Comput Geosci (2014) 18:17–27
DOI 10.1007/s10596-013-9379-6

ORIGINAL PAPER

Taxila LBM: a parallel, modular lattice Boltzmann framework for simulating pore-scale flow in porous media

Ethan T. Coon · Mark L. Porter · Qinjun Kang

Received: 25 January 2013 / Accepted: 10 October 2013 / Published online: 31 October 2013

Abstract The lattice Boltzmann method is a popular tool for pore-scale simulation of flow. This is likely due to the ease of including complex geometries such as porous media and representing multiphase and multifluid flows. Many advancements, including multiple relaxation times, increased isotropy, and others have improved the accuracy and physical fidelity of the method. Additionally, the lattice Bolzmann method is computationally very efficient, thanks to the explicit nature of the algorithm and relatively large amount of local work. The combination of many algorithmic options and efficiency means that a software framework enabling the usage and comparison of these advancements on computers from laptops to large clusters has much to offer. In this paper, we introduce *Taxila LBM*, an open-source software framework for lattice Boltzmann simulations. We discuss the design of the framework and lay out the features available, including both methods in the literature and a few new enhancements which generalize methods to complex geometries. We discuss the trade-off of accuracy and performance in various methods, noting how the Taxila LBM makes it easy to perform these comparisons for real problems. And finally, we demonstrate a few common applications in pore-scale simulation, including the characterization of permeability of a Berea sandstone and analysis of multifluid flow in heterogenous micromodels.

Keywords Pore-scale simulation · Lattice Boltzmann method · Software framework

1 Introduction

Modeling pore-scale flow in geologic materials is increasingly important for applications of carbon sequestration, hydrofracking, oil recovery, microfluidics, and more. While simulations are often done at the continuum scale on which Darcy's equation is valid, increasingly pore-scale effects are being recognized as important. These pore-scale effects are often used to characterize the media (calculating properties for use at the continuum scale) or understand multiphase interfacial physics. Pore-scale fluid flow is a difficult problem, as it implies the solution of Navier–Stokes equations for fluid dynamics within a complex geometry of interconnected pore space. One successful alternative to standard computational fluid dynamics (CFD) methods for these problems is the lattice Boltzmann method (LBM). The LBM is a kinetic theory based upon a discrete Boltzmann equation for distributions of particles on a lattice. The success of the LBM is partly due to the ease of including complex boundaries such as porous media and partly due to heuristic methods for including fluid–fluid interfacial effects, which avoid the need for special treatments usually required by standard CFD methods [4, 19, 22].

Another feature of the LBM is that it is computationally localized and explicit, making it scale extremely well on large parallel machines. This is both an advantage (as it maps well to high-performance computing and modern

E. T. Coon (✉)
Applied Mathematics and Plasma Physics, Los Alamos National Laboratory, MS B284, Los Alamos, NM 87544, USA
e-mail: ecoon@lanl.gov

M. L. Porter
Earth Systems Observations, Los Alamos National Laboratory, MS B284, Los Alamos, NM 87544, USA

Q. Kang
Computational Earth Sciences, Los Alamos National Laboratory, MS B284, Los Alamos, NM 87544, USA

architectures such as GPUs, enabling the solution of real-world problems) and a disadvantage (as it can result in very slow convergence to steady-state solutions as compared to implicit methods). The LBM is split between a streaming step, which involves communication but little or no computation, and a collision and forcing step, which involves computation but little or no communication. Therefore, the latter step, but also the algorithm on the whole, scales very well on the largest computers. At extreme scales (tens to hundreds of thousands of cores), communication and i/o routines can start to dominate simulation time. Therefore, a computational framework which builds upon existing libraries with advanced strategies for communication and i/o has much to offer, greatly increasing the size of problems which are accessible to users.

Both a feature and downside of the LBM is the large number of specialized variations and advanced algorithms available in the literature. Many advancements of the LBM introduce more computation and wider communication patterns to the collision step while correcting physical inconsistencies and improving accuracy. For instance, [11] shows that careful treatment of fluid–fluid interfacial terms result in viscosity-independent surface tensions, improving accuracy. This comes at the expense of increased computational cost. This is typical of improved methods and indicates a spectrum of trade-offs between accuracy and computational performance. On identical geometries with identical resolution, very different accuracy and performance are expected as more or different terms and methods are used.

With these trade-offs in mind, we have designed Taxila LBM, an open-source research code for solving the LBM on complex geometries typical of geologic media. The code is designed to be modular and easily extensible, so as to enable algorithm researchers to add and improve methods and easily compare to existing methods. This versatility also makes it easy for domain scientists to test and compare various advancements, finding the ideal trade-off between computation and performance for their problem. And finally, it is designed from the ground up for performance and scalability, allowing a code developed and tested on laptops to scale to tens to hundreds of thousands of cores.

This paper is primarily intended to introduce the open-source lattice Boltzmann simulator Taxila LBM and its flexibility and performance for both application scientists and method developers. In the course of generalizing several algorithms in the literature for Taxila LBM, we have also introduced a few minor algorithmic changes regarding boundary conditions and fluid–fluid interfacial forces in the presence of a solid. In Section 2, we summarize the lattice Boltzmann methods supported by Taxila LBM and specify these changes. The full range of methods are characterized by their relaxation and forcing terms and include boundary and initial conditions. In Section 3, we discuss the major design principles of Taxila LBM. Taxila LBM is designed to be modular and easily extensible, allowing easy comparison of accuracy and performance of multiple methods on the same problem. This is demonstrated by a measure of the trade-offs between accuracy and performance for simple bubble tests. We discuss the parallelization approach of Taxila LBM and demonstrate scaling results. Finally, Section 4 exercises the features of Taxila LBM through a series of demonstrations of common applications.

2 The lattice Boltzmann method

The lattice Boltzmann (LB) class of methods solves the discrete Boltzmann equation for an ensemble-averaged distribution of particles, $f_i^k(\mathbf{x}, t) = f^k(\mathbf{x}, \mathbf{e}_i, t)$, moving on a discrete lattice of sites $\mathbf{x}$ at time t with fixed velocities $\mathbf{e}_i$ corresponding to movements from site to site. Here, k indexes each of n_k components, where i indexes each of n_i available velocity directions (plus the stationary direction, $i = 0$). A discrete lattice is typically specified by the dimension and number of available velocity vectors. Currently, Taxila LBM supports the two most common lattices: D2Q9 (two-dimensional, nine lattice velocities, see Fig. 1) and D3Q19 (three-dimensional, 19 lattice velocities), but is easily extensible to arbitrary lattices.

Given a lattice, the distribution function is evolved using the discrete Boltzmann equation, typically in the form

$$f^k(\mathbf{x}+\mathbf{e}_i\Delta t, \mathbf{e}_i, t+\Delta t) - f^k(\mathbf{x}, \mathbf{e}_i, t) = \Omega^k_{\text{coll}} + \Omega^k_{\text{forces}}. \quad (1)$$

Here, the equation includes three terms: the left-hand side describes the time-evolution of parcels traveling for one time unit at lattice velocities $\mathbf{e}_i$ (also known as the streaming step), Ω_{coll} provides momentum changes in the particle distribution due to collisions, and Ω_{forces} provides momentum changes in the particle distribution due to other forces.

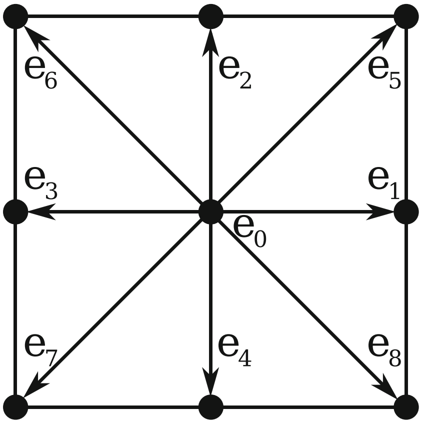

Fig. 1 The D2Q9 lattice and its available directions, $\mathbf{e}_i$

While the streaming step is standard across all LB methods, the collision and forcing operators vary greatly from method to method. Here, we follow the method as formulated in [11]. In this formulation, the collision operator is defined by

$$\Omega^k_{\text{coll}} \equiv \mathbf{\Lambda}^k \left[f_i^{\text{eq,k}}(\mathbf{x}, t) - f_i^k(\mathbf{x}, t) \right] \tag{2}$$

where $\mathbf{\Lambda}^k$ is the relaxation operator and $f_i^{\text{eq,k}}$ is the equilibrium distribution function, expressed as follows [13]:

$$f_i^{\text{eq,k}} \equiv w_i \rho_k \left[1 + \frac{\mathbf{e}_i \cdot \mathbf{u}^{\text{eq}}}{c_s^2} + \frac{(\mathbf{e}_i \cdot \mathbf{u}^{\text{eq}})^2}{2c_s^4} + \frac{\mathbf{u}^{\text{eq}} \cdot \mathbf{u}^{\text{eq}}}{2c_s^2} \right], \tag{3}$$

where w_i are weights specific to the chosen lattice (see [1], Table 1) and $\mathbf{u}^{\text{eq}}$ is a relaxation operator-dependent effective velocity chosen to conserve momentum in the absence of forces (see [11], (23)).

Forcing terms are incorporated through the forcing operator, given by

$$\Omega^k_{\text{forces}} \equiv \frac{\Delta t}{2} \left[f_i^{F,k}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) + f_i^{F,k}(\mathbf{x}, t) \right], \tag{4}$$

where

$$f_i^{F,k} \equiv \frac{\mathbf{F}^k \cdot (\mathbf{e}_i - \mathbf{u}^{\text{eq}})}{\rho_k c_s^2} f_i^{\text{eq,k}}. \tag{5}$$

Here, $\mathbf{F}^k$ is a vector of forces, c_s is the speed of sound, ρ_k is the macroscopic number density, and $\mathbf{u}_k$ is the macroscopic fluid velocity of component k.

The transformation $\bar{f}_i^k = f_i^k - \frac{\Delta t}{2} f_i^{F,k}$ [6, 7] is applied to Eq. 1 to yield an explicit scheme expressed as follows:

$$\bar{f}_i^k(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - \bar{f}_i^k(\mathbf{x}, t) \\ = \mathbf{\Lambda}^k \left[f_i^{\text{eq,k}}(\mathbf{x}, t) - \bar{f}_i^k(\mathbf{x}, t) - \frac{\Delta t}{2} f_i^{F,k} \right] + \Delta t f_i^{F,k}. \tag{6}$$

Given this transformation, the macroscopic properties for density and momentum of each component are defined as follows:

$$\rho_k = \sum_i \bar{f}_i^k, \tag{7}$$

$$\rho_k \mathbf{u}_k = \sum_i \bar{f}_i^k \mathbf{e}_i + \frac{\Delta t}{2} \mathbf{F}^k, \tag{8}$$

and the total velocity of the fluid mixture is

$$\mathbf{u} = \frac{\sum_k \rho_k \mathbf{u}_k}{\sum_k \rho_k}. \tag{9}$$

2.1 Relaxation operators

The relaxation operator $\mathbf{\Lambda}^k$ is an important consideration in any LB method. The single relaxation time method (SRT) uses a single scalar relaxation time τ for each fluid component,

$$\mathbf{\Lambda}^k = \frac{1}{\tau_k}. \tag{10}$$

This relaxation time is related to the kinematic fluid viscosity (in lattice units) by $\nu_k = (\tau_k - 0.5)/3$. This approach is computationally extremely efficient and, with appropriate treatment of the forcing terms, can capture much of the qualitative physics.

Alternatively, multiple relaxation time methods (MRT) [9, 10, 12] enable the relaxation of different moments of the distribution function independently. In this form, the relaxation operator is given by a full matrix operator:

$$\mathbf{\Lambda}^k = \mathbf{T}^{-1} \tilde{\mathbf{\Lambda}}^k \mathbf{T} \tag{11}$$

where $\mathbf{T}$ is a transformation matrix which maps the distribution function to its moments (see [10]) and $\tilde{\mathbf{\Lambda}}^k$ is a diagonal matrix of relaxation times corresponding to each moment. In this way, conserved moments are relaxed independently of the viscosity, resulting in significantly reduced spurious currents [10] and therefore more accurate simulations. This also enables higher viscosity ratios in multicomponent/multiphase simulations.

In Taxila LBM, both MRT and SRT are available by command line or input file options, and relaxation times are specified for each component.

2.2 Forces on the distribution function

The forcing term $\mathbf{F}^k$ can capture many physical effects, including body forces such as gravity, internal forces such as inter-particle potentials (surface tension), and cohesive forces to solid geometry (fluid wettability). Currently, Taxila LBM provides these three forces and is easily extended to others.

2.2.1 Gravity

Body forces including gravity are given by including a scalar force, i.e.,

$$\mathbf{F}^k = \rho_k \mathbf{g}. \tag{12}$$

In Taxila LBM, body forces can be specified in any direction of any magnitude on the command line or in an input file.

2.2.2 Fluid–solid forces

Cohesive forces to solid geometry are included through a force of the following form:

$$\mathbf{F}^k = -\rho_k(\mathbf{x}) \sum_{m=1}^{n_m} g_k^m \sum_{\mathbf{x}' \in \mathcal{W}_m} (\mathbf{x}' - \mathbf{x}). \tag{13}$$

where g_k^m indicates fluid component k's affinity for mineral m and $\mathcal{W}_m$ indicates the set of neighboring sites $\mathbf{x}+\mathbf{e}_i$ which represent a wall node of mineral m. In multicomponent or multiphase simulations, this yields a contact angle, where positive g_k^m indicates a non-wetting fluid and negative g_k^m indicates a wetting fluid.

In Taxila LBM, fluid component affinities for each mineral type are specified on the command line or in an input file.

2.2.3 Fluid–fluid forces

Interparticle potential forces, as introduced in [16, 17], enable phase separation and surface tension to be included easily in the LBM. This likely explains the popularity of this LBM variation. In the original paper, this force is given by the gradient of an interaction potential $\mathcal{V}$:

$$\mathcal{V}(\mathbf{x},\mathbf{x}') = \mathcal{G}_{k,\bar{k}}(|\mathbf{x}'-\mathbf{x}|)\psi_k(\mathbf{x})\psi_{\bar{k}}(\mathbf{x}') \tag{14}$$

where ψ is an effective mass function used to enable equations of state, and $\mathcal{G}_{k\bar{k}}$ is a Green's function for interaction forces between components k and $\bar{k}$ used to induce phase separation in single-component systems and mimic surface tension in multicomponent systems.

Green's function falls off quickly as distance $\left|\mathbf{x}'-\mathbf{x}\right|$ increases. In the simplest case, only nearest neighbors are chosen,

$$\mathcal{G}_{k,\bar{k}} = \begin{cases} 0 & \left|\mathbf{x}'-\mathbf{x}\right| > c \\ g_{k,\bar{k}} & \left|\mathbf{x}'-\mathbf{x}\right| = c \end{cases} \tag{15}$$

where the $g_{k,\bar{k}}$ are constants. Then, this reduces to the commonly seen form:

$$\mathbf{F}_k(\mathbf{x}) = -\nabla\mathcal{V}(\mathbf{x}) = -\psi_k(\mathbf{x})\sum_{\bar{k}=1}^{n_k} g_{k,\bar{k}}\sum_{i=0}^{n_i}\psi_{\bar{k}}(\mathbf{x}+\mathbf{e}_i)\mathbf{e}_i \tag{16}$$

for a system of n_k components on a lattice with n_i directions. We note that this is valid only in the case of no nearby wall sites.

For increased accuracy, others have proposed long-range pseudo-potential functions which include more neighbors [18, 21] in Green's function. These methods work by increasing the order of isotropy in the gradient of the pseudo-potential; see [21] and [14] for detailed discussions of these advances. This improved accuracy comes at the expense of an increased stencil size, which increases both communication size and computational work for the forcing term. Additionally, these are typically described only in the absence of a solid; this must be generalized for use in the porous media.

To encompass all of these approaches, we use the more general form; forces are given by a pseudo-potential function [21]:

$$\mathbf{F}_k(\mathbf{x}) = -c_0\psi_k(\mathbf{x})\sum_{\bar{k}}\mathcal{G}_{k\bar{k}}\nabla\psi_{\bar{k}}(\mathbf{x}). \tag{17}$$

Then the gradient operator is discretized using a summation of one-sided differences:

$$\mathbf{F}_k(\mathbf{x}) = -\nabla\mathcal{V}(\mathbf{x}) = -\psi_k(\mathbf{x}) \times\sum_{\bar{k}=1}^{n_k}\sum_{\mathbf{x}'\in\mathcal{N}_\mathbf{x}}\mathcal{G}_{k,\bar{k}}\left(\left|\mathbf{x}'-\mathbf{x}\right|\right)\frac{\psi_{\bar{k}}(\mathbf{x}')-\psi_{\bar{k}}(\mathbf{x})}{|\mathbf{x}'-\mathbf{x}|}\left(\mathbf{x}'-\mathbf{x}\right). \tag{18}$$

This summation is taken over a set of lattice sites $\mathcal{N}_\mathbf{x}$ which do not include wall sites and whose direct path from $\mathbf{x}$ to $\mathbf{x}'$ is not blocked by wall sites. This is critical in the case of long-range interactions, which should not act through walls. We note that in the simplest case of no nearby walls and a Green's function given by Eq. 15, cancelation ensures that Eq. 18 reduces to Eq. 16.

In Taxila LBM, three isotropy orders are available, and appropriate Green's functions are scaled by a single constant for each component pair. In this way, $\mathcal{G}_{k,\bar{k}} = g_{k,\bar{k}}\widehat{\mathcal{G}}$, where $\widehat{\mathcal{G}}$ enforces isotropy and spatial extent while the user-provided $g_{k,\bar{k}}$ specifies strength of the forces and equivalently the surface tension.

Note that this approach for multicomponent simulations is validated using Laplace's law, as shown in [11].

The generality of this approach is demonstrated in Fig. 2. In this simulation, three immiscible fluids are introduced in a configuration with two bubbles surrounded by a third fluid. The bubbles move along a solid surface due to an applied body force of 1 × 10^{-4} in the x-direction. In this case, the green and red fluids are non-wetting to the solid

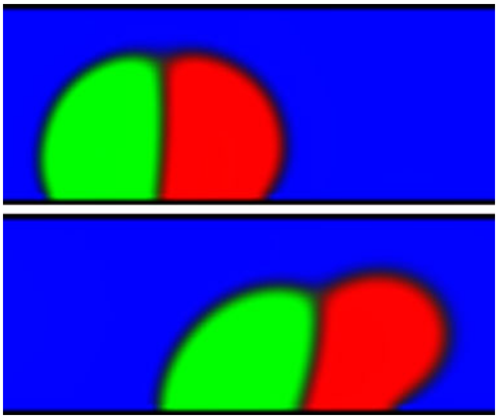

Fig. 2 Two snapshots in time from a simulation of three immiscible fluids under body forces from left to right. Contact angle hysteresis is seen at the advancing and receding fronts as the non-wetting bubbles advance along the wall

with $g_k^m = 0.10$, whereas the blue fluid is wetting with $g_k^m = -0.10$. The fluid–fluid force between each fluid pair is $g_{k,\bar{k}} = 0.26$. As the bubbles move to the right they exhibit contact angle hysteresis with different advancing and receding contact angles.

We note that while Taxila LBM currently only implements this variation of the Shan and Chen LBM for multicomponent flows, it is designed to support any method that can fit the form of Eq. 1. This indicates that other methods such as the free-energy [20] and mean-field [8] methods could be easily implemented within the Taxila LBM framework.

2.3 Complex geometries

One of the main advantages of the LB method is its ease in incorporating complex geometries. Assuming a no-slip boundary condition on the solid wall boundary, lattice sites are labeled as wall sites and a bounce-back condition is applied. The true boundary is assumed to exist at the halfway point between wall sites and fluid sites. Then, the streaming step is modified; distribution functions whose streaming step take them to a wall site are reversed in direction and placed back at the source fluid site. This approach is second order for aligned boundaries [5].

In Taxila LBM, geometry may be included via a precomputed binary or text-based file or through an initialization subroutine compiled into the simulator. The geometry may also include labels for mineral type of the wall structure. This enables the solution of problems including mixed wettabilities, such as that in Fig. 3. In this problem, a droplet is placed on a surface made of two minerals at the intersection of the two minerals. The fluid is chosen to be wetting to one mineral and non-wetting to the other. Cohesive forces cause the droplet to migrate onto the wetting portion of the surface.

2.4 Initial conditions

In the LBM, the discrete Boltzmann Eq. 1 is made well-posed by supplying initial and boundary conditions. A complete initial condition consists of each component of the distribution function, i.e., $f^k(\mathbf{x}, \mathbf{e}_i, 0)$. A typical application's initial condition may include the initial density and velocity of the fluid, but usually not the higher moments. Therefore, these moments are mapped into the distribution function space using the equilibrium function (3):

$$f^k(\mathbf{x}, \mathbf{e}_i, 0) = f_i^{\mathrm{eq,k}}(\rho, \mathbf{u}) \tag{19}$$

In Taxila LBM, initial conditions may be specified as a subroutine compiled into the simulator, as a set of files for densities and velocities or as a complete single file for the distribution function. Restart and checkpoint capabilities are implemented in which the total distribution function is saved to disk. This alone captures all the necessary information for continuing a simulation.

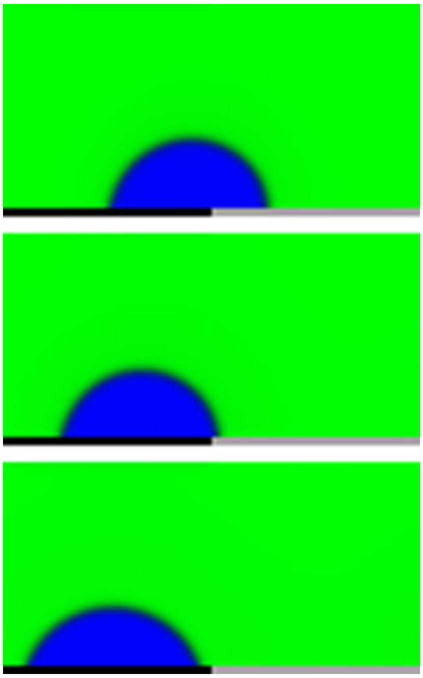

Fig. 3 Three timesteps in a multicomponent simulation of a droplet on a bi-material surface. The *blue* fluid is wetting on the *black* mineral and non-wetting on the *gray* mineral. Cohesive forces pull the droplet to the wetting surface

2.5 Boundary conditions

As in the case of initial conditions, boundary conditions are typically not completely specified. A complete set of boundary conditions must specify the incoming distribution functions for each component, i.e., all $f^k(\mathbf{x}, \mathbf{e}_i, t)$ such that $\mathbf{x}$ is on the boundary of the domain and $\mathbf{e}_i \cdot \widehat{\mathbf{n}} < 0$ where $\widehat{\mathbf{n}}$ is the outward normal of the domain boundary. This corresponds to $3n_k$ values for the D2Q9 lattice and $5n_k$ for the D3Q19 lattice. Typically, real problems are posed regarding the moments of the distribution function, however. For instance, an application may wish to specify pressure or velocity on the boundary. This problem is underconstrained, and some freedom is allowed in specifying the exact discrete distribution function.

To do this, Taxila LBM uses an extension of the approach of [3]. This extension enables multiple components and accounts for forcing terms at the boundary. In this approach, the incoming distribution functions are written as a sum of a predictor, given by a previous timestep's value, and a vector momentum corrector. This corrector is chosen to satisfy the required constraints on moments on the boundary.

Consider the case of the D2Q9 lattice on a boundary node whose outward normal is given by $\mathbf{e}_2$, as seen in Fig. 4. To provide data for the streaming step, inward distribution functions for each component must be specified. After streaming, these unknowns are seen in the collision and forcing terms as f_4^k, f_7^k, and f_8^k. We represent these unknowns as a sum of a predictor f^* $\left(\text{chosen here as } f_i^{k*} = f_i^k(\mathbf{x}, t - \Delta t)\right)$ and vector corrector $\mathbf{Q}$:

$$f_i^k = f_i^{k*} + \frac{\alpha_i}{c}\mathbf{e}_i \cdot \mathbf{Q}^k. \tag{20}$$

Then density and momentum can be written through Eqs. 7–8 as follows:

$$\begin{aligned}\rho_k &= f_0^k + f_1^k + f_2^k + f_3^k + \left(f_4^{k*} - \alpha_4 Q_y^k\right) \\ &\quad + f_5^k + f_6^k + \left(f_7^{k*} - \alpha_7\left(Q_x^k + Q_y^k\right)\right) \\ &\quad + \left(f_8^{k*} - \alpha_8\left(-Q_x^k + Q_y^k\right)\right)\end{aligned} \tag{21}$$

$$\begin{aligned}\rho_k u_k &= f_1^k - f_3^k + f_5^k - f_6^k - \left(f_7^{k*} - \alpha_7\left(Q_x^k + Q_y^k\right)\right) \\ &\quad + \left(f_8^{k*} - \alpha_8\left(-Q_x^k + Q_y^k\right)\right) + \frac{\Delta t}{2}\mathbf{F}_x^k\end{aligned} \tag{22}$$

$$\begin{aligned}\rho_k v_k &= f_2^k - \left(f_4^{k*} - \alpha_4 Q_y^k\right) + f_5^k + f_6^k \\ &\quad - \left(f_7^{k*} - \alpha_7\left(Q_x^k + Q_y^k\right)\right) \\ &\quad - \left(f_8^{k*} - \alpha_8\left(-Q_x^k + Q_y^k\right)\right) + \frac{\Delta t}{2}\mathbf{F}_y^k\end{aligned} \tag{23}$$

noting that we have dropped bars on the $\bar{f}_i$ for clarity. This system is $3n_k$ equations in $5n_k$ unknowns, ρ_k, u_k, v_k, Q_x^k, and Q_y^k (more generally, for dimension d, the system is of $(d+1)n_k$ equations for $(2d+1)n_k$ unknowns). Therefore, given $5n_k - 3n_k = 2n_k$ constraints on moments, the $\mathbf{Q}^k$ may be solved for the incoming f_i^k calculated, and the problem is closed.

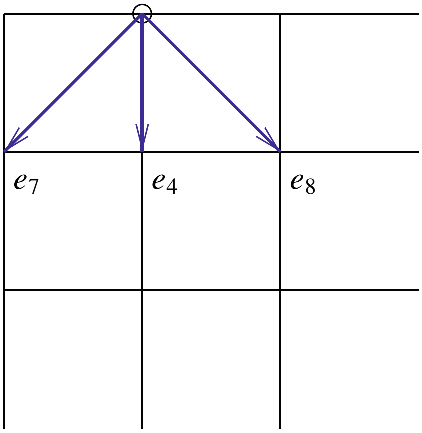

Fig. 4 Lattice diagram near a boundary. A full boundary condition, applied post-streaming, requires the specification of f_4^k, f_7^k, and f_8^k

Several choices are then available to provide these $2n_k$ constraints. Taxila LBM implements multiple modes in language more common to geologic media simulations for ease of use.

- *Density boundary conditions.* Each of the ρ_k are provided, and tangential sources of momentum are zero. This implies, for the above example with normal $\mathbf{e}_2$, that $\rho_k u_k = \frac{\Delta t}{2}\mathbf{F}_x^k$.
- *Pressure boundary conditions.* For the single-component case, ρ is directly calculated from pressure. For the multicomponent case, compositional fractions are also provided, i.e., $\rho_1/(\rho_1+\rho_2)$ for the two-component case. From these, densities are calculated and tangential momentum sources are again assumed zero.
- *Pressure outlet boundary conditions.* For a boundary on which outflow is expected a priori, a "zero-gradient" condition on compositional fractions may be assumed. Pressure is provided, and compositional fractions are taken from the node one lattice site toward the interior normal of the boundary.
- *Flux boundary conditions.* A flux of each component k is specified, which is placed into the normal momentum components. Tangential momentum sources are again assumed zero.
- *Flux outlet boundary conditions.* Like the pressure outlet boundary conditions, total outward flow is specified, and a zero-gradient condition is used on compositional fractions to determine the compositional fluxes.
- Additionally, periodic and symmetric boundary conditions are allowed.

Each of these modes are available on the command line for constant conditions or through compiled subroutines provided by the user for non-constant conditions.

3 Taxila LBM design

Taxila LBM is designed to support two main user bases: the algorithm developer and the application scientist. In developing for both audiences, we have designed a framework to enable the following:

- rapid prototyping and implementation of new algorithms,
- easy testing and comparison to existing methods,
- and a single implementation for laptops and supercomputers, meaning that new algorithms are immediately available for large problems with no additional work.

3.1 Design for rapid testing

The LBM is more accurately a class of methods. Each method is characterized by its collision operator and external force representation. Different choices in these algorithms result in different levels of physical fidelity (qualitative correctness), accuracy (quantitative correctness), and computational performance. As most applications of the LBM require many lattice sites to model the system of interest, this trade-off between correctness and performance is difficult to balance. The most accurate method may not always be the best choice, unless it is required to correctly capture the physics of interest. Furthermore, these trade-offs can vary greatly from problem to problem.

Through a modular, object-oriented approach, each of these algorithmic choices may be made on the command line, allowing the same problem to be solved using different algorithms quickly and easily. This approach facilitates both user models. Algorithm developers can easily run and compare identical problems with different algorithms by changing a single command line entry, testing how their approach compares to existing methods. Domain scientists can quickly compare accuracy and performance for multiple algorithms on their problem, making informed decisions before using the same code to scale their problem to the domain of interest.

For example, consider a comparison of trade-offs on a two-phase bubble test. Two critical choices in a bubble test are the choice of relaxation operator and the order of isotropy required of derivatives in the fluid–fluid potential force. By expanding the size of the stencil used to calculate the $\nabla\psi$ term in Eq. 18, a more accurate potential is applied.

We consider two relaxation operators—SRT and MRT as described in Section 2.1—along with fourth-, eighth-, and tenth-order isotropy as described in [14]. The MRT offers decreased spurious currents over SRT at the cost of significantly more computational resources but no increased communication. Increased isotropy offers decreased spurious currents at the cost of slightly more computational resources and an increased communication for derivative calculations.

The SRT requires one parameter for each component in the bubble test which is specified by supplying the options:

```
-relaxation_mode srt
-tau_component1 1.
-tau_component2 1.
```

The multiple relaxation time operator requires a parameter for each set of modes for each component. These relaxation times are provided by (for D2Q9):

```
-relaxation_mode mrt
-s_c_component1 1. # conserved moments
-s_e_component1 1. # energy moment
-s_e2_component1 1. # energy moment squared
-s_q_component1 1. # energy flux moments
-s_nu_component1 1. # viscous stresses
```

along with values for the second component. Note that default values are shown here; using these values would reduce identically to SRT. Typically, the conserved moments are kept with a relaxation time of one. The viscous stresses relaxation time is governed by the fluid viscosity, with $s_k^\nu = {}^1/_{\tau_k}$. The other relaxation times are free parameters and can be chosen to minimize spurious currents.

The order of isotropy enters the fluid–fluid potential forces in Eq. 18. Increasing the number of neighbors used in Green's function results in lower spurious currents. To change the isotropy order, a single option is required:

```
-isotropy_order 4
```

where choices are fourth- (the default), eighth-, or tenth-order isotropy. With these changes in options, either on the command line or in an input file, identical problems are easily solved using the different algorithms.

A bubble test is used to compare the algorithms. Two components are initialized within a periodic domain and allowed to relax to a bubble, as shown in Fig. 5. Once the problem has reached steady state, the actual solution should have zero velocities, and pressure difference across the fluid interface gives the surface tension between the two fluids. Solutions comparing accuracy and performance are shown in Fig. 6. Accuracy is measured through the L_∞ norm of the spurious currents. Performance is measured as the wall-clock time to steady state on eight cores of a workstation. Steady state here is defined by when $\|f_i^k(\mathbf{x}, t + \Delta t) - f_i^k(\mathbf{x}, t)\|_\infty < 10^{-5}$. This tolerance is chosen to be more than an order of magnitude smaller than the spurious currents to ensure that the error is resolved. The time reported is the minimum of three runs to limit differences in machine load.

For this problem, Fig. 6 suggests that increasing the isotropy order is a more efficient strategy than switching to MRT, as the tenth-order SRT method is both more accurate and faster than the fourth-order MRT method. However, the most expensive methods are barely more than 50 % more expensive than the least expensive methods. We note that these conclusions can not always be generalized; experience suggests that different fluids, relaxation parameters, fluid configurations, the presences of walls, and the resolution of geometric features can all have significant effects on accuracy.

This example demonstrates the trade-offs between accuracy and performance. Six combinations of algorithms are easily tested and compared, allowing a user to make educated decisions for a specific application. This is enabled by the modular design Taxila LBM.

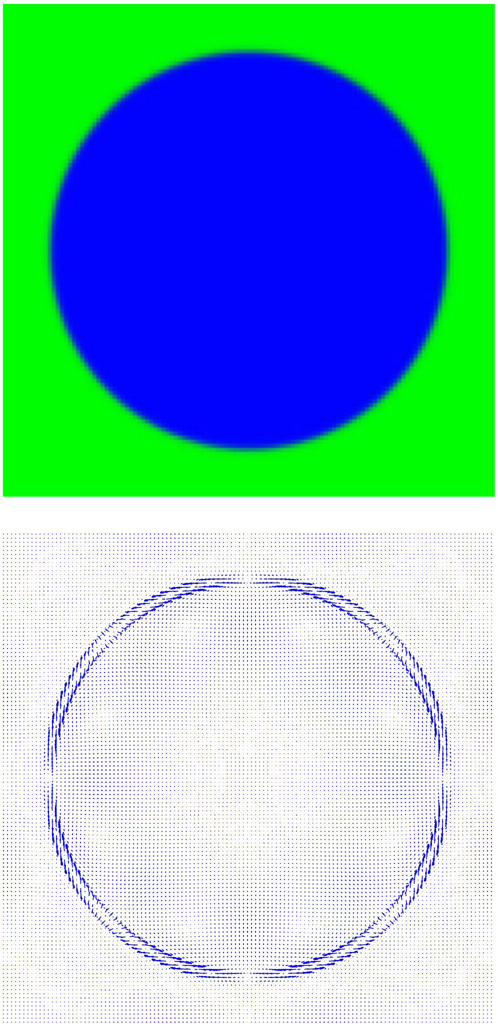

Fig. 5 Final fluid configuration of a bubble test at steady state. In this steady-state calculation, error is measured by spurious currents, plotted in the lower image. The maximum spurious current in this simulation is of magnitude 0.0044 in lattice units. Many algorithms have been derived to reduce these currents

3.2 Parallel performance

Most pore-scale simulations of realistic rock or micromodel media are, by their nature, computationally expensive. For a typical simulation of a microfluidic device experiment, domains are on the scale of thousands of pixels in each dimension. Additionally, multiphase simulations are often run for millions of timesteps. Even these basic problems can take tens of thousands of CPU hours.

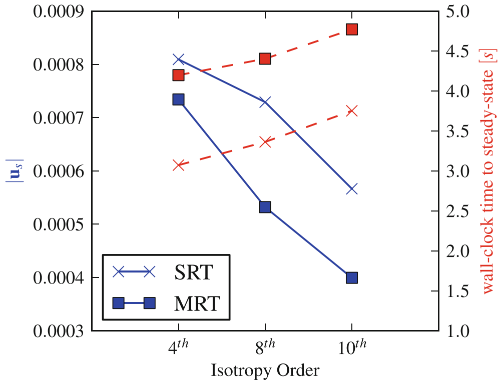

Fig. 6 Comparison of performance and accuracy of six relaxation methods on a standard two-component bubble test. Both SRT and MRT methods of three different isotropy orders are shown, with accuracy (in terms of the maximum value of spurious currents) in *solid blue* and performance (in terms of time to reach steady state) in *dashed red*

Therefore, parallel performance of lattice Boltzmann simulators is critical. Fortunately, most of the work in a LBM simulation is done in the collision operator, which is completely local in most spatial domain decomposition approaches. Therefore, strong scaling to very few lattice sites per process is expected. Figure 7 shows strong scaling on two problems of different sizes, up to one run on 2^{16} cores.

Much of this performance is due to the use of the Portable, Extensible Toolkit for Scientific computation (PETSc) [2] for data structures and communication and parallel i/o. PETSc provides, amongst many other capabilities, a set of data structures to define structured meshes with multiple degrees of freedom, decompose these data structures spatially across many processors, and communicate nearest neighbor(s) information across these processors. Building on MPI, a standard for message-based parallelism, PETSc abstracts both data structures and communication, allowing science-driven codes such as Taxila LBM to accomplish excellent parallel performance while using a high-level

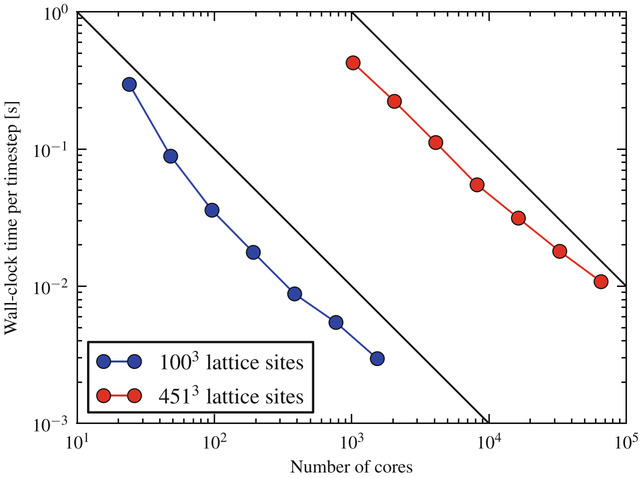

Fig. 7 Strong scaling of 100^3 and 451^3 lattice site versions of the problem in Section 4.1. The 100^3 problem is run on LANL's Lobo, while the 451^3 problem is run on ORNL's Jaguar XT-5. *Black lines* indicate ideal scaling

interface for parallelism. PETSc has been tested and benchmarked on some of the largest machines for a variety of scientific applications [2].

Distribution functions are defined on a DMDA (PETSc's structured mesh data structures) and communicated with varying stencil sizes (depending upon the isotropy order) using PETSc's communication methods for those data structures. Furthermore, interleaving of communication and calculation allows most of the communication cost to be nullified. Non-blocking communication patterns in PETSc allow density to be calculated and communication begun. While communication is occurring, fluxes and local forces are calculated. Then receipt of the messages is checked before nonlocal forces are calculated. In this way, for many applications and many machines, the effect of increased communication due to increased isotropy order is eliminated; all communication costs in the collision and forcing step are eliminated.

Additionally, PETSc's spatially distributed data structures and communication patterns used in Taxila LBM allow algorithms and simulations to be written with little knowledge on the parallel structure. Each physical algorithm receives a standard multidimensional array representing its local portion of the domain, along with a data structure specifying that process's local start and end (and the start and end of ghost values). Ghost cells containing the data from neighboring processes can be filled by a single method call for communication. This allows code that will run on hundreds of thousands of cores to be easily developed and tested on single and several-core machines.

4 Examples

To demonstrate the capabilities of Taxila LBM, we include two examples of common problems.

4.1 Permeability measurements of Berea sandstone

First, we demonstrate the problem of determining the permeability of a Berea sandstone pore geometry measured via computed x-ray tomography. The data used here is reconstructed at 4.5 μm/voxel resolution, and a 451-cubed voxel subset of the core sample is selected to simplify the domain.

To measure the permeability of the sandstone, a pressure gradient is imposed in the z-direction (along the symmetry axis of the original core sample). Boundary conditions are set to require a $0.01/3$ (nondimensional) pressure difference across the 451 lattice sites, and initial conditions are set to a linear profile in pressure and zero velocity. The domain is specified to be periodic in x and y coordinate directions to limit edge effects. With these specifications, Taxila LBM is run to 100,000 timesteps on 1,028 cores, which is at steady state.

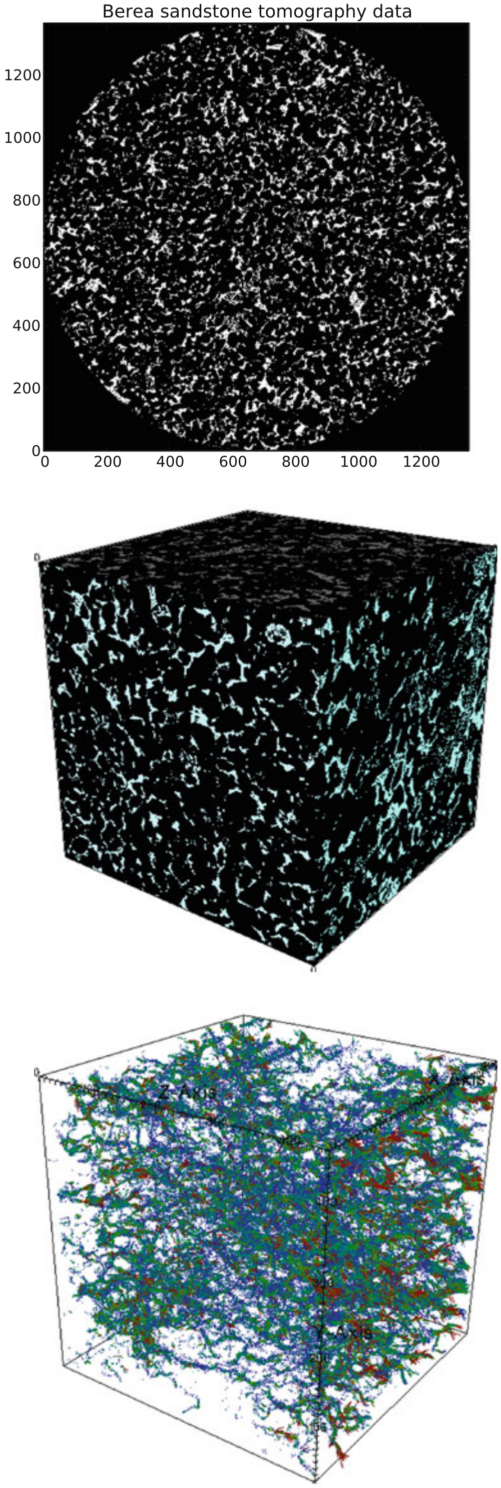

Fig. 8 *Top view* of CMT Berea core sample. Region on which the LBM is used to simulate flow and calculate permeability. Streamlines of flow in the final steady-state solution

The problem is dimensionalized using a length scale given by the voxel size and a time scale given by the viscosity of the fluid, $\nu = L^2/t$, noting that the viscosity is related to the relaxation time by $\tau = \frac{\nu-0.5}{3}$. A mass scale is given by the density of the fluid.

Results of the simulation are shown in Fig. 8, including solid configuration and streamlines of the velocity field. Porosity of the sample is 0.14. Relating the mean velocity in the z-direction to the applied pressure gradient, a permeability of 1.99×10^{-13} m^2 is calculated. This lies within the expected range for sandstones of 5×10^{-17}–9×10^{-13} [15].

4.2 Multicomponent flows in microfluidic devices

In this example, we simulate immiscible displacement in a heterogenous micromodel with a porosity of 47 %. The simulation domain is 4090 × 2855 pixels with a resolution of 3.6 μm/pixel obtained from microscopic imaging of an experimental system. The smallest pore throat has a width of three pixels, which makes these simulations especially challenging. On one hand, we would prefer higher resolution in the smallest pore throats to ensure accurate numerical results everywhere in the micromodel. On the other hand, there are approximately 12 million lattice sites in the current simulation domain, making a higher-resolution simulation computationally infeasible. In order to ensure stable numerical simulations at low resolution, we have used a tenth-order isotropy and chosen the MRT relaxation parameters such that the spurious currents are minimized.

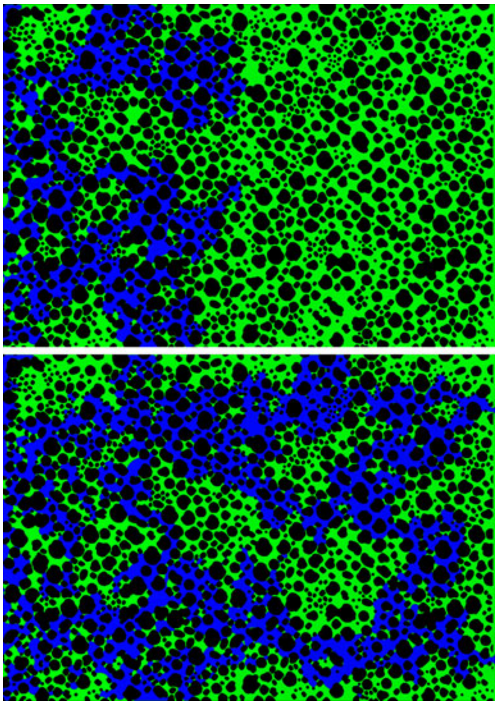

Fig. 9 Example of immiscible displacement of the wetting phase (*green*) by the non-wetting phase (*blue*) in a heterogeneous micromodel

The micromodel is initially fully saturated with the wetting phase, and we apply a flux boundary condition for the non-wetting phase at the inlet and a constant pressure boundary condition at the outlet. The contact angle is 65°, which matches the experimental system. Figure 9 shows the complex phase distributions within the pores at two different times.

5 Conclusions

In this paper, we have introduced the Taxila LBM, a framework for lattice Boltzmann methods designed for geologic porous media. The code solves both single and multicomponent systems on regular lattices in both two and three dimensions. It is based upon the Shan and Chen LBM, but includes many advances including higher-order isotropy in the fluid–fluid interfacial terms and multiple relaxation methods. Taxila LBM uses the forcing formulation in [11], which enables robust simulation in cases of high viscosity ratios and allows custom equations of state for high-density ratios. Multimineralic wall materials are enabled, allowing mixed wetting systems with independent contact angles. Arbitrary, heterogeneous boundary conditions and custom initial conditions are easily included, making it easily applied to most systems.

Additionally, Taxila LBM has been demonstrated to scale well on 60,000+ cores, and identical code is run on both laptops and supercomputers. This enables users to test small problems on laptops and workstations, confident that the larger problem will run on larger machines. Taxila LBM's framework for algorithmic advances enables developers to quickly and easily compare and test new algorithms while quickly letting users leverage these advancements. We have demonstrated here how Taxila LBM enables these use models and demonstrated the performance of Taxila LBM on two common cases: the characterization of permeability on computed x-ray tomography data and the simulation of multicomponent flows in microfluidic devices.

Taxila LBM is available freely as open-source software at http://software.lanl.gov/taxila/.

Acknowledgments The development of Taxila LBM has been supported by LANL LD-RD 20100025DR (Bill Carey, P.I.). Support for this work of both technical and academic nature was provided by J. David Moulton. Thanks go especially to Dorthe Wildenschild for the CMT data in Section 4.1.

References

1. Aidun, C.K., Clausen, J.R.: Lattice-Boltzmann model for complex flows. Annu. Rev. Fluid Mech. **42**, 439–472 (2010)
2. Balay, S., Brown, J., Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., Zang, H.: PETSc users manual, Technical report ANL-95/11. Revision 3.3. Argonne National Laboratory, Lemont (2012)
3. Chang, C., Liu, C.-H., Lin, C.-A.: Boundary conditions for lattice Boltzmann simulations with complex geometry flows. Comput. Math. Appl. **58**, 940–949 (2009)
4. Chen, S., Doolen, G.W.: Lattice-Boltzmann method for fluid flows. Annual Rev. Fluid Mech. **30**, 329–364 (1998)
5. He, X., Zou, Q., Luo, L.-S., Dembo, M.: Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice Boltzmann BGK model. J. Stat. Phys. **87**, 115–136 (1997)
6. He, X., Chen, S., Doolen, G.D.: A novel thermal model for the lattice Boltzmann method in incompressible limit. J. Comp. Phys. **146**, 282–300 (1998a)
7. He, X., Shan, X., Doolen, G.D.: Discrete Boltzmann equation model for nonideal gases. Phys. Rev. E **57**(1), R14 (1998b)
8. He, X., Chen, S., Zhang, R.: A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh–Taylor instability. J. Comp. Phys. **152**(2), 642–663 (1999)
9. Lallemand, P., Luo, L.-S.: Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, Gallilean invariance, and stability. Phys. Rev. E **61**(6), 6546–6562 (2000)
10. McCracken, M.E., Abraham, J.: Multiple-relaxation-time lattice-Boltzmann model for multiphase flow. Phys. Rev. E **71**, 036701 (2005)
11. Porter, M.L., Coon, E.T., Kang, Q., Moulton, J.D., Carey, J.W.: Multicomponent interparticle-potential lattice Boltzmann model for fluids with large viscosity ratios. Phys. Rev. E **86**, 036701 (2012)
12. Premnath, K.N., Abraham, J.: Three-dimensional multi-relaxation time (MRT) lattice-Boltzmann models for multiphase flow. J. Comput. Phys. **224**, 539–559 (2007)
13. Qian, Y., D'Humières, D., Lallemand, P.: Lattice BGK models for Navier-Stokes equation. Europhys. Lett. **17**, 479–484 (1992)
14. Sbragaglia, M., Benzi, R., Biferale, L., Succi, S., Sugiyama, K., Toschi, F.: Generalized lattice-Boltzmann method for multirange pseudopotential. Phys. Rev. E. Stat. Nonlin. Soft. Matter. Phys. **75**(2 Pt 2), 026702 (2007)
15. Schwartz, F.W., Zhang, H.: Fundamentals of Ground Water. Wiley, New York (2002)
16. Shan, X., Chen, H.: Lattice-Boltzmann model for simulating flows with multiphases and components. Phys. Rev. E **47**, 1815–1819 (1993)
17. Shan, X., Chen, H.: Simulations of non-ideal gases and liquid-gas phase transitions by the lattice-Boltzmann equation. Phys. Rev. E **49**, 2941–2948 (1994)
18. Shan, X.: Analysis and reduction of the spurious current in a class of multiphase lattice Boltzmann models. Phys. Rev. E **73**(4), 047701 (2006)
19. Shan, X., Doolen, G.: Diffusion in a multicomponent lattice Boltzmann equation model. Phys. Rev. E **54**(4), 3614–3620 (1996)
20. Swift, M.R., Orlandini, E., Osborn, W.R., Yeomans, J.M.: Lattice Boltzmann simulation of liquid-gas and binary fluid systems. Phys. Rev. E **54**(5), 5041–5052 (1996)
21. Yuan, P., Schaefer, L.: Equations of state in a lattice Boltzmann model. Phys. Fluids **18**, 042101 (2006)
22. Zhang, J.: Lattice Boltzmann method for microfluidics: models and applications. Microfluid Nanofluid **10**, 1–28 (2011)