



A New Lattice Boltzmann Equation to Simulate Density-Driven Convection of Carbon Dioxide

by

**Rebecca Allen
Tim Reis
Shuyu Sun**



SPE-163658-MS

A New Lattice Boltzmann Equation to Simulate Density-Driven Convection of Carbon Dioxide

Rebecca Allen, SPE, King Abdullah University of Science and Technology (KAUST)

Tim Reis, Oxford Centre for Collaborative Applied Mathematics (OCCAM)

Shuyu Sun, SPE, KAUST

Copyright 2013, Society of Petroleum Engineers

This paper was prepared for presentation at the SPE Reservoir Simulation Symposium held in The Woodlands, Texas USA, 18–20 February 2013.

This paper was selected for presentation by an SPE program committee following review of information contained in an abstract submitted by the author(s). Contents of the paper have not been reviewed by the Society of Petroleum Engineers and are subject to correction by the author(s). The material does not necessarily reflect any position of the Society of Petroleum Engineers, its officers, or members. Electronic reproduction, distribution, or storage of any part of this paper without the written consent of the Society of Petroleum Engineers is prohibited. Permission to reproduce in print is restricted to an abstract of not more than 300 words; illustrations may not be copied. The abstract must contain conspicuous acknowledgment of SPE copyright.

Abstract

The storage of CO₂ in fluid-filled geological formations has been carried out for more than a decade in locations around the world. After CO₂ has been injected into the aquifer and has moved laterally under the aquifer's cap-rock, density-driven convection becomes an important transport process to model. However, the challenge lies in simulating this transport process accurately with high spatial resolution and low CPU cost. This issue can be addressed by using the lattice Boltzmann equation (LBE) to formulate a model for a similar scenario when a solute diffuses into a fluid and density differences lead to convective mixing. The LBE is a promising alternative to the traditional methods of computational fluid dynamics. Rather than discretizing the system of partial differential equations of classical continuum mechanics directly, the LBE is derived from a velocity-space truncation of the Boltzmann equation of classical kinetic theory.

We propose an extension to the LBE, which can accurately predict the transport of dissolved CO₂ in water, as a step towards fluid-filled porous media simulations. This is achieved by coupling two LBEs, one for the fluid flow and one for the convection and diffusion of CO₂. Unlike existing lattice Boltzmann equations for porous media flow, our model is derived from a system of moment equations and a Crank-Nicolson discretization of the velocity-truncated Boltzmann equation. The forcing terms are updated locally without the need for additional central difference approximation. Therefore our model preserves all the computational advantages of the single-phase lattice Boltzmann equation and is formally second-order accurate in both space and time. Our new model also features a novel implementation of boundary conditions, which is simple to implement and does not suffer from the grid-dependent error that is present in the standard "bounce-back" condition.

The significance of using the LBE in this work lies in the ability to efficiently simulate density-driven convection of CO₂ through water. From an implementation viewpoint, the locality of our algorithm exploits massively parallel modern computer architectures, including graphics processing units (GPUs), which would lead to very fast computations that scale linearly with the number of processors.

Introduction

Density-driven convection is a process that is observed in real-world problems, especially during CO₂ storage in saline aquifers. Since CO₂ storage in geological formations has been proposed as an effective mitigation strategy for reducing atmospheric CO₂ levels [1], it is important to determine what parameters impact density-driven convection and the ultimate rate of CO₂ dissolution. Past work (i.e. [2-6]) has covered this topic, and concludes the convective mixing that takes place during the dissolution process of CO₂ storage increases the rate of CO₂ dissolution into the aquifer. The implication is that an increase in the CO₂ dissolution rate leads to an enhanced storage capacity of the aquifer.

In research, the traditional approaches for modeling CO₂ dissolution and density-driven convection relied on finite volume methods (or the use of commercial simulations built upon finite volume methods) for numerical simulation [7]. Some work has been done on the use of lattice Boltzmann modeling for CO₂ storage applications (i.e. [8]). Indeed, the use of the lattice Boltzmann equation (LBE) for porous media has received attention (i.e. [9]), and provides more research opportunity for the future.

The motivation for this work is to simulate density-driven convection of CO₂ in water, and the novelty of our approach is as follows: we use a LBE that is second order accurate in time and space, we apply moment-based boundary conditions that exactly satisfy the macroscopic boundary conditions, our collision operator is a multiple relaxation time scheme which stabilizes the flow and allows for high Rayleigh numbers on coarse meshes, and the multiple relaxation time for the transport equation is constructed to minimize numerical diffusion [10, 11]. By modeling the convective mixing of CO₂ in water in a non-porous domain, results from this work set a solid foundation to build when considering a porous domain.

To present this work, we begin by discussing the discrete Boltzmann equations used for hydrodynamics and concentration modeling. An overview is given of how the discrete Boltzmann equations satisfy the macroscopic equations, which include the Navier-Stokes equations and the convective-diffusive transport equation. Then the discrete Boltzmann equations are reduced to the lattice Boltzmann equations. After the mathematical model section, an overview of the multiple relaxation time scheme for the collision operator is given, followed by a discussion of the moment-based boundary conditions and the specific boundary conditions used in this problem. Results are presented and findings are summarized in the conclusion section.

Mathematical Model

Discrete Boltzmann Equations

In this problem, the change in fluid density as well as solute density are solved for over time, thus two discrete Boltzmann equations (DBEs) are used: the first DBE is for the hydrodynamics and the second DBE is for concentration. They are

Equ. 1

$$\begin{aligned}\frac{\partial f_k}{\partial t} + \underline{c}_k \cdot \nabla f_k &= \Omega_{kj}(f_j - f_j^{eq}) + R_k \\ \frac{\partial g_k}{\partial t} + \underline{c}_k \cdot \nabla g_k &= \Omega_{kj}(g_j - g_j^{eq})\end{aligned}$$

where the discrete distribution functions, $f_k(\underline{x}, t)$ and $g_k(\underline{x}, t)$, are the fluid and solute particle distribution functions in the k -th direction, respectively, f^{eq} and g^{eq} are the equilibrium distribution functions, \underline{c}_k is the discrete particle velocities, Ω_{kj} is the collision operator term (which is $-(1/\tau)\delta_{kj}$ if the BGK collision operator is used, where δ_{kj} is the Kronecker delta function), R_k is the forcing term, and t is time. The inclusion of j in the DBEs implies Einstein's summation convention. To get the lattice Boltzmann equations, we need to fully discretize the DBEs (i.e. in space and time).

The equilibrium distribution functions are constructed in such a way that the conservation of mass and linear momentum conditions are met. In this work, f_k is computed on a D2Q9 model grid (two-dimensional, with nine discrete particle velocities), thus f^{eq} is:

Equ. 2

$$f_k^{eq}(\underline{x}, t) = w_k^f \rho(\underline{x}, t) \left[1 + 3(\underline{u}(\underline{x}, t) \cdot \underline{c}_k) + \frac{9}{2}(\underline{u}(\underline{x}, t) \cdot \underline{c}_k)^2 - \frac{3}{2}\underline{u}(\underline{x}, t)^2 \right]$$

where $w_k^f=4/9$ for $k=0$, $w_k^f=1/9$ for $k=1-4$, $w_k^f=1/36$ for $k=5-8$. For g_k , a D2Q5 model grid is used (two-dimensional, with five discrete particle velocities), thus g^{eq} is:

Equ. 3

$$g_k^{eq}(\underline{x}, t) = w_k^g C(\underline{x}, t) [1 + 3(\underline{u}(\underline{x}, t) \cdot \underline{c}_k)]$$

where $w_k^g=1/3$ for $k=0$, and $w_k^g=1/6$ for $k=1-4$. Figure 1 shows the D2Q9 grid, with the discrete particle velocities for a lattice node located at (i, j) . Nine vectors exist in the D2Q9 grid, including a rest particle centered on the node. In the D2Q5 grid, only the discrete velocities labeled 0, 1, 2, 3, and 4 exist.

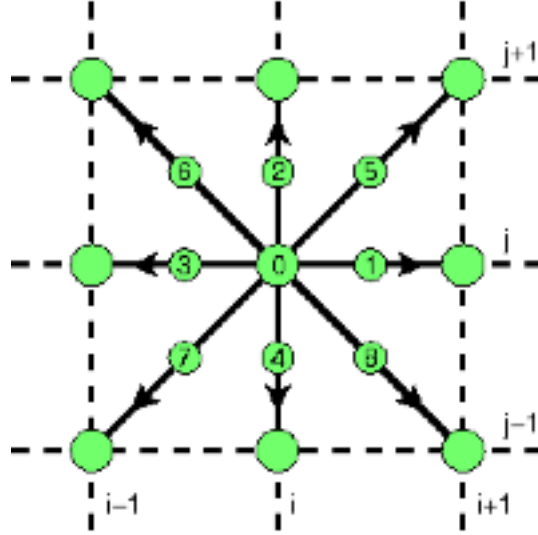


Figure 1: D2Q9 lattice model grid.

Regarding the forcing term R_k that appears in the first DBE, its form must satisfy the Navier-Stokes equations and it must satisfy the following constraints:

Equ. 4

$$\sum_k R_k = 0; \quad \sum_k R_k \underline{c}_k = \underline{F}; \quad \sum_k R_k \underline{c}_k \underline{c}_k = \underline{F} \underline{u} + \underline{u} \underline{F}$$

A possible form for R is shown in Equ. 5 [12], although this work incorporates the force directly onto the moments, so R is not solved for explicitly in the algorithm expect during initialization.

Equ. 5

$$R_k(x, t) = \omega_k \left(\frac{\underline{c}_k - \underline{u}(x, t)}{c_s^2} + \frac{(\underline{c}_k \cdot \underline{u}(x, t)) \underline{c}_k}{c_s^4} \right) \cdot \underline{F}$$

The fluid density, fluid momentum, and fluid momentum flux, (as well as higher order moments, Q), solute density, and solute flux are computed respectively by:

Equ. 6

$$\rho = \sum_k f_k, \quad \rho \underline{u} = \sum_k f_k \underline{c}_k, \quad \Pi = \sum_k f_k \underline{c}_k \underline{c}_k, \quad Q = \sum_k f_k \underline{c}_k \underline{c}_k \underline{c}_k, \quad C = \sum_k g_k, \quad \psi = \sum_k g_k \underline{c}_k$$

Then, by taking the moments of the first DBE in Equ. 1 (for f) we find the following set of equations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \underline{u}) = 0$$

$$\frac{\partial (\rho \underline{u})}{\partial t} + \nabla \cdot \Pi = 0$$

$$\frac{\partial \Pi}{\partial t} + \nabla \cdot Q = \Omega(\Pi - \Pi^{(0)})$$

Equ. 7

Since Π and Q are non-conserved moments, they are expanded using Chapman-Enskog expansion in order to seek solutions over slowly varying timescales that are much longer than the relaxation parameter, τ_f . These expansions are

Equ. 8

$$\Pi = \Pi^{(0)} + \tau_f \Pi^{(1)} + \mathcal{O}(\tau_f^2) \quad Q = Q^{(0)} + \tau_f Q^{(1)} + \mathcal{O}(\tau_f^2)$$

where the relaxation time for the fluid is $\tau_f = \nu/c_s^2 = 3\nu$. To preserve the ordering we must also expand the temporal derivatives, for example:

Equ. 9

$$\partial_t = \partial_{t_0} + \tau \partial_{t_1} + \mathcal{O}(\tau^2)$$

By taking a leading-order approximation of the equation listed in the above set, and a few other steps, the terms in the momentum flux expansion are

Equ. 10

$$\Pi^{(0)} = \frac{1}{3} \rho I + \rho \underline{u} \underline{u}, \quad \Pi^{(1)} = -\frac{1}{3} \rho (\nabla \underline{u} + (\nabla \underline{u})^T) + O(Ma^3)$$

Then, the Navier-Stokes equations are obtained by substituting the expansion terms for Π into the set of equations.

$$\begin{aligned} \frac{\partial(\rho \underline{u})}{\partial t} + \underline{u} \cdot \nabla(\rho \underline{u}) &= -\nabla p + \rho F + \mu \nabla^2 \underline{u} \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \underline{u}) &= 0 \end{aligned}$$

Note that $\Pi^{(1)}$ with the coefficient $-\rho \tau_f/3$ gives the Newtonian viscous stress tensor (plus an $O(Ma^3)$ error term) where the kinematic viscosity, ν , is $\tau_f/3$. In this work, the force term in the linear momentum balance accounts for a buoyancy term that depends on the concentration of CO_2 in the fluid, and is written as $F = \alpha g(C - C_c)$, where C_c is a reference concentration, α is the volumetric expansion due to solute concentration, and g is gravitational acceleration. As CO_2 dissolves into water, the resulting fluid density increases linearly with CO_2 concentration [13]. As the density of the fluid increases, convective fingering will be initiated once the interface between the two fluids becomes disturbed. Then, as these convective fingers penetrate into the depth of the domain, the fluid is susceptible to the Rayleigh-Taylor instability. Also in this work, we assume the fluid is incompressible, and thus the change in density over time is zero, as well as the spatial density gradient. We further make the assumption that the density does not vary greatly, and thus the Boussinesq approximation implies that density is important in the buoyancy term only, as noted by the transformation of the mass balance equation into the continuity equation.

The multiple scales expansion of Equ. 1 for g yields

$$\begin{aligned} \frac{\partial C}{\partial t} + \nabla \cdot \psi &= \Omega(C - C^{(0)}) \\ \frac{\partial \psi}{\partial t} + \nabla \cdot \Gamma &= \Omega(\psi - \psi^{(0)}) \end{aligned}$$

Equ. 11

Expanding the non-conserved moments ψ and Γ , where the relaxation parameter for the solute distribution function is $\tau_g = d/c_s^2 = 3d$, we have

$$\psi = \psi^{(0)} + \tau_g \psi^{(1)} + \mathcal{O}(\tau_g^2) \quad \Gamma = \Gamma^{(0)} + \tau_g \Gamma^{(1)} + \mathcal{O}(\tau_g^2)$$

Equ. 12

and by taking leading order approximation we find

Equ. 13

$$\psi = (1/3) \nabla C - \underline{u}(\underline{u} \cdot \nabla) C + O(\tau^3)$$

As shown in Equ. 13, the BGK equation for the solute field gives the desired equations with an additional numerical diffusion term. However, this numerical diffusion term can be controlled with a multiple relaxation time scheme for the collision operator, which will be discussed shortly.

Non-Dimensionalization

These modeling equations are made to be non-dimensional in order to make the model more general and applicable to the same physical problem but for a variety of different fluid parameters. First, to simplify the problem, the density is set to be unity. The dimensional variables (u , p , C , x , t) are related to the non-dimensional variables (u' , p' , C' , x' , t') through characteristic or reference values (x_0 , t_0 , g_0 , C_0). These relationships are:

$$t = t_0 t', \quad \underline{x} = x_0 \underline{x}', \quad \underline{u} = \sqrt{\alpha g_0 \Delta C x_0} \underline{u}', \quad p = \frac{x_0^2}{t_0^2} p', \quad C = (\Delta C) C' + C_c$$

Also, we can define non-dimensional numbers to use during the dimensional analysis. These are the Rayleigh and Schmidt numbers:

Equ. 14

$$Sc = \frac{\mu}{\rho d} = \frac{\nu}{d}; \quad Ra = \frac{\alpha \Delta C g_0 x_0^3}{\nu d}$$

After substituting the above relationships into the Navier-Stokes and the convection-diffusion equations, the non-dimensional modeling equations are (while neglecting density variation due to the small Mach number assumption):

Equ. 15

$$\frac{\partial \underline{u}}{\partial t} + \underline{u} \cdot \nabla \underline{u} = -\nabla p + \sqrt{\frac{Sc}{Ra}} \nabla^2 \underline{u} + C$$

Equ. 16

$$\nabla \cdot \underline{u} = 0$$

Equ. 17

$$\frac{\partial C}{\partial t} + \nabla \cdot (\underline{u} C) = \sqrt{\frac{1}{Ra Sc}} \nabla^2 C$$

Discretization of the Discrete Boltzmann Equations

Now that the macroscopic equations have been presented and we have discussed how the discrete Boltzmann equation can satisfy the Navier-Stokes equations, we will present the approach taken in this work to reduce the DBE to a fully discrete form. This discrete form gives us our lattice Boltzmann equations. However, before we discretize the DBEs, we will say

Equ. 18

$$\Omega_{kj} f_k^{(eq)} + R_k = \Omega_{kj} f_k^{(0)}$$

so that the DBE for f_k becomes

Equ. 19

$$\frac{\partial f_k}{\partial t} + \underline{c}_k \cdot \nabla f_k = \Omega_{kj} (f_j - f_j^{(0)})$$

The following discretization steps will be shown only for f_k , although the same applies for g_k . Integrating the DBE along a characteristic time gives:

Equ. 20

$$f_k(\underline{x} + \underline{c}_k \Delta t, t + \Delta t) - f_k(\underline{x}, t) = \int_0^{\Delta t} \Omega_k(\underline{x} + \underline{c}_k s, t + s) ds$$

Then, the integral can be approximated using the trapezium rule, and becomes:

Equ. 21

$$f_k(\underline{x} + \underline{c}_k \Delta t, t + \Delta t) - f_k(\underline{x}, t) = \frac{\Delta t}{2} (\Omega_k(\underline{x} + \underline{c}_k \Delta t, t + \Delta t) + \Omega_k(\underline{x}, t)) + O(\Delta t^3)$$

The above is a second order discretization of the DBE shown in Equ. 19, but yields an implicit system of algebraic equations. However, we may obtain an explicit system at each time step by a variable transformation. New variables are defined as the distribution function (f or g) plus half a time-step's worth of collisions [14]:

Equ. 22

$$\bar{f}_k(\underline{x}, t) = f_k(\underline{x}, t) + \frac{\Delta t}{2} \Omega_{kj} (f_j(\underline{x}, t) - f_j^{(0)}(\underline{x}, t))$$

and after substituting the new variables into Equ. 21, and resulting explicit form is (for the BGK collision operator):

Equ. 23

$$\bar{f}_k(\underline{x} + \underline{c}_k \Delta t, t + \Delta t) = \bar{f}_k(\underline{x}, t) - \frac{\Delta t}{\tau_f + \Delta t/2} (\bar{f}_k(\underline{x}, t) - f_k^{(0)}(\underline{x}, t))$$

where x is a spatial location, $c = \Delta x / \Delta t$ (the distance the particle travels over time at the discrete particle velocity), and t is time. The more general equation where the collision operator is not necessarily the BGK is [15]:

Equ. 24

$$\bar{f}_k(\underline{x} + \underline{c}_k \Delta t, t + \Delta t) = \bar{f}_k(\underline{x}, t) - \left[\Delta t \Omega \left(1 + \frac{\Delta t}{2} \Omega \right)^{-1} \right]_{kj} (\bar{f}_k(\underline{x}, t) - f_k^{(0)}(\underline{x}, t))$$

From the explicit forms of the LBEs, \bar{f} and \bar{g} are computed, not the regular f or g that was used to compute the macroscopic variables explained previously. However, we can use \bar{f} and \bar{g} to get to the macroscopic variables. For the fluid density,

Equ. 25

$$\bar{\rho} = \sum_k \bar{f}_k = \sum_k f_k + \frac{\Delta t}{2\tau} \sum_k (f_k - f_k^{eq}) - \frac{\Delta t}{2} \sum_k R_k = \rho$$

For the fluid velocity, we find:

Equ. 26

$$\bar{\rho} \bar{u} = \sum_k \bar{f}_k c_k = \sum_k f_k c_k + \frac{\Delta t}{2\tau} \sum_k (f_k c_k - f_k^{eq} c_k) - \frac{\Delta t}{2} \sum_k R_k c_k = \sum_k f_k c_k - \frac{\Delta t}{2} \sum_k R_k c_k = \rho u - \frac{\Delta t}{2} F$$

Also since $\rho = \bar{\rho}$ and $F = \rho \underline{g}$, the velocity is computed by:

Equ. 27

$$\underline{u} = \bar{u} + \frac{\Delta t}{2} \underline{g}$$

For the solute density,

Equ. 28

$$\bar{C} = \sum_k \bar{g}_k = \sum_k g_k + \frac{\Delta t}{2\tau} \sum_k (g_k - g_k^{eq}) = C$$

Multiple Relaxation Time Collision Operator

The idea behind the multiple relaxation time (MRT) scheme is to use different rates to relax each moment, and to perform collisions on the moments, rather than on the distribution functions that eventually calculate these moments. Being able to relax moments at different rates gives us control over which moments relax directly to zero after collisions. The reason why it is beneficial to relax certain moments directly to zero is this can dampen the oscillations that non-conserved moments might cause at high Reynolds numbers (i.e. when τ is small).

In this work, we use the MRT scheme presented in [15], with the relaxation times of the kinetic moments set to $\Delta t/2$ in order to make them decay straight to zero after collisions. The collision operator in the MRT DBE is made up of a diagonal matrix Λ

that contains the relaxation times. M is a transformation matrix that maps between the distribution function and its moments, by $m = Mf$, or $f = M^{-1}m$. Thus, the DBE can be written as the following:

Equ. 29

$$\frac{\partial f_k}{\partial t} + c_k \cdot \nabla f_k = M^{-1} \Lambda(m_k - m_k^{(eq)})$$

In the D2Q9 lattice grid used for f , the mapping between nine equilibrium moments and the nine equilibrium distribution functions for the fluid give the construction of M . The first six equilibrium moments correspond to the hydrodynamic moments (density and the momentum in the x and y direction, the three more make up the momentum flux tensor Π), and the final three correspond to the ‘ghost’ moments. These three ghost moments are found from Σh_k , $\Sigma h_k c_k^x$, and $\Sigma h_k c_k^y$, where $h_k = (1, -2, -2, -2, -2, 4, 4, 4, 4)^T$ [15], corresponding to the next three Hermite polynomials. The collision times for the first six moments are set to be τ in order to get the viscosity to be $\tau/3$, and the collision times for the final three moments are set to be $\tau = \Delta t/2$. By doing so, the ghost moments decay straight to zero after collisions.

In the D2Q5 lattice grid used for g , the mapping between five moments and the five distribution functions for the solute give a construction for M that is made up of the following lattice vectors, which are found by Gramm-Schmidt orthogonalization:

$$\{1\}, \quad c_k^x, \quad c_k^y, \quad 5(c_x^2 + c_y^2) - 4, \quad c_x^2 - c_y^2$$

We construct the collision operator for g to avoid a checkerboard instability and to control the numerical diffusion, while making sure that it has fourth order isotropy [10, 16, 17]. The equilibrium moments of g are

$$C, \quad u^x C, \quad u^y C, \quad aC, \quad 0$$

where a is chosen according to $d = \tau_k(4+a)/10$, where $\tau_k = (\sqrt{3})/6$, and the other two relaxation times are $(\sqrt{3})/3$ [10]. The MRT collision operator for g allows us to reduce the numerical error and make the numerical diffusion isotropic to fourth order, while the BGK collision operator cannot [11].

Boundary Conditions

A simple and traditionally-applied method to implement boundary conditions in lattice Boltzmann modeling is to set the outward pointing f equal to the inward pointing f at the lattice node located on or near the wall. This method is known as the ‘bounce-back’ method, however when bounce-back is used with the BGK collision operator, the no-slip boundary condition is not satisfied. A method that does not require constraints to be placed on the collision matrix in order to satisfy boundary conditions is the moment-based method [18, 19]. Figure 2 illustrates that the moment-based method simulates a velocity profile that is exact with machine precision to the analytical solution in the Poiseuille flow problem (i.e. flow through a long narrow channel). The figure also shows that the bounce-back method simulates a non-zero velocity along the wall, which is due to numerical error and not due to physical slip [20], and thus the velocity profile does not match the analytical solution.

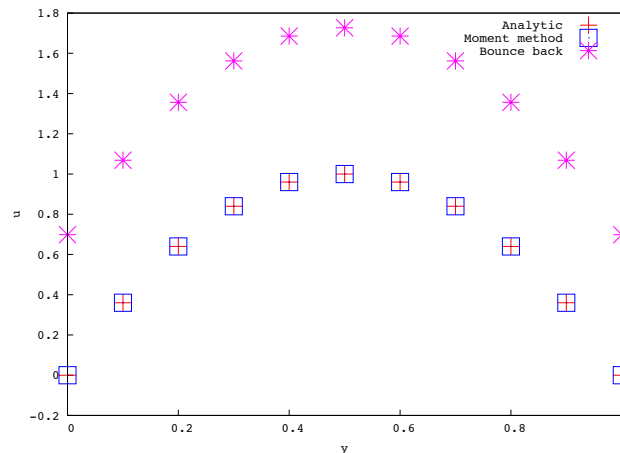


Figure 2: Comparison of velocity profiles (simulated using two different boundary methods) against the analytical solution for Poiseuille Flow problem. Run with Reynolds number = 1, and width resolution = 11 lattice nodes (walls located at $y=0$ and $y=1$). Moment method gives an exact solution, while bounce-back simulates non-physical slip on walls.

A comparison of the bounce-back method for different Reynolds numbers is presented in Figure 3. When $Re=100$, the agreement with the analytic solution is good, however not exact. When $Re=1$, the simulated velocity profile in the channel is poor. The error in the bounce-back method is a function of viscosity, which means bounce-back will predict unreliable permeabilities in flows in porous media [9, 21, 22]. Since flows in porous media often have a very low Reynolds number, bounce-back boundary conditions are not accurate enough to implement for porous media simulations.

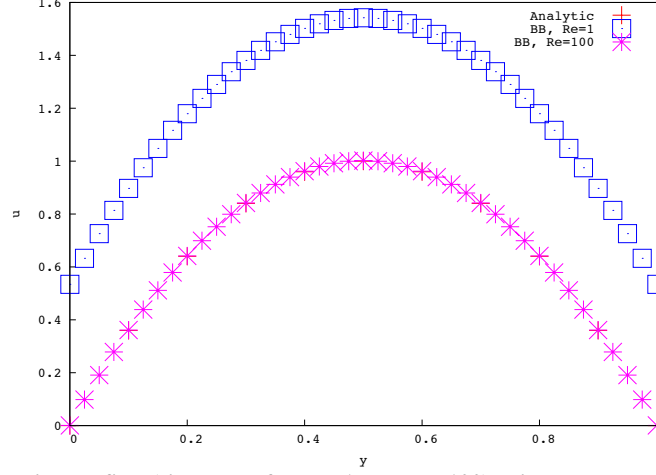


Figure 3: Comparison of velocity profiles (simulated for $Re=1$ and $Re=100$) using bounce-back boundary conditions for Poiseuille Flow problem. Width resolution = 41 lattice nodes (walls located at $y=0$ and $y=1$).

Since the moment-based method satisfies the no-slip boundary condition while the bounce-back method does not, we choose to implement moment-based boundary conditions in this work. In moment-based boundary conditions, conditions are imposed on the *moments* of f and g , rather than on f and g directly. Then translation from the conditions in the moment basis to conditions in the particle basis takes place. For example, when using the D2Q9 model, a south wall has three unknown distribution functions (f_2, f_5, f_6), so three linearly independent equations are needed. These equations come from the moments, and the moments at the wall arrange themselves into linearly independent groups. The table below presents these linearly independent moment groupings for the south wall.

Table 1: Moment groups for a south wall

<i>Moments</i>	<i>Combination of unknowns in moment equation</i>
$\rho, \rho u^y, \Pi^{yy}$	$f_2 + f_5 + f_6$
$\rho u^x, \Pi^{xy}, Q^{xyy}$	$f_5 - f_6$
$\Pi^{xx}, Q^{xxy}, R^{xxyy}$	$f_5 + f_6$

Note that a similar table can be constructed for any of the other walls, with the appropriate moment groups that provide a set of three linearly independent equations. One moment from each group is selected, and then a constraint is imposed on each selected moment (i.e. the boundary conditions of the wall), and then the unknowns are solved. For example, if a horizontal wall is impermeable to fluid flow, then $\rho u^y=0$, or if the horizontal wall has ‘ q ’ fluid flux across it, then $\rho u^y=q$. If there is no-slip along the horizontal wall, then $\rho u^x=0$. The same principal applies for vertical walls, except with the appropriate velocity components to describe no-slip and fluid flow. These flux and no-slip boundary conditions give two moments, however one more is needed to solve for the three unknowns. For the horizontal wall, Π_{xx} provides another independent moment equation, because it is the only other independent hydrodynamic moment. When there is no-slip on the wall, then $\delta u/\delta x = 0$, so then $\Pi_{xx} = \Pi_{xx}^{(0)}$.

For example, at a south wall where we need to solve for f_2, f_5, f_6 , the equations to use are the following:

$$\begin{aligned}
\rho \underline{u}^x &= 0 = \sum_k f_k c_k^x = f_1 - f_3 + f_5 - f_6 - f_7 + f_8 \\
\rho \underline{u}^y &= 0 = \sum_k f_k c_k^y = f_2 - f_4 + f_5 + f_6 - f_7 - f_8 \\
\Pi^{xx} &= \frac{\rho}{3} + \rho \underline{u}^x \underline{u}^x = \sum_k f_k c_k^x c_k^x = f_1 + f_3 + f_5 + f_6 + f_7 + f_8
\end{aligned}
\tag{Equ. 30}$$

When using the \bar{f} scheme and with a forcing term present, the form of these moment equations change slightly, since we have \bar{f} but we want to find u (not \bar{u}). The way to determine u from \bar{u} was shown above, so now the moment equations that will be used for a horizontal wall for the \bar{f} scheme are:

$$\begin{aligned}
\rho \bar{u}^x &= \rho \underline{u}^x = \sum_k \bar{f}_k c_k^x \\
\rho \bar{u}^y &= \rho \underline{u}^y - \frac{\Delta t}{2} \rho g = \sum_k \bar{f}_k c_k^y \\
\bar{\Pi}^{xx} &= \Pi^{xx} + \frac{\Delta t}{2} \Omega (\Pi^{xx} - \Pi^{(0),xx}) = \Pi^{xx} = \frac{\rho}{3} + \rho \underline{u}^x \underline{u}^x = \sum_k \bar{f}_k c_k^x c_k^x
\end{aligned}
\tag{Equ. 31}$$

For the solute boundary conditions, the D2Q5 lattice grid is used, and thus a wall only has one unknown g that needs to be solved. The solute flux is given by the expansion shown in Equ. 8, and thus the vertical solute flux across a horizontal wall is

$$\psi_y = \psi_y^{(0)} + \tau_g \psi_y^{(1)} = \sum_k g_k c_k^y \tag{Equ. 32}$$

Since we prescribe $u=0$ on the wall, by Chapman-Enskog we find

$$\psi_y^{(0)} = 0, \quad \psi_y^{(1)} = \frac{1}{3} \frac{\partial C}{\partial y} \tag{Equ. 33}$$

Thus, when $u_y=0$, the solute flux across the horizontal wall is

$$\psi_y = -d \frac{\partial C}{\partial y} = \sum_k g_k c_k^y = g_2 - g_4 \tag{Equ. 34}$$

where $d=\tau_g/3$. So, by prescribing the flux on the boundary, the unknown g coming off the wall can be solved. However, in this work, we use the LBE shown in Equ. 23 (but for solute), which came from defining a new variable. The non-equilibrium part of the expansion for the flux is thus

$$\psi^{(1)} = \left(\frac{2\tau_g}{2\tau_g + 1} \right) (\bar{\psi} - \psi^{(0)}) \tag{Equ. 35}$$

and after some substituting, we find that for a wall where $u=0$, the unknown g coming off the wall with a no-solute flux condition may be solved from rearranging

$$\psi_y = \left(\frac{2\tau_g}{2\tau_g + 1} \right) (\bar{g}_2 - \bar{g}_4) \tag{Equ. 36}$$

If the boundary condition is such that a horizontal wall has a constant solute concentration, the moment-based boundary condition is simply a rearrangement of Equ. 28. For example, if a north wall has a constant solute concentration of C , the unknown g_4 is found by

Equ. 37

$$\bar{g}_4 = C - \bar{g}_0 - \bar{g}_1 - \bar{g}_2 - \bar{g}_3$$

In this problem, the boundary conditions are presented in Figure 4, and are the following: there is no fluid flow across the top and bottom boundaries (i.e. the north and south boundaries are impermeable cap-rock), there is no solute flux across the bottom boundary, however the top boundary has a constant solute concentration of $C = 1$ (i.e. supplied by a layer of fluid that is fully saturated with CO_2). The side boundaries are periodic, in order to represent an infinitely wide domain. Initially, the domain contains no solute concentration, and thus $C = 0$.

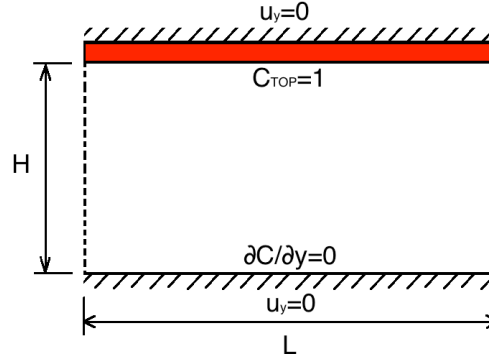


Figure 4: Boundary Conditions for constant solute concentration on top, and no solute flux on bottom

Results

Model Validation

The macroscopic equations used in this work are the same that are used in the Rayleigh-Benard thermal convection problem, where a top and bottom boundary that is held at a high and low temperature, respectively, leads to buoyancy-driven convective mixing. Wang et al. [10] used results (i.e. Nusselt numbers) of the spectral solution in [23] for the Rayleigh-Benard problem as a tool to compare the accuracy of their lattice Boltzmann modeling work. In our work, we also use the Nusselt numbers from the spectral solution in [23] as a validation tool. The following table compares our findings (for two different flows) against the corresponding spectral solution. Cases were run at different mesh resolutions for a Rayleigh number of 2,000 with a Prandtl number of 0.71, and for a Rayleigh number of 50,000 with a Prandtl number of 7.0. The Prandtl number is similar to the Schmidt number but refers to thermal diffusion.

Table 2: Comparison of Nusselt numbers for two different Rayleigh number simulations (using MRT) to Spectral Solution for Rayleigh-Benard Problem.

Mesh Resolution	Nu for Rayleigh=2,000, Pr=0.71		Nu for Rayleigh=50,000, Pr=7.0	
	This work	Spectral Solution	This work	Spectral Solution
21x11	1.1194871	1.212	3.2392477	3.894
41x21	1.1982246		3.7475402	
81x41	1.2066359		3.8639611	
161x81	1.2089087		3.898769	
321x161	1.2097955		3.917320	

By using the MRT scheme for simulation of the Rayleigh-Benard problem, we observe that it converges with second order accuracy, and that the error between this work's simulated Nusselt number to that of the spectral solution is decreasing as the mesh resolution is increased.

Time Evolution of Concentration Field

Set 1: Impact of Schmidt number

In this work, the development of convective fingering is of interest, and the following concentration fields are used to visualize the impact that the Rayleigh number and Schmidt number have on the convective and diffusive transport processes. In the first set of results, the Rayleigh number is set at 98,000, while the Schmidt number is 0.5 for case (1.a) and 300 for case (1.b). Both cases were run with a mesh resolution of 161 by 41 lattice nodes (an aspect ratio of 4:1), and since the non-dimensional height of the domain is 1, the lattice nodes have a spatial step of 0.025. The time step is 0.00025, and the non-dimensional force is 1. The boundary conditions are as explained in Figure 5. Results are shown at select times in order to visualize the evolution of the concentration field.

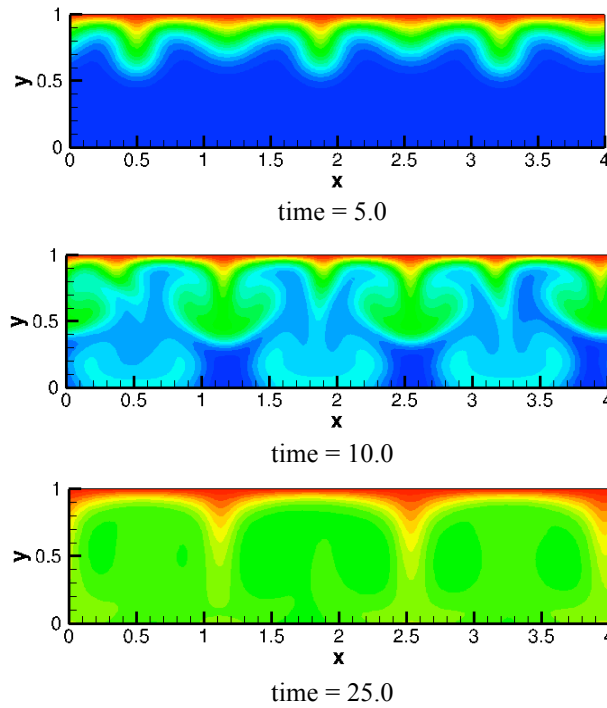
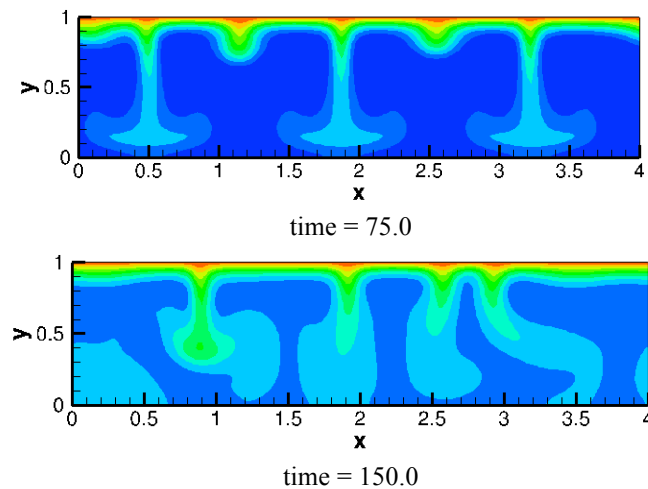


Figure 5: Concentration field for case (1.a) where Schmidt=0.5

In case (1.a), the Schmidt number was set to be below unity in order to simulate the case when the diffusivity is higher than the viscosity ($Schmidt = \nu/d$). The time evolution of the concentration field shows that the domain will approach a saturated state at an early time when the d parameter is high. This is because the flux of solute at the top boundary is high. The fields also reveal that when d is high, the convective fingers develop into wide fingers, which is likely influenced by the domain depth.



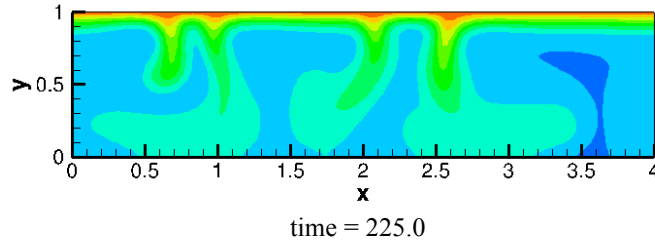


Figure 6: Concentration field for case (1.b), where Schmidt = 300

In contrast to case (1.a), the Schmidt number is set to be above unity for case (1.b) in order to visualize the impact of a high viscous to diffusive ratio (i.e. diffusivity is lower than viscosity). As the diffusivity is lowered, the solute flux across the top boundary is lowered as well, and thus in case (1.b) we observe that the domain does not approach a saturated state as quickly as it does in case (1.a). However, we observe that a lower diffusivity impacts the development of the convective fingers in that they are thinner and penetrate into the domain faster than in case (1.a). Also, over time these fingers do not merge into adjacent fingers as much as they do in case (1.a), and more individual fingers are sustained over time.

It is important to note that the development of the convective fingers shown in the above cases is influenced by the depth of the domain as well as the degree of turbulence in the fluid. Once the convective finger has lengthened enough to reach the bottom boundary, the advancing front of the convective finger cannot develop in the vertical direction (downwards) anymore, so it begins to develop in the horizontal direction. Then, the front of the finger moves vertically upward. In fact, during the whole process of the finger growth, upwelling of low-density fluid is occurring too, as the high-density finger is advancing into the domain. Initially, the advancing high-density fingers and the upwelling low-density fluid regions form symmetrical patterns along the width of the domain, which is especially noticed in case (1.a), until a point is reached in time when the convective mixing is no longer symmetrical. This is influenced by turbulent mixing, and at a high Rayleigh number, the convective fingers are less likely to develop symmetrically over time.

Set 2: Impact of Rayleigh number

Since the degree of turbulence is related to the Rayleigh number, the second set of results illustrate what happens when the Schmidt number is the same in two cases (Schmidt=1), but the Rayleigh number is either (2.a) 2,000 or (2.b) 1,000,000.

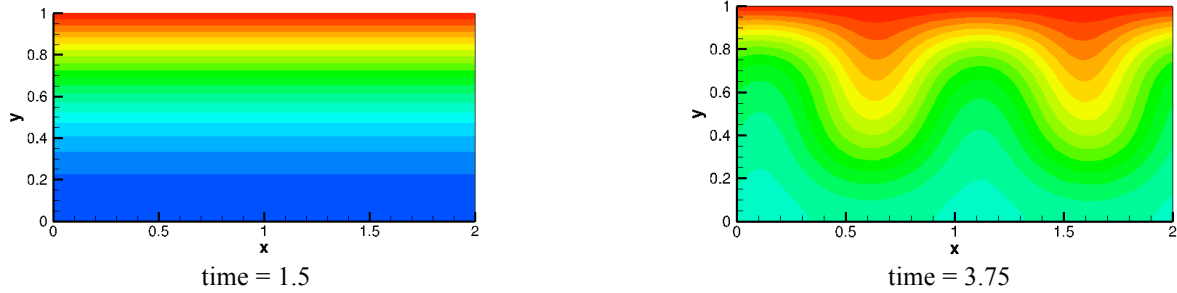


Figure 7: Concentration field for case (2.a) Rayleigh = 2,000

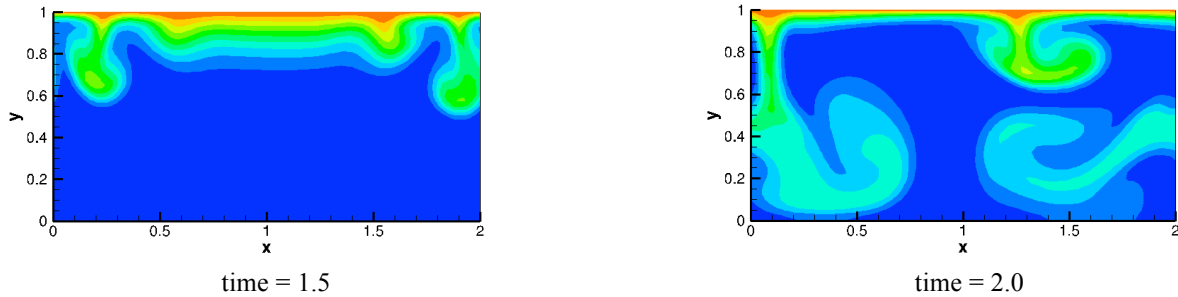


Figure 8: Concentration field for case (2.b) Rayleigh = 1,000,000

By comparing the concentration fields in Figure 7 and Figure 8, we observe that the system is less stable when Rayleigh number is high, and convective fingering is initiated earlier in time. In case (2.a), we can see that at time=1.5, no convective finger has developed, since the density difference is not large enough to produce any noticeable instability in the fluid yet. By time=3.75, fingers fill the domain in a symmetric pattern, and past that time the fingers continue to evolve in a symmetric pattern as the a saturated state is approached. In case (2.b), we can see that at time=1.5, convective fingers exist and are already turbulent. This early instability is caused by a higher density difference between solute-saturated fluid and pure fluid. Then, by time=2.0, turbulent convective mixing through out the whole domain is noticed.

Conclusion

We have used a new lattice Boltzmann equation to simulate density-driven convection, which is an important process that happens during CO₂ storage in saline aquifers. Although some studies (i.e. [8]) have attempted to apply the LBE for CO₂ storage simulations, the novelty of our work includes the use of a second order accurate LBE, the use of moment-based boundary, and a multiple relaxation time collision operator for simulating the density-driven flow of solute in water. Since the moment-based boundary condition ensures no-slip is simulated on a wall, it is the method of choice and is superior over the traditionally used bounce-back method. The results of this work give us the foundation to move onto applying density-driven convection in a porous domain.

Some general fluid flow physics and fingering development were noticed from the evolution of the concentration fields, including the observation that when diffusivity is large, convective fingers are wider and merge together quickly, resulting in fewer fingers over time. When diffusivity is small, the fingers are thinner and develop into long fingers quickly. Since the fingers are thinner in the small diffusivity case, the merging is less obvious and more individual fingers seem to be sustained as time evolves. Also noticed was the impact of the Rayleigh number, in that the system quickly moves to an unstable state with turbulent convective mixing when the Rayleigh number is high.

The results discussed in this work are important for CO₂ storage simulations. However, in order to increase the practical application of this work to the real-world case, we need to reflect on the typical aquifer characteristics of CO₂ storage projects. When modeling fluid flow behavior in an underground reservoir, temperature and pressure variations impact the hydrodynamics and the miscibility of solutes in fluid. In our work, the macroscopic equations were made non-dimensional with the Rayleigh and Schmidt numbers. In particular, the Schmidt number is a function of temperature, so it would be beneficial to investigate this relationship further and to incorporate non-isothermal conditions in our simulations.

Acknowledgements

The authors would like to acknowledge King Abdullah University of Science and Technology for providing the research opportunity for this work. The research of Dr Reis is supported by Award No. KUK-C1-013-04 by King Abdullah University of Science and Technology.

References

1. EGUMedia, *Geothermal energy versus CO₂-storage: can we use the underground more than once?*, in *EGU2011 Webstreaming of General Assembly* 2011.
2. Farajzadeh, R., et al., *Numerical simulation of density-driven natural convection in porous media with application for CO₂ injection projects*. International Journal of Heat and Mass Transfer, 2007. **50**(25-26): p. 5054-5064.
3. Kneafsey, T.J. and K. Pruess, *Laboratory flow experiments for visualizing carbon dioxide-induced, density-driven brine convection*. Transport in porous media, 2010. **82**(1): p. 123-139.
4. Slim, A.C. and T. Ramakrishnan, *Onset and cessation of time-dependent, dissolution-driven convection in porous media*. Physics of Fluids, 2010. **22**: p. 124103.
5. Neufeld, J.A., et al., *Convective dissolution of carbon dioxide in saline aquifers*. Geophys. Res. Lett, 2010. **37**.
6. Ennis-King, J. and L. Paterson, *Role of convective mixing in the long-term storage of carbon dioxide in deep saline formations*. SPE Journal, 2005. **10**(3): p. 349-356.
7. Class, H., et al., *A benchmark study on problems related to CO₂ storage in geologic formations*. Computational Geosciences, 2009. **13**(4): p. 409-434.
8. Chen, C. and D. Zhang, *Pore-scale simulation of density-driven convection in fractured porous media during geological CO₂ sequestration*. Water Resources Research, 2010. **46**(11): p. W11527.
9. Pan, C., L.S. Luo, and C.T. Miller, *An evaluation of lattice Boltzmann schemes for porous medium flow simulation*. Computers & fluids, 2006. **35**(8): p. 898-909.

10. Wang, J., et al., *Lattice Boltzmann simulations of thermal convective flows in two dimensions*. Computers and Mathematics with Applications, 2012.
11. Ginzburg, I., F. Verhaeghe, and D. d'ÄôHumieres, *Study of simple hydrodynamic solutions with the two-relaxation-times lattice Boltzmann scheme*. Communications in Computational Physics, 2008. **3**(3): p. 519-581.
12. Guo, Z., C. Zheng, and B. Shi, *Discrete lattice effects on the forcing term in the lattice Boltzmann method*. Physical Review E, 2002. **65**(4): p. 46308.
13. Weir, G.J., S.P. White, and W.M. Kissling, *Reservoir storage and containment of greenhouse gases*. Transport in porous media, 1996. **23**(1): p. 37-60.
14. He, X., X. Shan, and G.D. Doolen, *Discrete Boltzmann equation model for nonideal gases*. Physical Review E, 1998. **57**(1): p. 13-16.
15. Dellar, P.J., *Incompressible limits of lattice Boltzmann equations using multiple relaxation times*. Journal of Computational Physics, 2003. **190**(2): p. 351-370.
16. Ginzburg, I., *Equilibrium-type and link-type lattice Boltzmann models for generic advection and anisotropic-dispersion equation*. Advances in Water Resources, 2005. **28**(11): p. 1171-1195.
17. Ginzburg, I., *Generic boundary conditions for lattice Boltzmann models and their application to advection and anisotropic dispersion equations*. Advances in Water Resources, 2005. **28**(11): p. 1196-1216.
18. Bennett, S., *A Lattice Boltzmann model for diffusion of binary gas mixtures*. 2010.
19. Bennett, S., P. Asinari, and P.J. Dellar, *A lattice Boltzmann model for diffusion of binary gas mixtures that includes diffusion slip*. International journal for numerical methods in fluids, 2011.
20. He, X., et al., *Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice Boltzmann BGK model*. Journal of Statistical Physics, 1997. **87**(1): p. 115-136.
21. Ginzburg, I. and D. d'ÄôHumieres, *Multireflection boundary conditions for lattice Boltzmann models*. Physical Review E, 2003. **68**(6): p. 066614.
22. Ginzbourg, I. and P. Adler, *Boundary flow condition analysis for the three-dimensional lattice Boltzmann model*. Journal de Physique II, 1994. **4**(2): p. 191-214.
23. Clever, R. and F. Busse, *Transition to time-dependent convection*. J. Fluid Mech, 1974. **65**(4): p. 625-645.

RECENT REPORTS

12/79	Fast and Accurate Computation of Gauss-Legendre and Gauss-Jacobi Quadrature Nodes and Weights	Hale Townsend
12/80	On the spectral distribution of kernel matrices related to radial basis functions	Wathen Zhu
12/81	Inner product computation for sparse iterative solvers on distributed supercomputer	Zhu Gu Liu
12/82	A new pathway for the re-equilibration of micellar surfactant solutions	Griffiths Breward Colegate Dellar Howell Bain
12/83	Object-Oriented Paradigms for Modelling Vascular Tumour Growth: a Case Study	Connor Cooper Byrne Maini McKeever
12/84	Chaste: an open source C++ library for computational physiology and biology	Mirams Arthurs Bernabeu Bordas Cooper Corrias Davit Dunn Fletcher Harvey Marsh Osborne Pathmanathan Pitt-Francis Southern Zemzemi Gavaghan
12/85	A two-pressure model for slightly compressible single phase flow in bi-structured porous media	Schlackow Marguerat Proudfoot Bähler Erban Gullerova
12/86	Boolean modelling reveals new regulatory connections between transcription factors orchestrating the development of the ventral spinal cord	Lovrics Gao Juhász Bock Byrne Dinnyés Kovács
12/87	Asymptotic solutions of glass temperature profiles during steady	Taroni

12/93	Diffusion of multiple species with excluded-volume effects	Bruna Chapman
12/94	The Mechanics of a Chain or Ring of Spherical Magnets	Hall Vella Goriely
12/95	On-Lattice Agent-based Simulation of Populations of Cells within the Open-Source Chaste Framework	Figueredo Joshi Osborne Byrne Owen
12/96	Mathematical Biomedicine and Modeling Avascular Tumor Growth	Byrne
12/97	Inference of the genetic network regulating lateral root initiation in <i>Arabidopsis thaliana</i>	Muraro Voß Wilson Bennett Byrne De Smet Hodgman King
12/98	Axisymmetric bifurcations of thick spherical shells under inflation and compression	deBotton Bustamante Dorfmann
12/99	Calculus from the past: Multiple Delay Systems arising in Cancer Cell Modelling	Wake Byrne
12/100	Nonlocal models of electrical propagation in cardiac tissue: electrotonic effects and the modulated dispersion of repolarization	Bueno-Orovio Kay Grau Rodriguez Burrage
12/101	Microfluidic Immunomagnetic Multi-Target Sorting A Model for Controlling Deflection of Paramagnetic Beads	Tsai Griffiths Stone

Copies of these, and any other OCCAM reports can be obtained from:

**Oxford Centre for Collaborative Applied Mathematics
Mathematical Institute
24 - 29 St Giles'
Oxford
OX1 3LB**

England
www.maths.ox.ac.uk/occam