci.* **2002**, *3*, 73−87.
(21) Wellsa, J. T.; Janeckyb, D. R.; Travisc, B. J. *Physica D* **1991**, *47*, 115−123.
(22) He, X. Y.; Li, N.; Goldstein, B. *Mol. Simul.* **2000**, *25*, 145−156.
(23) Kang, Q.; Zhang, D.; Chen, S.; He, X. *Phys. Rev. E* **2002**, *65*, 036318.
(24) Verhaeghe, F.; Arnout, S.; Blanpain, B.; Wollants, P. *Phys. Rev. E* **2005**, *72*, 036308.

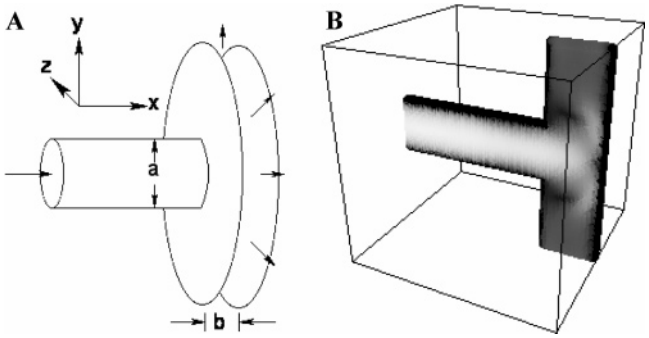

Figure 4. Wall-jet simulation. (A) Schematic diagram of the wall-jet system. Fluid enters axially from the left through a tube with diameter *a* and flows transversely over the walls at the right through a gap of height *b*. The length of the tube and diameter of the walls are determined by the number of cells used in the simulation. (B) Cross section of simulated wall jet system. Overall dimensions, 165 × 165 × 165 cells. Body force driven flow with periodic boundaries. Jet diameter, 41 cells. Jet gap, 41 cells. Black denotes zero velocity, and white denotes maximum velocity.

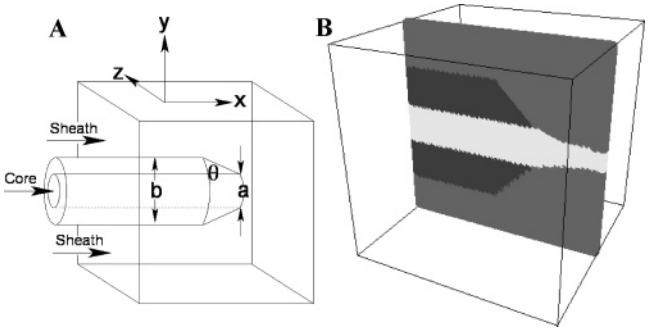

Figure 5. Sheath-flow simulation. (A) Schematic diagram of the sheath flow system. The nozzle inner diameter is *a*, the nozzle thickness is *b*, nozzle angle is θ, and the dimensions of the outer duct are determined by the number of cells used in the simulation. (B) Cross section of simulated sheath-flow system. Overall dimensions, 128 × 128 × 128 cells. Pressure-driven flow with fixed-pressure boundaries at inlet and outlet. Nozzle core diameter, 25 cells. Nozzle outer diameter, 78 cells. Nozzle angle, 45°. Dark gray is nozzle, medium gray is sheath fluid, and light gray is core fluid. Core fluid was traced with a trajectory tracking algorithm.

The trajectories of many tracers are computed to determine the effect of diffusion. Reaction and diffusion have also been studied by incorporating these effects within the LB framework.[26−28] One version of this approach uses a modified lattice-Boltzmann equation, which includes diffusive terms in the streaming step and reactive terms in the collision step. Other versions use the conventional (convection-only) LB equation for the solvent and add modified equations for the solutes. Generally, each component of the system requires a set of vectors for storing a set of densities, increasing memory usage considerably. While incorporating diffusion and reaction into the LB equations is an elegant and consistent approach, it does not allow the use of established finite-difference code and expertise for simulating convection−diffusion and surface reaction. The author therefore developed a hybrid scheme that coupled a conventional LB algorithm for solvent convection with an explicit finite-difference algorithm for diffusion and reaction. The convection−diffusion equation

$$\partial C/\partial t = D\nabla^2 C - \nabla vC$$

is replaced with the explicit finite-difference form:

$$\frac{\Delta C^i}{\Delta t} = D\,\frac{C^i_{x+1} - 2C^i_x + C^i_{x-1}}{\Delta u^2} - v_x\frac{C^i_{x+1} - C^i_{x-1}}{\Delta u}$$

$$\frac{\Delta C^i}{\Delta t} = D\frac{C^i_{x+1} - 2C^i_x + C^i_{x-1}}{\Delta u^2} - v_x\,\frac{C^i_{x+1} - C^i_{x-1}}{2\Delta u} - C^i_x\,\frac{\mathrm{v}_{x+1} - \mathrm{v}_{x-1}}{2\Delta u}$$

for each component *i* and each direction *u* (*x*,*y*,*z*). After streaming and collision of the solvent, this equation is solved for the new concentration of each component C^i using the local velocity of the solvent. This scheme is local and requires only one additional storage element per component. While further study is required to establish the stability and accuracy of this hybrid approach, results from simulations of diffusion in a microfluidic mixer are promising. The cross-mixer studied by Wong et al.[29] was simulated at domain sizes of 360 × 340 × 40 cells (flow only) and 145 × 137 × 17 (flow and diffusion). The degree of mixing was strongly affected by the diffusion coefficient, as shown in Figure 3. The microfluidic mixer with projections[29] was also simulated successfully with this approach and showed the expected improvement in mixing efficiency (data not shown).

New Flow Systems. The ability of this software package to readily generate complex boundary conditions in multiple dimensions and correctly perform calculations for such complex geometries is illustrated by simulations of wall-jet and sheath-flow systems. Neither of these systems has been simulated in the LB literature. Boundary conditions for these systems are established with a single function call (InitWallJet, InitSheathFlow) utilizing only the dimensions of the domain and a few parameters. The wall-jet or impinging-jet system consists of two semi-infinite plane surfaces separated by a gap of height *b*. Liquid emerges from an opeing with diameter *a* in one of the walls and forms a jet that impinges at right angles on the opposite wall. The wall-jet produces a high flux at the impinging surface and is frequently used in electrochemical[30] and colloidal[31] studies, but simulation results have been limited. The geometry of the system is shown in Figure 4A, and simulation results are shown in Figure 4B. The sheath-flow geometry is widely used in flow cytometers as well as related microfluidics devices.[5,32] A nozzle with inner diameter *a*, outer diameter *b*, and convergence angle θ is centered in a rectangular

(25) Jimenez-Hornero, F. J.; Giraldez, J. V.; Laguna, A. *Vadose Zone J.* **2005**, *4*, 310−16.
(26) Flekkøy, E. G. *Phys. Rev. E* **1993**, *47*, 4247−4257.
(27) Flekkøy, E. G.; Oxaal, U.; Feder, J. Jøssang, T. *Phys. Rev. E* **1995**, *52*, 4952−4962.
(28) Ponce, Dawson, S.; Chen, S.; Doolen, G. D. *J. Chem. Phys.* **1993**, *98*, 1514−1523.
(29) Wong, S. H.; Bryant, P.; Ward, M. Wharton, C. *Sens. Actuators B* **2003**, *95*, 414−424.
(30) Alden, J. A.; Hakoura, S.; Compton, R. G. *Anal. Chem.* **1999**, *71*, 827−836.
(31) Adamczyk, Z.; Warszynski, P.; Szyk-Warszynska, L.; Weronski, P. *Coll. Surf. A* **2000**, *165*, 157−187.
(32) Schrum, D. P.; Culbertson, C. T.; Jacobson, S. C.; Ramsey, J. M. *Anal. Chem.* **1999**, *71*, 4173−4177.
(33) Chun, H.; Chung, T. D.; Kim, H. C. *Anal. Chem.* **2005**, *77*, 2490−2495.

or cylindrical duct, which carries a flow of sheath liquid. Liquid emerging from the nozzle forms a distinct cylindrical core completely surrounded by the sheath liquid. The geometry and simulation results for the sheath-flow system are shown in Figure 5A and B, respectively.

Using the LB Simulation Package. The simulation package is designed to be easy to learn and use. The C engine can generally be treated as a "black box"—the user need only define the simulation problem and boundary conditions in Mathematica, call the C engine with the appropriate parameters, and display/analyze the data in Mathematica. There are predefined Mathematica functions for setting up and displaying many flow geometries, and these can be readily modified or combined for greater flexibility. Coding has emphasized clarity over efficiency, and comments and descriptive variable names are used for this purpose. Full source code and demonstration routines are provided in the Supporting Information and are available from the author.

CONCLUSIONS

The processing speed and memory capacity of a modern desktop computer was adequate to perform lattice-Boltzmann fluid dynamics simulations of large-scale ($128 \times 128 \times 128$) three-dimensional systems in reasonable times (hours), particularly when a PID loop was used to speed relaxation. Initial results with a hybrid method for simulating convection−diffusion were promising, indicating that the LB method could be coupled with a conventional finite-difference scheme and allowing existing code and expertise to be used to solve new problems. The combination of a specialized LB computation engine for simulation with a general-purpose technical computing package for setup and data display provides a cross-platform system that is easy to learn and use.

ACKNOWLEDGMENT

Software provided by Prof. Alan Ladd and helpful discussions with Dr. Robert S. Maier and Dr. Xiaoyi He are gratefully acknowledged. Mention of brand or firm names does not constitute an endorsement by the U.S. Department of Agriculture over others of a similar nature not mentioned.

SUPPORTING INFORMATION AVAILABLE

Additional information as noted in text. This material is available free of charge via the Internet at http://pubs.acs.org.

Received for review November 17, 2006. Accepted January 29, 2007.

AC062178V