POLITECNICO DI TORINO
Repository ISTITUZIONALE

Lattice Boltzmann Method for pore-scale multi-phase flow simulation in Underground Porous Media: a review

Original
Lattice Boltzmann Method for pore-scale multi-phase flow simulation in Underground Porous Media: a review / Raeli, A., Salina Borello, E., Viberti, D.. - In: GEAM. GEOINGEGNERIA AMBIENTALE E MINERARIA. - ISSN 1121-9041. - ELETTRONICO. - 171(2024), pp. 21-36. [10.19199/2024.171.1121-9041.021]

Availability:
This version is available at: 11583/2999157 since: 2025-04-14T12:53:35Z

Publisher:
Patron

Published
DOI:10.19199/2024.171.1121-9041.021

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

26 November 2026

DX.DOI.ORG//10.19199/2024.171.1121-9041.021

Lattice Boltzmann Method for pore-scale multi-phase flow simulation in Underground Porous Media: a review

Alice Raeli*
Eloisa Salina Borello*
Dario Viberti*

* Dipartimento di Ingegneria dell'Ambiente, del Territorio e delle Infrastrutture, DIATI, Politecnico di Torino

Corresponding author:
alice.raeli@polito.it

The study of fluid flow in porous media is relevant in various applications from chemical to environmental and geoenergy engineering. Understanding and properly simulating multi-phase flow phenomena at both micro and macro scales is fundamental in particular in the view of underground storage of fluids like methane, hydrogen, and CO_2. This work aims at resuming the idea behind the Lattice Boltzmann Method (LBM) for modeling multi-phase flows through underground porous media, aiming at highlighting the advantages, and the possible approaches for single-phase and multi-phase flow. Moreover, an algorithmic analysis is given in single-phase and compared to more popular multi-phase Lattice Boltzmann models proposed and their main computing strategies. Direct numerical simulation (DNS) of flow in porous structures has become popular in recent years due to the novel high-performance computing architectures, which involve many cores or threads in calculation. This work concludes with an overview of the most common open-source codes available online to approach porous media flow simulation in this context and emphasize the most common for HPC applications.

***Keywords**: Underground Porous Media, multi-phase fluid flow, Lattice Boltzmann Method, parallel computing.*

1. Introduction

The pore structure characterization (Blunt, 2001; Blunt *et al.*, 2013; Dong and Blunt, 2009; Ghanbarian *et al.*, 2013; Panini, 2022; Panini *et al.*, 2024; Raeli *et al.*, 2024; Salina Borello *et al.*, 2022; Viberti *et al.*, 2020) is fundamental in reservoir engineering for a thorough characterization and understanding of hydrocarbon production, Underground Gas Storage (UGS), Underground Hydrogen Storage (UHS), CO_2 storage or sequestration, Enhanced Oil Recovery, and groundwater flow. Porous media are complex systems presenting pore and grain dimensions varying over a wide range (Ghanbarian *et al.*, 2013), that affect the transport properties of the medium (Huang *et al.*, 2021); moreover, the flow phenomena comprehension requires hydrodynamic numerical simulations, most of them discretizing modified versions of Navier-Stokes Equations (NSE), among which lies the Lattice Boltzmann Method (LBM) (Krüger *et al.*, 2017; Succi, 2001). In particular, multi-phase flow phenomena are often modeled using the macro-scale (Liu *et al.*, 2016) simulations, which do not capture the micro-scale effects of the flow; in contrast, they require observation at the micro-scale (Blunt, 2001; Venturoli and Boek, 2006). The Lattice Gas Automata method (McNamara and Zanetti, 1988), from which LBM derives, has been used to estimate permeability, tortuosity, and effective porosity in synthetic 2D porous media (Koponen *et al.*, 1997, 1996). The application of LBM in geometric porous media characterization involves tortuosity and permeability in 2D and 3D (Blunt, 2001; Dong and Blunt, 2009; Eshghinejadfard *et al.*, 2016; Ghassemi and Pak, 2011; Pan *et al.*, 2006; Panini, 2022; Yang *et al.*, 2017) test cases, simulating single-phase and two-phase flows (Ferréol and Rothman, 1995; Xu and Liu, 2018). Recent works also involve LBM in porous media simulating effective diffusion gas phenomena (Huang *et al.*, 2021), non-stationary fluid flows in homogenized cases (Simonis *et al.*, 2023), including the water displacement by CO_2 at reservoir conditions (Chen *et al.*, 2019, 2018), advection-diffusion applications including chemical reactions (Liu *et al.*, 2023), flow and solute transport problems (Boek *et al.*, 2014), and computing diffusive tortuosity (Zahedi and Vakili, 2023; Zhang *et al.*, 2021).

LBM is a family of popular mesoscopic methods based on microscopic fluid fictitious particle models and mesoscopic kinetic equations. The advantage concerning continuum modeling approaches, based on Darcy velocity, is the possibility of handling in a relatively simple way complex geometries (Meskas and Bao, 2014) including porous media domains. The LBM implementation is a valuable approach to pore scale simulation. Unlike computational Fluid Dynamics (CFD) methods, which numerically solve the conservation equations of

If there are references to colour figures in the text, the articles are available in open-access mode on the site www.geam-journal.org

macroscopic variables (i.e., mass, momentum, and energy), LBM describes the evolution of the velocity distribution of fictitious particles (Krüger *et al.*, 2017; Song *et al.*, 2022; Succi, 2001). The fluid is conceived as a set of fictitious particles that represent packets of fluid particles; the motion of particles is tracked by a distribution function over a discrete domain, composed of elements called lattices. The macroscopic variables (density, momentum, energy) are obtained by integrating over each lattice the particle distribution function combined with the velocity (Blunt, 2001; Krüger *et al.*, 2017; Liu *et al.*, 2016; Succi, 2001). To model the Navier-Stokes equations properly, a choice of a collision operator is necessary, depending on the application e.g. distinguishing miscible and immiscible fluid interactions. The LBM is a relatively new method (Hardy *et al.*, 1973) and it became popular over the years for its versatility and algorithm simplicity (Latt *et al.*, 2021b, 2021a; Liu *et al.*, 2023; Simonis *et al.*, 2023). The greater advantages of LBM are: being suitable for massive parallelization in computation (Latt *et al.*, 2021a; Liu *et al.*, 2016), its applicability to multi-phase or multi-component flows, and the handling of complex geometries and complex boundaries (Sukop and Thorne, 2006). On the other hand, the uniform grid has to be fine enough to capture the physical phenomena in porous media, and large enough to limit the high computational cost still required in LBM simulations (Chen *et al.*, 2018). Compared to Pore Network Modelling (PNM), it has been demonstrated that, in the context of multi-phase flow in porous media, LBM is more time-consuming, but, more efficient on the microstructure transport properties computation (García-Salaberri *et al.*, 2015; Huang *et al.*, 2021).

Being a particle-based solver, LBM does not discretize directly the Naver-Stokes equation for fluid motion (Krüger *et al.*, 2017). However, it satisfies the incompressible Navier-Stokes equations (Huang *et al.*, 2015; Succi, 2001) in the limit of small Knudsen and Mach numbers. Thus, LBM is suitable for simulating all phenomena that can be addressed with NS (Junk and Klar, 2000). The Lattice Boltzmann Method can be seen as a discretization of the Boltzmann transport equation by a finite difference method (FDM) in space, on a uniform Cartesian grid, and in time derivatives by a fixed time step (Meskas and Bao, 2014). The regularity in space and time of this method are the main drawbacks. The most challenging part of using finite difference methods is the handling of the non-linearities of the operators and the complex geometries in boundaries and internal discontinuities (Raeli, 2017; Raeli *et al.*, 2018). However, Cartesian grids, defined as *lattice* for LBM and representing a group of particles in the mesoscale, are characterized by squared (2D) or cubic (3D) structures on which there are two natural choices to discretize differential operators: vertex-centered, common for unstructured mesh but rarely used in LBM simulation (Pellerin *et al.*, 2017), or cell-centered (Krüger *et al.*, 2017).

This work aims to give an overview of the LBM method with a focus on the applications to multi-phase fluid flows (Liu *et al.*, 2016) in underground porous media, including the model discretization and the possible choices in boundary conditions (Yin and Zhang, 2012; Zou and He, 1997) and distribution functions (Grunau *et al.*, 1993; Gunstensen *et al.*, 1991; He *et al.*, 1999; Lee and Liu, 2010; Shan and Chen, 1993; Swift *et al.*, 1996). Moreover, a series of open-source libraries and software dedicated to LBM is presented. The list, potentially huge, is focused on application to porous media and is limited to the most used tools, based on citations and online availability (Chen *et al.*, 2018; Latt *et al.*, 2021b; Simonis *et al.*, 2023).

2. The Lattice Boltzmann Model

The Lattice Boltzmann Method derives from the Lattice Gas Automata (LGA) (Frisch *et al.*, 1986) method. The main idea is to simulate gases/fluids as a collection of particles with random motions interacting in two main steps of streaming and collision (Liu *et al.*, 2016). The first Lattice Gas Automaton model was developed in 1973 by Hardy, de Pazzis, and Pomeau (HPP). It simulates gas particles on grid intersections, then each particle can populate one of the directions connecting the cells with a neighbour (Hardy *et al.*, 1973). The description of fluids at the mesoscopic scale lies between the individual particle tracking (microscopic) and the continuum theory (macroscopic). The mesoscale simulations capture molecular details of the phenomena and guarantee an accurate description of the macro-scale physics because mass conservation and momentum conservation are ensured. Moreover, other constraints, such as miscibility and intermolecular fluid interactions, play a significant role in multi-phase flows (Huang *et al.*, 2015; Shan and Chen, 1993; Swift *et al.*, 1996).

The whole domain where the particles interact is called a *lattice* and it is discretized in a regular mesh. Each mesh element, representing the space occupied by a particle (or a group of particles), is called the *lattice cell* (or *node*). The velocity directions are also discretized; the number of velocity directions allowing movement from one lattice cell to another determines the number of neighbours for each lattice cell.

We can briefly describe the two molecular phases considered by LBM and deriving methods in independent steps (Krüger *et al.*, 2017) occurring during a time step:

1. Collision (or relaxation) step: particles are represented after collisions in function of their position and their equilibrium distributions. The macroscopic quantities are used to compute the equilibrium distribution. The collision is a local operation on the lattice cell.
2. Streaming step: post-collisional distributions move to another subsequent spatial position according to a predefined set of velocity directions. The streaming is associated with neighbouring cells.

The equations referring to LBM collision/streaming are detailed in Section 2.3. The LBM discretization schemes are usually identified by *DdQq*, where *d* = 1,2,3,..., stands for the dimension of the problem, and *q* is the number of neighbours involved for an internal cell. By way of example, *D*2*Q*9, which is characterized by 9 neighbours including the one (at rest) in the centre of the lattice cell (Figure 1a), is the most used discretization in 2D, while *D*3*Q*19, which is characterized by 19 neighbours, is the most common discretization in 3D (Figure 1b). However, the chosen discretization often depends on the minimum number of constraints necessary for being accurate; for instance, the *D*3*Q*7 model has been applied by (Huang *et al.*, 2021) computing the parameters to obtain the effective diffusivity of gas in porous media by the Fick's law.

For the sake of simplicity, we can figure out each lattice cell as the grid cell on which the computation lies. This ensures discretized positions for a single-particle distribution function $f(\boldsymbol{x}, \boldsymbol{u}, t)$ which represents the probability of a particle having velocity $\boldsymbol{u}$ to hold the position $\boldsymbol{x}$ at the established time t.

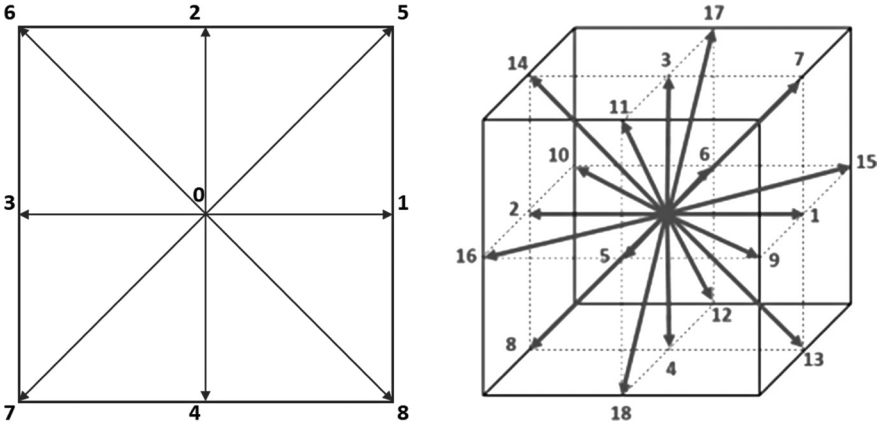

Fig. 1 – Lattice cell velocity directions examples D2Q9 (left) and D3Q19 (right – Panini, 2022).

The Boltzmann transport equation (1) describes the evolution in time of $f(\boldsymbol{x}, \boldsymbol{u}, t)$ (Krüger *et al.*, 2017; McNamara and Zanetti, 1988; Swift *et al.*, 1996) redefining the distribution function at the considered time. The equation is expressed as:

$$\frac{\partial f}{\partial t} + \boldsymbol{u} \cdot \nabla f + \frac{\boldsymbol{F}}{m} \cdot \nabla_{\boldsymbol{u}} f = \Omega\left(f, f^{eq}\right) \quad (1)$$

where $\boldsymbol{u}(\boldsymbol{x}, t)$ is the macroscopic particle velocity; $\Omega(f, f^{eq})$ is the collision operator describing the relaxation to the local equilibrium state of particles or, in other terms, the non-linear collisional dynamics, m represents the point masses of the particles set, and $\boldsymbol{F}$ the external force contribution.

Macroscopic quantities such as density, velocity, and energy can be obtained by the distribution function as integrals combining the distribution function with the velocity. We omit the expressions here because they are widely presented in the literature (Blunt, 2001; Krüger *et al.*, 2017; Liu *et al.*, 2016; Succi, 2001) for the interested reader; their discretized versions, as local sums, are reported in this paper in Subsection 2.2.

Any solution of the Boltzmann equation requires the expression of the collision operator. In this context, we define the equilibrium distribution function f^{eq} as the particle distribution such that the local equilibrium is satisfied verifying the formula (Simonis *et al.*, 2023):

$$\Omega(f, f^{eq}) = 0. \quad (2)$$

Different formulations for the collision operator have been proposed for the Lattice Boltzmann Equation (LBE) according to the scope of the resolution and the complexity of the problem:

Single Relaxation Time (SRT), also called Bhatnagar-Gross-Krook (BGK) (Bhatnagar *et al.*, 1954), is widely used for monophase flow in porous media simulation; however, it is affected by numerical instabilities and strong viscosity dependencies on the boundary (Chen *et al.*, 2019; Pan *et al.*, 2006).

Two Relaxation Time (TRT) and Multiple Relaxation Time (MRT) were introduced in order to improve the stability of the method (Lallemand and Luo, 2000).

The equilibrium distribution (f^{eq}) is a function of density ρ, gas constant R, temperature T, microscopic velocity $\boldsymbol{c}$, and macroscopic velocity $\boldsymbol{u}$. Its expression is the Maxwellian distribution function (He and Luo, 1997a) :

$$f^{eq} = \frac{\rho}{\left(2\pi RT\right)^{\frac{d}{2}}} \exp^{-\frac{(\boldsymbol{c}-\boldsymbol{u})^2}{2RT}} \quad (3)$$

where d is the dimension number.

In the following sections (2.1-2.3) we introduce the discretiza-

tion of equations (1) and (3) in single-phase adopting a single relaxation time. Boundary conditions are addressed in Section 2.4. The generalization to multi-phase models is addressed in Section 3, where the most popular multi-phase LBM models are presented.

2.1. The single-phase Lattice BGK discrete equation

The Bhatnagar-Gross-Krook (BGK) or Single Relaxation Time (SRT) collision operator is defined as

$$\Omega\left(f, f^{eq}\right) = -\frac{\left(f - f^{eq}\right)}{\tau} \quad (4)$$

where f is the distribution function of particles, f^{eq} is the equilibrium function defined in the previous section, and τ is a non-dimensional parameter called relaxation time, which is associated with the collisional relaxation to the local equilibrium and the kinematic viscosity of the fluid (Huang *et al.*, 2015). The value of τ influences the stability of the system. It can be determined in function of the Reynolds number (Jadidi *et al.*, 2022), but, more in general, its value has to overcome $\Delta t/2$ and stands below Δt to ensure stability (Krüger *et al.*, 2017). Equation (4) can be interpreted as the tendency of the particles described by f to approach the equilibrium state f^{eq} after a fictitious time step τ. Combining equation (4) with equation (1) we derive the BGK Boltzmann equation (BGKBE) as:

$$\frac{\partial f}{\partial t} + \boldsymbol{u}\cdot\nabla f + \boldsymbol{F}\cdot\nabla_{\boldsymbol{u}} f = -\frac{\left(f - f^{eq}\right)}{\tau} \quad (5)$$

The term $f - f^{eq}$ is the non-equilibrium distribution f^{neq}, used to connect the mesoscopic to the macroscopic scales in Chapman-Enskog analysis (Ihle, 2009). We use the definition of material derivative $\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \boldsymbol{u}\cdot\nabla f$ to rewrite equation (5) applying a first-order finite difference discretization:

$$\frac{Df}{Dt} = \frac{f\left(x + \boldsymbol{u}\Delta t, \boldsymbol{u}, t + \Delta t\right) - f\left(x, \boldsymbol{u}, t\right)}{\Delta t} + O\left(\Delta t^2\right) \quad (6)$$

The lattice for a generic scheme *DdQq* is a uniform cartesian grid data structure, on which each cell has a limited number of neighbours $i = 0,\ldots,q-1$ to which a particle moves following a tensor velocity $\boldsymbol{c}_i$, obtaining the displacement in space $\boldsymbol{x} + \boldsymbol{c}_i\Delta t$ at the subsequent time step $t + \Delta t$. So, we obtain the discretized Lattice Boltzmann equation with BGK collision (LBGK equation) from (6) as:

$$f_i\left(x + \boldsymbol{c}_i\Delta t, t + \Delta t\right) - f_i\left(x,t\right) = -\frac{\Delta t}{\tau}\left(f_i\left(x,t\right) - f_i^{eq}\left(x,t\right)\right) + S_i\left(x,t\right) \quad (7)$$

where $S_i(\boldsymbol{x}, t)$ is the source term induced by external forces (Lallemand and Luo, 2000) such as gravity or intermolecular interactions.

The distribution function $f(\boldsymbol{x}, \boldsymbol{u}, t)$, is discretized in each velocity direction (i), obtaining the so-called discrete velocity distribution function ($f_i(\boldsymbol{x}, t)$), which represents the probability of streaming in the i-th direction. As an example, the D2Q9 model (Figure 2) on two dimensions involves 9 neighbours, on which we associate 9 lattice velocity vectors $\boldsymbol{c}_i$, $i \in \{0, \ldots, 8\}$ to move by a $\Delta x = 1$ (*lattice unit*) in the direction of the neighbours respectively (microscopic velocities):

$$\boldsymbol{c}_0 = (0, 0),\ \boldsymbol{c}_{1,3} = (\pm\delta_x, 0),\ \boldsymbol{c}_{2,4} = (0, \pm\delta_y),\ \boldsymbol{c}_{5,6,7,8} = (\pm\delta_x, \pm\delta_y) \quad (8)$$

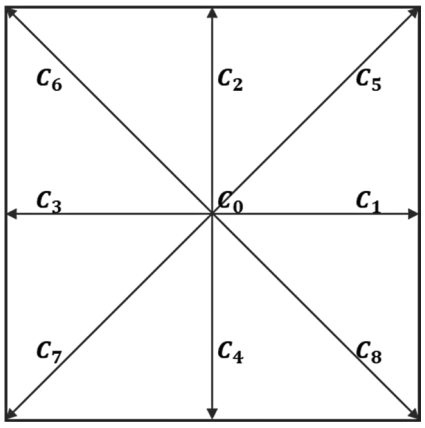

Fig. 2 – Lattice normal velocities in the D2Q9 scheme.

For small Mach numbers, the exponential function in equation (3) can be approximated to obtain a discretized lattice cell formulation by polynomials as:

$$f_i^{eq}\left(x,t\right) = \omega_i\rho\left[1 + \frac{\boldsymbol{c}_i\cdot\boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{c}_i\cdot\boldsymbol{u})^2}{2c_s^4} - \frac{\boldsymbol{u}^2}{2c_s^2}\right] \quad (9)$$

where c_s is the sound speed in lattice space, obtained in isothermal conditions as $c_s^2 = p/\rho$, according to the equation of state presented in Section 2.2. Its value changes with the adopted *DdQq* scheme. In the most common schemes (among them *D2Q9* and *D3Q19*) the value is set as $c_s = \sqrt{\frac{1}{3}}$ (Krüger *et al.*, 2017). The discretization quadrature weights ω_i can be derived taking into account the weighted association schemes (He and Luo, 1997b):

- the sum of the weights is equal to 1: $\sum_i\omega_i = 1$;
- the sum of the weighted discrete velocities is 0 for even velocities occurrences: $0 = \sum_i\omega_i\,\boldsymbol{c}_{i\alpha} = \sum_i\omega_i\,\boldsymbol{c}_{i\alpha}\,\boldsymbol{c}_{i\beta}\boldsymbol{c}_{i\gamma} = \sum_i\omega_i\,\boldsymbol{c}_i\alpha\,c_{i\beta}\,\boldsymbol{c}_{i\gamma}\,\boldsymbol{c}_{i\nu}\,\boldsymbol{c}_{i\mu} = \cdots$
- the sum of the weighted discrete velocities is related to the Kronecker delta and powers of sound speed lattice velocity for odd velocities occurrences:

$$\sum_i\omega_i\boldsymbol{c}_{i_\alpha}\boldsymbol{c}_{i_\beta} = c_s^2\delta_{\alpha\beta}, \sum_i\omega_i\boldsymbol{c}_{i_\alpha}\boldsymbol{c}_{i_\beta}\boldsymbol{c}_{i_\gamma}\boldsymbol{c}_{i_\nu} = c_s^4\left(\delta_{\alpha\beta}\delta_{\gamma\nu} + \delta_{\alpha\gamma}\delta_{\beta\nu} + \delta_{\alpha\nu}\delta_{\beta\gamma}\right), \ldots$$

More details are easily available in (Huang *et al.*, 2015; Krüger *et al.*, 2017; Lallemand and Luo, 2000; Meskas and Bao, 2014; Succi, 2001; Sukop and Thorne, 2006). The resulting weights reported for examples for the D2Q9 and D3Q19 schemes are given in Table 1 and Table 2 respectively.

Tab. 1 – Weights and lattice velocities for the D2Q9 scheme.

i	0	1	2	3	4	5	6	7	8
ω_i	4/9	1/9	1/9	1/9	1/9	1/36	1/36	1/36	1/36
$\boldsymbol{c}_{i,x}$	0	1	0	–1	0	1	–1	–1	1
$\boldsymbol{c}_{i,y}$	0	0	1	0	–1	1	1	–1	–1

2.2. Macroscopic variables computation

Once the LBE is solved and the values are calculated for the equilibrium state f_i^{eq}, we have to compute the macroscopic fluid density $\rho(\boldsymbol{x}, t)$ and momentum, which implies the macroscopic velocity $\boldsymbol{u}(\boldsymbol{x}, t)$. The macroscopic fluid density follows the mass conservation principle over the lattice cell:

$$\rho(\boldsymbol{x},t) = \sum_{i=0}^{8} f_i(\boldsymbol{x},t) \qquad (10)$$

We can accordingly describe the macroscopic velocity using the momentum conservation principle:

$$u(\boldsymbol{x},t) = \frac{1}{\rho(\boldsymbol{x},t)} \sum_{i=0}^{8} \boldsymbol{c}_i f_i(\boldsymbol{x},t). \qquad (11)$$

Together eq. (10) and eq. (11) describe the discrete conservation laws, that are locally conserved in any collision process and recomputed during new solutions. In isothermal conditions, we can compute the fluid pressure as a function of the density and of the speed of sound presented in equation (9) according to the equation of state:

$$p = c_s^2 \rho. \qquad (12)$$

2.3. Collision and Streaming

Analogously to the particle-based methods (such as LGA and CA), the LBM is performed by two main steps: collision and streaming. The equilibrium variables are computed into the collision step, simulating the interactions among particles moving within each lattice cell. The resulting new distribution function f_i^{new} is propagated to the neighbours to populate a new collision step and this second localization of particles is called streaming. We assume in lattice units $\Delta x = \Delta t = 1$ (Krüger *et al.*, 2017) and we omit for simplicity the source terms, then we rewrite equation (7) as:

$$f_i(\boldsymbol{x}+\boldsymbol{c}_i, t+1) = \left(1-\frac{1}{\tau}\right) f_i(\boldsymbol{x},t) + \frac{1}{\tau} f_i^{eq}(\boldsymbol{x},t) \qquad (13)$$

That is the Lattice Boltzmann discrete equation with the BGK collision operator (discrete BGKLBE). In LBM the above non-dimensionalization induces the CFL (Courant-Friedrichs-Lewy) coefficient to 1. The CFL condition in the numerical discretization of partial differential equations is a constraint for stability and convergence (Liu, 2020). We can split the equation (13) in two steps:

$$\begin{aligned} & f_{i_{new}}(\boldsymbol{x},t) = \\ & = \left(1-\frac{1}{\tau}\right) f_i(\boldsymbol{x},t) + \frac{1}{\tau} f_i^{eq}(\boldsymbol{x},t) \qquad \text{(collision)} \\ & f_i(\boldsymbol{x}+\boldsymbol{c}_i, t+1) = \\ & = f_{i_{new}}(\boldsymbol{x},t) \qquad \text{(streaming)} \end{aligned} \qquad (14)$$

where $f_i(\boldsymbol{x}, t)$ is the pre-collision distribution function, $f_i^{eq}(\boldsymbol{x}, t)$ is the equilibrium defined in equation (9), $f_i^{new}(\boldsymbol{x}, t)$ is the post-collision distribution that is imposed as pre-collision distribution for the subsequent time step $f_i(\boldsymbol{x} + \boldsymbol{c}_i, t + 1)$ during the streaming phase. Streaming and collision together (eq. (14)) describe how the population evolves within a lattice timestep ($t + 1$) through direction $\boldsymbol{c}_i \Delta t = \boldsymbol{c}_i$.

2.4. Boundary Conditions

Computation of boundary conditions in LBM simulation of porous media is often challenging (Xu, 2014; Yin and Zhang, 2012; Zou and He, 1997). Moreover, LBM can be seen as a discretization of the Boltzmann transport equation by a finite difference method (FDM) (Meskas and Bao, 2014). Thus, it inherits the FD common problem of defining a mesh fine enough to capture complex geometries (Coco and Russo, 2018; Min *et al.*, 2006). In porous media, the pore space geometry changes abruptly in the entire domain (Ghanbarian *et al.*, 2013; Raeli *et al.*, 2024; Salina Borello *et al.*, 2022), and the stability and

Tab. 2 – Weights and lattice velocities for the D3Q19 scheme.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ω_i	1/3	1/18	1/18	1/18	1/18	1/18	1/18	1/36	1/36	1/36	1/36	1/36	1/36	1/36	1/36	1/36	1/36	1/36	1/36
$\boldsymbol{c}_{i,x}$	0	1	–1	0	0	0	0	1	–1	1	–1	0	0	1	–1	1	–1	0	0
$\boldsymbol{c}_{i,y}$	0	0	0	1	–1	0	0	1	–1	0	0	1	–1	–1	1	0	0	1	–1
$\boldsymbol{c}_{i,z}$	0	0	0	0	0	1	–1	0	0	1	–1	1	–1	0	0	–1	1	–1	1

convergence of finite difference methods can be affected by inexact boundary conditions due to the propagation errors (Raeli, 2017).

In the scenario of flow in porous media, LBM boundary conditions regard the populations of the fluid region. The boundary reflects the solid-fluid interface along the pores as well as the inlet/outlet zone. We omit in this work the periodic boundary conditions, due to their poor application in porous media.

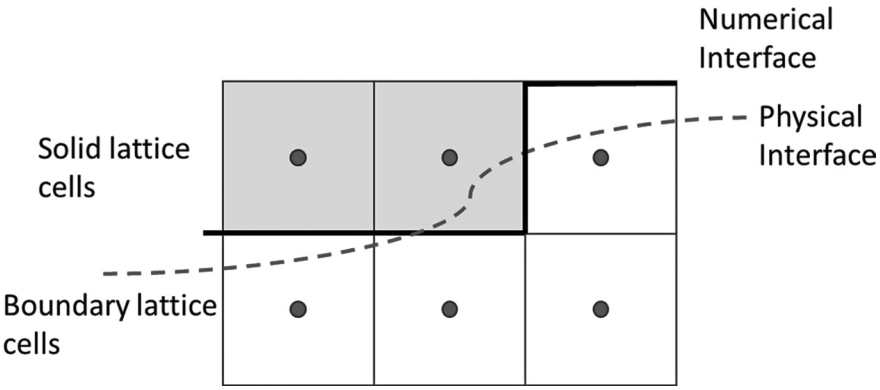

Fig. 3 – Solid-fluid boundary approximation.

2.4.1. Solid-fluid interface

Along the pores, the physical interface between grains and fluid is represented by solid lattice cells and boundary lattice cells. A simplified example is presented in Figure 3. The grains act as solid walls, and zero bulk velocity is imposed both across and along the boundary. This is the so-called no-slip velocity condition. It can be implemented using the fullway bounce-back method (Meskas and Bao, 2014; Succi, 2001; Yin and Zhang, 2012) or the halfway bounce-back method (Jiang *et al.*, 2021; Meskas and Bao, 2014; Pan *et al.*, 2006).

Both approaches assume that the boundary is located between solid lattice and boundary lattice cells (Krüger *et al.*, 2017). The two approaches, fullway and halfway, provide similar results in steady state and significant differences in time-depending problems.

In the fullway bounce-back, particles are streamed to the solid lattice cells, where the collision step reflects their velocities; then, they are streamed back to boundary cells (Figure 4).

As an example of fullway bounce-back, we consider the 2D wall configuration in Figure 4 with the D2Q9 scheme. At a given time step three distribution functions (orange, green, and black coloured) would leave the computational domain; we follow the green one, $f_2(x, y, t-1)$, streamed to the new location. During the collision, it is inversed and $f_4^{new}(x, y+1, t) = f_2(x, y+1, t)$ is computed. During the streaming the particles come back into the region with the same, but opposite, velocities for the subsequent collision $f_2(x, y, t+1)$. This implies a total reflection of the velocity $f_4(x + \boldsymbol{c}_4, t+1) = f_2(x, t)$ (Krüger *et al.*, 2017), see equations (7)-(14). A fullway bounce-back condition is relatively easy to implement, we only need to check whether we are on a solid node or not (Krüger *et al.*, 2017). It can be used for complex geometries as porous media; however, being first-order accurate, it can introduce further errors when the refinement is not fine enough to capture the pore geometry (Pan *et al.*, 2006).

Halfway bounce-back conditions are second-order accurate (Huang *et al.*, 2015; Krüger *et al.*, 2017). However, their implementation is more complicated because dependent on the geometry: the streaming step differs for populations propagating towards a solid node (bounce-back) compared to populations propagating towards an internal or boundary node (which implies normal streaming).

In Figure 5 the halfway bounce-back criterium is depicted focusing on f_2 (the green arrow): during the streaming phase the particles reach the boundary in half of the time step.

2.4.2. Inlet/outlet conditions

At the inlet and/or outlet, macroscopic boundary conditions of the fluid motion, such as known pressure, known velocity, and known flow rate, have to be imposed. The macroscopic representation at the mesoscale involves a high number of degrees of freedom on the lattice

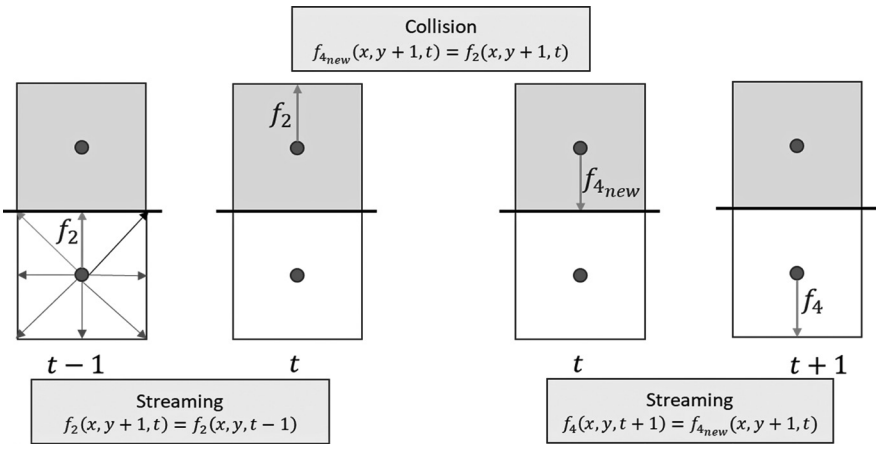

Fig. 4 – Fullway Bounce-Back boundary condition: the distribution functions are reflected with the same velocity during a time step.

nodes (i.e. distribution functions at specific locations of the lattices).

(Zou and He, 1997) proposed a method to compute the distribution functions from the known densities or velocities at inlets/outlets boundaries obtaining an approximation about the second order. Their method is based on the idea of bounce-back of the non-equilibrium part with some modifications. However, Zou and He conditions are difficult to implement for complex geometries because they are strongly dependent on the wall orientation (Zou and He, 1997). Equations (10) and (11) are combined to obtain a linear system for unknown f_i. The linear system is then completed (if necessary) taking into account the bounce-back for the non-equilibrium part (Meskas and Bao, 2014; Zou and He, 1997).

For example, for a known pressure condition, the density can be obtained by eq. (10). Considering the schematic example in Figure 6 the distributions f_4, f_7, and f_8 are unknown after streaming. From the mass and momentum conservation laws (eq. (10) and (11)) supposing that $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$ are specified they are used to recover missing populations and the density ρ:

$$\begin{aligned} &f_4 + f_7 + f_8 = \rho - (f_0 + f_1 + f_2 + f_3 + f_5 + f_6), \\ &f_8 - f_7 = \rho \boldsymbol{u}_x - (f_1 + f_3 + f_5 + f_6) \\ &f_4 + f_7 + f_8 = -\rho \boldsymbol{u}_y - (f_2 + f_5 + f_6) \\ &\Rightarrow \rho = \frac{f_0 + f_1 + f_3}{1 + \boldsymbol{u}_y} \end{aligned} \qquad (15)$$

In addition to (15) the configuration depicted in Figure 7 imposes that the bounce-back rule is correct for the non-equilibrium part

$$f_4 - f_4^{eq} = f_2 - f_2^{eq}, \qquad (16)$$

closing the linear unknowns system.

Other possible cases managed with (Zou and He, 1997) approach are:

– Given $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$, find ρ and unknown f_i
– Given ρ and the velocity along the boundary, find the velocity normal to the boundary and the unknown f_i.

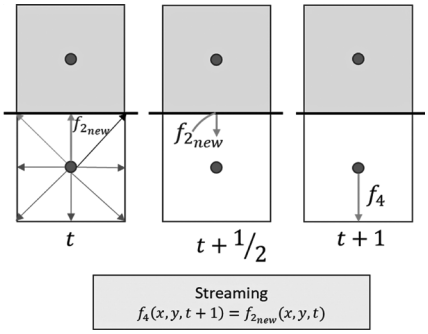

Fig. 5 – Halfway Bounce-Back condition. The updated streaming involves $\Delta t/2$, then is reflected in the new collision of the subsequent time step.

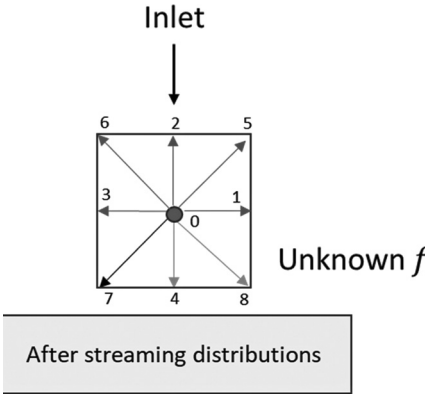

Fig. 6 – Zou-He velocity boundary conditions.

The known flow rate of a fluid injected into the sample is also a laboratory condition of interest. It can exploit velocity and pressure Zou-He conditions along a surface direction (Boek *et al.*, 2014):

$$q = -k/\mu(\nabla p - g\rho) \qquad (17)$$

q is the flow rate per unit area, μ the dynamic viscosity, related to the kinematic viscosity by $\mu = \nu\rho$, where ρ is the density. This formulation has been applied by (Boek *et al.*, 2014) comparing LB simulation to the experimental drainage process. In our perspective, this condition would be applied to confirm the laboratory experiments on microfluidic devices (Massimiani *et al.*, 2023).

2.5. LBM parallelization

LBM in HPC has been significantly popular in recent years when more performing architectures allow more complex geometries and detailed particle analysis. A parallel approach suits well to LBM because collision and streaming steps are independent in the single time step. In the beginning, The research on LBM was focused on improving the lattice model (Liu *et al.*, 2017) manipulating properly the time step relaxation in order to accelerate the local convergence.

Moreover, parallelization of LBM computation is possible in a multi-core approach through domain distribution: each core handles a sub-domain (i.e. a part of the domain) independently until the communication is necessary. With the MPI Forum Message Passing Interface (Forum Message Passing Interface., 1994) approach the node has no shared memory, so this kind of parallelization is subjected to temporary barriers when the cores are not balanced properly in terms of computation charge, i.e. a core finishes the computation waiting for the results from the others for a long time (Tesser, 2018). The optimization of data accesses in interprocess communication (Patil *et al.*, 2014) allows a massive parallelization of super-computers (Feichtinger *et al.*, 2015). Accelerators in PDE resolution, and LBM in particular (Liu *et al.*, 2017), provide a cache-optimization for direct resolutions on distributed domains passing through the so-called *ghost* regions to avoid the message passing lacks due to local memory. The ghost layers are data associated with neighbouring domains related to other cores (Balay *et al.*, 2023). However, enlarging the local memory including these regions avoids several interprocess communication. Moreover, due to the independent computation of collisions, that are local on each node, LBM suits well also for OpenMP

(Chandra, 2001) parallelism (Krause *et al.*, 2020; Latt *et al.*, 2021b), or GPU architectures (Kuznik *et al.*, 2010). This parallel nature of the method can be accelerated towards hybrid parallelization strategies (Heuveline *et al.*, 2009), that typically involve a greater number of cores distributing the independent computation in parallel performing shared memory.

2.6. LBM algorithm

The sketch of the algorithm for the single-phase LBM is given in Figure 6: Figure 7. After a pre-processing phase including mesh construction and non-dimensionalization, the lattice distribution functions are initialized to the equilibrium state given by the initial conditions of velocity and density (Liu *et al.*, 2023). Successive cycles of collision and streaming are performed until the equilibrium is reached; at the end of each cycle, boundary conditions are imposed and macroscopic quantities are updated. Results are dimensionalized in a post-processing phase.

3. The Multi-phase LBM algorithm

The multi-phase or multi-component flow simulation with LBM requires significant modifications. Different fluid phases interact with each other, generating a fluid-fluid interaction, and with porous rock, creating a fluid-solid interaction. These interactions (such as wettability) are taken into account as internal or external forces depending on the chosen method. We briefly describe the most common multi-phase method in this section (Huang *et al.*, 2015; Liu *et al.*, 2016). Other multi-phase LBMs have been then implemented: mixing the mentioned methods with CFD techniques (Huang *et al.*, 2015; Li and Wagner, 2007), adapting them to more specific cases (Lee and Liu, 2010), taking advantage of the finite-differences approach on binary fluids (Xu, 2005), accounting for chemical reactions in porous media flows (Liu *et al.*, 2023).

Multi-phase simulation allows us to perform contact angles and estimate capillary pressure-saturation behavior and absolute and relative permeability.

3.1. Color-gradient model

The first multi-phase LBM model here reported is the color-gradient model (Gunstensen *et al.*, 1991), popular in real rock geometry simulations due to its easy implementation and its strict immiscible phase separation (Chen *et al.*, 2019). The different colors (red and blue in general), labeled in the name of the method itself, are associated with different distribution functions. This method born for immiscible fluid has been modified to better simulate different densities and viscosities (Grunau *et al.*, 1993). Compared to the single-phase algorithm (Figure 7), the main difference occurs during the collision step, as presented in Figure 8 (left). In fact, the collision step is followed by a second collision that "re-colors" the interaction of the fluid with each other in order to reach the new equilibrium distribution before the streaming phase.

The total distribution function is defined by f_i^R (red) and f_i^B (blue) as:

$$f_i = f_i^R + f_i^B \tag{18}$$

Analogously, the density and the local fluid velocity are computed by the distribution functions above:

$$\rho = \rho^R + \rho^B, \rho^R = \sum_i f_i^R, \rho^B = \sum_i f_i^B, \quad \text{and} \quad \rho \boldsymbol{u} = \sum_i f_i^B \sum_i \sum_{B,R} f_i^{\{B,R\}} \boldsymbol{c}_i \tag{19}$$

The relaxation times for red and blue $\tau^{(R,\,B)}$ equations are independent, moreover the collision operators $\Omega^{\{R,\,B\}}$ are composed of three parts, the first $\Omega_i^{\{R,B\}^1}$ as a function of the equilibrium and the relaxation time of the concerned fluid, the second $\Omega_i^{\{R,B\}^2}$ which contribute to mixing interfaces between fluids, and the third $\Omega_i^{\{R,B\}^3}$ controlling the *recoloring* (Liu *et al.*, 2016) in function of a parameter $\Phi(\rho^{\{R,\,B\}})$. Then, each lattice node involves the collision operator:

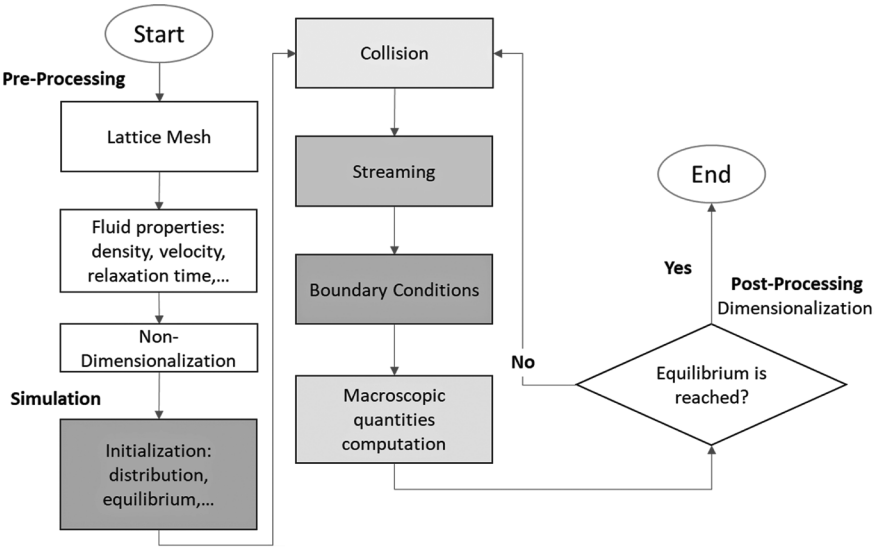

Fig. 7 – Single-phase LBM algorithm scheme. The different main phases, i.e. collision, and streaming, are colored to highlight the modifications of the algorithm moving to multi-phases simulations (Figure 8, Figure 9).

$$f_i^{\{R,B\}}\left(\boldsymbol{x}+\boldsymbol{c}_i\Delta t,t+\Delta t\right)= \\ =f_i^{\{R,B\}}\left(\boldsymbol{x},t\right)+\Omega_i^{\{R,B\}} \quad (20)$$

$$\Omega_i^{\{R,B\}}=\Omega_i^{\{R,B\}^3}\cdot\left(\Omega_i^{\{R,B\}^1}+\Omega_i^{\{R,B\}^2}\right),$$

where

$$\Omega_i^{\{R,B\}^1}=-\frac{1}{\tau^{(R,B)}}\left(f_i^{(R.B)}-f_i^{eq(R.B)}\right)$$

This method and its successors handle efficiently Multi Relaxation Time (MRT) implementations.

In (Gunstensen *et al.*, 1991) the interaction between two immiscible fluids in porous media is presented on randomly generated rock samples; the invading fluid simulation exploited without fluctuations confirms the LBM as a promising approach for multi-phase analysis. The color-gradient model is particularly efficient in the case of immiscible fluids flow, application given by (Chen *et al.*, 2019, 2018) where the method is applied to simulate the displacement of water by CO_2 injection in agreement with laboratory bubble test and capillary pressure rise test at reservoir pressure conditions in porous media. In (Chen *et al.*, 2019) for a *D*3*Q*19 model a parameter $\Phi\left(\rho^{\{R,B\}}\right)=\frac{\rho_R-\rho_B}{\rho_R+\rho_B}$ is chosen.

3.2. Shan-Chen model

The Shan-Chen model (Shan and Chen, 1993), SC in the following, is the most common one (and the most cited) among the different multi-phase methods (Huang *et al.*, 2015) due to the adaptivity both in single-component multi-phase (SC SCM) (Shan and Chen, 1994) model and multi-component multi-phase (MC SCM) (Shan and Doolen, 1995) model.

The main differences with respect to the single-phase algorithm are sketched and simplified in Figure 8 (right). In the multi-component Shan-Chen model, each component is described by its own distribution function f^{σ}, for each σ representing the number of components. Moreover, inter-particle interaction is included in the forcing term S_i presented in equation (7).

The Single Time Relaxation is applied to each component. The Shan-Chen force accounts for the attraction force calculated over the nearest neighbours of the pseudo-potential function Φ(ρ) depending on fluid densities and weighted by a coefficient controlling the strength of fluid attraction. Then, the equilibrium velocity of each fluid u_{σ}^{eq} takes into account the fluid-fluid cohesion forces F_{σ}^{f-f} and the fluid-solid cohesion forces F_{σ}^{f-s} (Huang *et al.*, 2015; Liu *et al.*, 2016) depending on Φ(ρ), such that:

$$\boldsymbol{u}_{\sigma}^{eq}=\boldsymbol{u'}+\frac{\tau_{\sigma}\boldsymbol{F}_{\sigma}}{\rho_{\sigma}} \quad (21)$$

Where $\boldsymbol{u'}=\sum_{\sigma}\frac{\rho_{\sigma}\boldsymbol{u}_{\sigma}}{\tau_{\sigma}}\Big/\sum_{\sigma}\frac{\rho_{\sigma}}{\tau_{\sigma}}$ is a common velocity, $F_{\sigma}=F_{\sigma}^{f-f}+F_{\sigma}^{f-s}+S$ the forces applied on the fluid σ, *S* external forces applied (i.e. gravity), $\rho_{\sigma}\left(\boldsymbol{x},t\right)=\sum_i f_i^{\sigma}\left(\boldsymbol{x},t\right)$ is the component density, and τ_{σ} the associated relaxation time. The total density is given by: $\rho(\boldsymbol{x},t)=\sum_{\sigma}\rho_{\sigma}(\boldsymbol{x},t)$.

Despite its popularity, the Shan-Chen model is affected by some limitations on density ratios among phases and inter-particle forces specification. However, it is simple and intuitive to compute (Huang *et al.*, 2015).

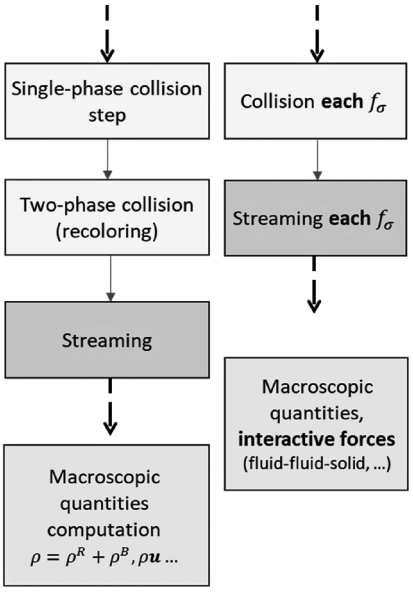

Fig. 8 – Color-gradient (left) and Shan-Chen multi-component (right) models' main modifications to the single-phase LBM flowchart scheme.

An example of fluid displacement in porous media with Shan-Chen approach is given in (Huang *et al.*, 2015).

3.3. He-Chen-Zhang (or interface tracking) model

The He-Chen-Zhang (HCZ) (He *et al.*, 1999) model has been proposed to simulate single-component multi-phase flow, it is also named the interface tracking model, and it presents two distribution functions and two corresponding LBEs. In (He *et al.*, 1999) the function of density is combined with new variables in order to simulate incompressible flows solving the Boltzmann equation; this distribution function *g* allows the computation of pressure. The two LBEs present also two relaxation times:

$$f_i\left(\boldsymbol{x}+\boldsymbol{c}_i\Delta t,t+\Delta t\right)-f_i\left(\boldsymbol{x},t\right)= \\ =-\frac{1}{\tau_f}\left(f_i\left(\boldsymbol{x},t\right)-f_i^{eq}\left(\boldsymbol{x},t\right)\right)+S_i\left(\boldsymbol{x},t\right) \\ g_i\left(\boldsymbol{x}+\boldsymbol{c}_i\Delta t,t+\Delta t\right)-g_i\left(\boldsymbol{x},t\right)= \\ =-\frac{1}{\tau_g}\left(g_i\left(\boldsymbol{x},t\right)-g_i^{eq}\left(\boldsymbol{x},t\right)\right)+S'_i\left(\boldsymbol{x},t\right) \quad (22)$$

τ_f related to the fluid mobility and τ_g related to the kinematic viscosity (Huang *et al.*, 2015). The model suits with the sharp interface constraint maintaining incompressible flows solving *f* as the distribution of the index function ϕ, i.e. $\phi=\sum_i f_i(\boldsymbol{x},t)$, and *g* as the distribution of the pressure separately. Each LBE associated to them involves, respectively in *S* and *S'*, an additional term resuming the surface tension and the intermolecular forces dependent of the pressure and the density by a function $\psi(\rho)=p-\rho RT$ (Huang *et al.*, 2015; Liu *et al.*, 2016) (see Figure

9, right). Given ϕ the remaining macroscopic variables are calculated through $p = \sum_i g_i - \frac{1}{2}\left(\boldsymbol{u}\nabla\psi\left(\rho\right)\Delta t\right)$, and $\rho\boldsymbol{u} = \frac{1}{c_s^2}\sum_i g_i \boldsymbol{c}_i + \frac{1}{2}\left(\boldsymbol{F_s}\Delta t\right)$, where $\boldsymbol{F}_s$ is the intermolecular force controlling the interfacial tension.

3.4. Free-energy model

The free-energy model, FE LBM, has been presented in (Swift *et al.*, 1996) for single and multi-component fluids flow applications. Here we briefly resume the multi-component approach represented by two independent densities, then, two independent sets of distribution functions f_i and g_i, each one computed according to a single relaxation time (SRT) with different $\tau_{f,g}$. Similarly to the color-grandient model, the total density is computed as a combination of both fluids densities. Moreover, a key point of this scheme is the relation with the macroscopic density and momentum conservation laws, built such that the difference in density between the two fluids is conserved following:

$$\Delta\rho = \rho_f - \rho_g \qquad (23)$$

In addition to (10) and (11) a new conservation law is introduced in any collision:

$$\Delta\rho = \sum_i g_i \qquad (24)$$

The three conservation laws are used in equilibrium computation for f_i^{eq} and g_i^{eq} respectively. Moreover, a specification of density difference amounts to a specification of pressure difference (Zou and He, 1997). The two distribution functions so affect both collision and streaming phases (applied twice independently), and the equilibrium requirement takes into account the density difference, as presented in Figure 9 (left) where the first equilibrium computation is not considered as a "modification" from single-phase.

3.4.1. Boundary conditions

The multi-component and multi-phase boundary and initial conditions are not trivial due to the pressure modifications that affect directly equation (12) introducing different densities and new parameters related to different physical constraints (such as gravity and miscibility) (Huang *et al.*, 2015; Ihle, 2009; Liu *et al.*, 2016; Wang *et al.*, 2022; Xu and Liu, 2018).

A generalization to the multi-phase flow of the Zou-He condition is possible. An example of the Zou-He condition applied to the Shan Chen model is given in (Huang *et al.*, 2015) for 2D and 3D simulated cases on which equations (16) are built for f_i^n and f_i^w representing the wetting and non-wetting phases respectively.

An example of a multi-reflection, or multi-bounce-back, condition is given in (Pan *et al.*, 2006).

4. Available Codes and Development

The LBM implementation is a common approach to model compressible and incompressible flows. The main drawback in LBM computation is the required uniform grid. On the one hand, this is a simplification in terms of finite difference discretization (Raeli *et al.*, 2018); on the other hand, it is hard to benefit in terms of memory and computation runtime. Some efforts to optimize computation have been made in terms of mesh (Kang *et al.*, 2002) approaching multiscale problems (Patil *et al.*, 2014) and superimposed meshes or parallel domain distribution (Latt *et al.*, 2021b; Patil *et al.*, 2014).

Moreover, being collision and streaming steps independent in the single time step, LBM is naturally conceived for a parallel approach. For this reason most of the open-source LB-based codes available focus on C++ (OpenMP/OpenCL/MPI) and Python languages. In Figure 10 a simplified list of GitHub open source codes responding to LBM keywords is sketched to give an idea of the programming languages' statistics; among the listed codes around 10% focus on porous media flow simulation. In this section, we limit to a brief outline of the most common available LBM codes with their possible advantages.

4.1. Common software and strategies

The first library here presented is Palabos, *pa*-rallel *la*-ttice *Bo*-ltzmann *s*-olver, a C++ open-source

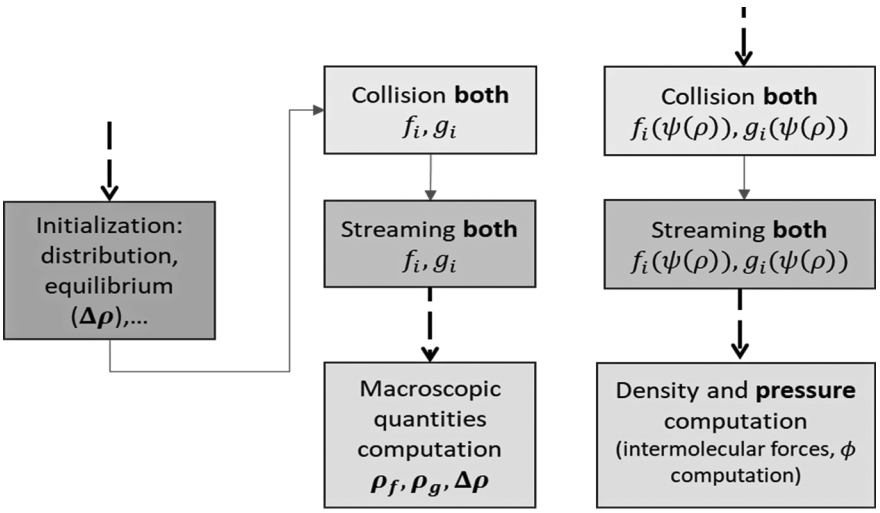

Fig. 9 – Free-energy (left) and interface tracking (right) models main modifications to single-phase LBM flowchart scheme.

Language	Public Repositories
C++	35
Python	17
MATLAB	9
C	6
Others (Julia, Fortran, Java, Cuda, ...)	25

Fig. 10 – Lattice Boltzmann Method libraries and software available on GitHub.

library developed at the University of Geneve (Latt *et al.*, 2021b). Palabos presents an MPI parallel approach based on domain distribution among processes that allows massive parallelization data handling for large-scale fluid dynamic computations (Mazzeo and Coveney, 2008). It also guarantees a flexible built-in data processor function allowing an efficient customization of the fields (scalars, tensors, physical) in parallel. Palabos developers list several joint works and codes based on the library on the website[1]. It boasts a rich list of references in other research, lots of them focusing on porous media single-phase fluid flow. Such as example, in order to study the diffusion coefficients of a porous media a work based on Palabos and simulating carbon saturated fiber for gas diffusion is presented in (García-Salaberri *et al.*, 2015).

MPLBM-UT is a Python software based on Palabos taking into consideration the multi-phase flow in porous media (Santos *et al.*, 2022). Python interface is user-friendly. MPLBM-UT performs the Shan-Chen model in two phases, BGK and MRT on single-phase simulation. LBM is also applied to advection-diffusion problems in various fields such as multi-phase CO_2 simulation (Ting *et al.*, 2022), and chemicals (Misaghian *et al.*, 2023). In this last framework (Liu *et al.*, 2023) the LBM has been applied for reactive transport coupled with a chemical reaction solver (PHREEQCRM): Palabos library has been used for LBM computation, PHREEQC solver handles the reactions with the porous medium, and the physical-chemical properties within the flow domain. The coupling method guarantees both a high parallel computational performance and data communication efficiency.

In 2021 on the Palabos forum page, STLBM code was announced: a full C++ parallel algorithm code (Latt *et al.*, 2021a) well performing on porous media LB simulations. It presents three different schemes and several multi-relaxation time operators.

The OpenLB project (Krause *et al.*, 2020) is a C++ package that can be easily extensible to different physical models. The first release of the code was published in 2007, the latest one in 2020. OpenLB is applied in miscellaneous fields from heat transfer in porous media to turbulences. The authors and developers of this software have recently explored the homogenized LBM for fluid flow through porous media (Simonis *et al.*, 2023). OpenLB was widely used in porous media flow simulation: in hydraulic fractures problems (Wang *et al.*, 2015), in multiple relaxation time for lithium displacement in tungsten porous structure (Kawashimo and Nieto-Pérez, 2023), in application to absolute permeability computation (Chukwudozie and Tyagi, 2013), in multi-phase water-air saturation based on the Shan-Chen model (Xu *et al.*, 2021), and in multi-component CO_2 injection in homogenized CH_4 saturated porous rocks (Wu *et al.*, 2023).

The MF-LBM code is a Fortran and MPI/OpenMP software for high-performance lattice Boltzmann (LB) applications on single and multi-phase flow in porous media. It also requires a (NVIDIA or Intel) Fortran compiler allowing higher FLOPS (operations per second) using GPU processors. Moreover, several optimizations in memory handling are implemented in this code taking advantage of the hybrid computing strategy. In (Chen *et al.*, 2018) the code was applied in a variation of the color-gradient model applied to simulate the water displacement due to CO_2 injection producing satisfactory trajectories with respect to the real geometry of the rock structure in reservoir conditions. In (Chen *et al.*, 2019) MF-LBM software has been applied, adopting a color-gradient MRT model, to simulate brine displacement in supercritical CO_2 conditions for a sandstone. In this context, the code was applied to real rock geometries and demonstrated 99% parallel scalability employing 20 NVIDIA V100 GPUs.

A simpler serial code totally in Fortran implementing LBM in porous media and handling multicomponent (Shan-Chen) simulations is ListLBM[2]. It can constitute a valuable starting point for compiling and running tests LBM before delving into more performing and parallel implementations.

Another software valid in the HPC context is LB3D (Schmieschek *et al.*, 2017), an MPI parallel Fortran90 implementation simulating miscible and non-miscible fluid mixtures, developed by the University College London. It implements the multicomponent Shan-Chen model (Liu *et al.*, 2016). It scaled linearly until 49,152 cores on a lat-

[1] Publications-Palabos-UNIGE.

[2] GitHub-sorush-khajepor/listLBM:ListLBMisasparselatticeBoltzmannsolverformultiphaseflowinporousmedia.

tice site of 1024 ×1024 ×1536 (Schmieschek *et al.*, 2017).

A free-energy model-based code, developed by the Univesity of Edinburgh for interacting colloidal particles, and also taking into account porous media, is the Ludwig implementation[3]; the latest release is available from January 2024.

Several other Python, MPI, and CUDA LBM software are available as (Łaniewski-Wołłk and Rokicki, 2016).

5. Conclusion

In this work we outlined the Lattice Boltzmann Method in single and multi-phase, we briefly introduced the discretization of the Lattice Boltzmann Equation, then, we focused on its application to porous media. The second part of this review presents the LBM scheme and its main modifications in multi-phase fluid flow simulation, describing the main popular methods for this application: the color gradient model, the Shan-Chen model, the He-Chen-Zhang (or interface tracking) model, and the free energy model. These are recalled in some software in the last section, in which we list the most popular and recent software available online for multi-phase flow simulations in porous media through LBM. Among them, OpenLB and Palabos are the reference software in this field for their easy adaptation.

This work aims to give a brief introduction to the LBM and related implementation and its applicability to multi-phase porous media flows. The mentioned software are not collaborating with this work and we are sure that the list is not complete, but we limited to the most popular ones.

[3] https://doi.org/10.5281/zenodo.12822477

List of Symbols

x, $\Delta \boldsymbol{x}$	d-dimensional spatial position/step
$\boldsymbol{u}$	Macroscopic velocity
t, Δt	Time, time/step
$f(\boldsymbol{x}, \boldsymbol{u}, t), f, f_i$	Particle distribution function
f^{eq}, f_i^{eq}	Equilibrium distribution function
$\Omega(f, f^{eq})$, Ω_i	Collision operator
$\boldsymbol{c}$	Microscopic velocity
$\boldsymbol{c_i}$	Lattice velocity vector
τ	Relaxation time
$S_i(\boldsymbol{x}, t)$	Source terms
ρ	Fluid density
ω_i	Method weights
c_s	Speed of sound in lattice space
k	Permeability
p	Pressure
μ	Dynamic viscosity
ν	Kinematic viscosity

References

Balay, S., Abhyankar, S., Adams, M., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M., Kong, F., Kruger, S., May, D., McInnes, L., Mills, R., Mitchell, L., Munson, T., Roman, J., Rupp, K., Sanan, P., Sarich, J., Smith, B., Zampini, S., Zhang, H., Zhang, J. (2023). PETSc/TAO Users Manual (Rev. 3.20) (No. ANL--21/39 Rev. 3-20, 2205494, 185712). https://doi.org/10.2172/2205494

Bhatnagar, P.L., Gross, E.P., Krook, M. (1954). A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. Phys. Rev. 94, 511-525. https://doi.org/10.1103/PhysRev.94.511

Blunt, M.J. (2001). Flow in porous media – pore-network models and multiphase flow. Current Opinion in Colloid & Interface Science 6, 197-207. https://doi.org/10.1016/S1359-0294(01)00084-X

Blunt, M.J., Bijeljic, B., Dong, H., Gharbi, O., Iglauer, S., Mostaghimi, P., Paluszny, A., Pentland, C. (2013). Pore-scale imaging and modelling. Advances in Water Resources 51, 197-216. https://doi.org/10.1016/j.advwatres.2012.03.003

Boek, E.S., Jianhui Yang, Chapman, E.M., Crawshaw, J.P. (2014). Displacement mechanisms in single and multiple pores from lattice boltzmann simulations and micro-fluidic experiments. https://doi.org/10.13140/2.1.1756.8000

Chandra, R. (Ed)., 2001. Parallel programming in OpenMP. Morgan Kaufmann Publishers, San Francisco, CA.

Chen, Y., Li, Y., Valocchi, A.J., Christensen, K.T. (2018). Lattice Boltzmann simulations of liquid CO2 displacing water in a 2D heterogeneous micromodel at reservoir pressure conditions. Journal of Contaminant Hydrology 212, 14-27. https://doi.org/10.1016/j.jconhyd.2017.09.005

Chen, Y., Valocchi, A.J., Kang, Q., Viswanathan, H.S. (2019). Inertial Effects During the Process of Supercritical CO_2 Displacing Brine in a Sandstone: Lattice Boltzmann Simulations Based on the Continuum-Surface-Force and Geometrical Wetting Models. Water Resources Research 55, 11144-11165. https://doi.org/10.1029/2019WR025746

Chukwudozie, C., Tyagi, M. (2013). Pore scale inertial flow simulations in 3-D smooth and rough sphere packs using lattice Boltzmann method. AIChE Journal 59, 4858-4870. https://doi.org/10.1002/aic.14232

Coco, A., Russo, G. (2018). Second order finite-difference ghost-point multigrid methods for elliptic problems with discontinuous coefficients on an arbitrary interface. Journal of Computational Physics 361, 299-330. https://doi.org/10.1016/j.jcp.2018.01.016

Dong, H., Blunt, M.J. (2009). Pore-network extraction from micro-computerized-tomography images. Phys. Rev. E 80, 036307. https://doi.org/10.1103/PhysRevE.80.036307

Eshghinejadfard, A., Daróczy, L., Janiga, G., Thévenin, D. (2016). Calculation of the permeability in porous media using the lattice Boltzmann method. International Journal of Heat and Fluid Flow 62, 93-103. https://doi.org/10.1016/j.ijheatfluidflow.2016.05.010

Feichtinger, C., Habich, J., Köstler, H., Rüde, U., Aoki, T. (2015). Performance modeling and analysis of heterogeneous lattice Boltzmann simulations on CPU–GPU clusters. Parallel Computing 46, 1-13. https://doi.org/10.1016/j.parco.2014.12.003

Ferréol, B., Rothman, D.H. (1995). Lattice-Boltzmann simulations of flow through Fontainebleau sandstone. Transp Porous Med 20, 3-20. https://doi.org/10.1007/BF00616923

Forum Message Passing Interface. (1994). MPI: A Message-Passing Interface Standard – Version 4.0. (Technical Report. University of Tennessee, Knoxville, TN).

Frisch, U., Hasslacher, B., Pomeau, Y. (1986). Lattice-Gas Automata for the Navier-Stokes Equation. Phys. Rev. Lett. 56, 1505-1508. https://doi.org/10.1103/PhysRevLett.56.1505

García-Salaberri, P.A., Gostick, J.T., Hwang, G., Weber, A.Z., Vera, M. (2015). Effective diffusivity in partially-saturated carbon-fiber gas diffusion layers: Effect of local saturation and application to macroscopic continuum models. Journal of Power Sources 296, 440-453. https://doi.org/10.1016/j.jpowsour.2015.07.034

Ghanbarian, B., Hunt, A.G., Ewing, R.P., Sahimi, M. (2013). Tortuosity in Porous Media: A Critical Review. Soil Science Society of America Journal 77, 1461-1477. https://doi.org/10.2136/sssaj2012.0435

Ghassemi, A., Pak, A. (2011). Pore scale study of permeability and tortuosity for flow through particulate media using Lattice Boltzmann method. Num Anal Meth Geomechanics 35, 886-901. https://doi.org/10.1002/nag.932

Grunau, D., Chen, S., Eggert, K. (1993). A lattice Boltzmann model for multiphase fluid flows. Physics of Fluids A: Fluid Dynamics 5, 2557-2562. https://doi.org/10.1063/1.858769

Gunstensen, A.K., Rothman, D.H., Zaleski, S., Zanetti, G. (1991). Lattice Boltzmann model of immiscible fluids. Phys. Rev. A 43, 4320-4327. https://doi.org/10.1103/PhysRevA.43.4320

Hardy, J., Pomeau, Y., De Pazzis, O. (1973). Time Evolution of a Two-Dimensional Classical Lattice System. Phys. Rev. Lett. 31, 276-279. https://doi.org/10.1103/PhysRevLett.31.276

He, X., Chen, S., Zhang, R. (1999). A Lattice Boltzmann Scheme for Incompressible Multiphase Flow and Its Application in Simulation of Rayleigh–Taylor Instability. Journal of Computational Physics 152, 642-663. https://doi.org/10.1006/jcph.1999.6257

He, X., Luo, L.-S., 1997a. Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation. Phys. Rev. E 56, 6811-6817. https://doi.org/10.1103/PhysRevE.56.6811

He, X., Luo, L.-S., 1997b. A priori derivation of the lattice Boltzmann equation. Phys. Rev. E 55, R6333–R6336. https://doi.org/10.1103/PhysRevE.55.R6333

Heuveline, V., Krause, M.J., Latt, J. (2009). Towards a hybrid parallelization of lattice Boltzmann methods. Computers & Mathematics with Applications 58, 1071-1080. https://doi.org/10.1016/j.camwa.2009.04.001

Huang, H., Sukop, M.C., Lu, X. (2015). Multiphase Lattice Boltzmann Methods: Theory and Application, 1st ed. Wiley. https://doi.org/10.1002/9781118971451

Huang, X., Zhou, W., Deng, D. (2021). Effective Diffusion in Fibrous Porous Media: A Comparison Study between Lattice Boltzmann and Pore Network Modeling Methods. Materials 14, 756. https://doi.org/10.3390/ma14040756

Ihle, T. (2009). Chapman–Enskog expansion for multi-particle collision models. Phys. Chem. Chem. Phys. 11, 9667. https://doi.org/10.1039/b910356b

Jadidi, M., Simmonds, M.J., Dadvand, A., Tansley, G. (2022). A systematic study for determining the optimal relaxation time in the lattice Boltzmann method at low Reynolds numbers. Alexandria Engineering Journal 61, 469-479. https://doi.org/10.1016/j.aej.2021.06.037

Jiang, M., Xu, Z.G., Zhou, Z.P. (2021). Pore-scale investigation on reactive flow in porous media considering dissolution and precipitation by LBM. Journal of Petroleum Science and Engineering 204, 108712. https://doi.org/10.1016/j.petrol.2021.108712

Junk, M., Klar, A. (2000). Discretizations for the Incompressible Navier--Stokes Equations Based on the Lattice Boltzmann Method. SIAM J. Sci. Comput. 22, 1-19. https://doi.org/10.1137/S1064827599357188

Kang, Q., Zhang, D., Chen, S. (2002). Unified lattice Boltzmann method for flow in multiscale porous media. Phys. Rev. E 66, 056307. https://doi.org/10.1103/PhysRevE.66.056307

Kawashimo, K., Nieto-Pérez, M. (2023). Pore Resolved Lattice Boltzmann Simulation of Liquid Lithium Flow in Porous Tungsten Structures 2023, CP11.089.

Koponen, A., Kataja, M., Timonen, J. (1997). Permeability and effective porosity of porous media. Phys. Rev. E 56, 3319-3325. https://doi.org/10.1103/PhysRevE.56.3319

Koponen, A., Kataja, M., Timonen, J. (1996). Tortuous flow in porous media. Phys. Rev. E 54, 406-410. https://doi.org/10.1103/PhysRevE.54.406

Krause, M.J., Avis, S., Kusumaatmaja, H., Dapelo, D., Gaedtke, M., Hafen, N., Haußmann, M., Jeppener-Haltenhoff, J., Kronberg, L., Kummerländer, A., Marquardt, J.E., Pertzel, T., Simonis, S., Trunk, R., Wu, M., Zarth, A. (2020). OpenLB Release 1.4: Open Source Lattice Boltzmann Code. https://doi.org/10.5281/ZENODO.4279263

Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., Viggen, E.M. (2017). The Lattice Boltzmann

Method: Principles and Practice, Graduate Texts in Physics. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-44649-3

Kuznik, F., Obrecht, C., Rusaouen, G., Roux, J.-J. (2010). LBM based flow simulation using GPU computing processor. Computers & Mathematics with Applications 59, 2380-2392. https://doi.org/10.1016/j.camwa.2009.08.052

Lallemand, P., Luo, L.-S. (2000). Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. Phys. Rev. E 61, 6546-6562. https://doi.org/10.1103/PhysRevE.61.6546

Łaniewski-Wołłk, Ł., Rokicki, J. (2016). Adjoint Lattice Boltzmann for topology optimization on multi-GPU architecture. Computers & Mathematics with Applications 71, 833-848. https://doi.org/10.1016/j.camwa.2015.12.043

Latt, J., Coreixas, C., Beny, J., 2021a. Cross-platform programming model for many-core lattice Boltzmann simulations. PLoS ONE 16, e0250306. https://doi.org/10.1371/journal.pone.0250306

Latt, J., Malaspinas, O., Kontaxakis, D., Parmigiani, A., Lagrava, D., Brogi, F., Belgacem, M.B., Thorimbert, Y., Leclaire, S., Li, S., Marson, F., Lemus, J., Kotsalos, C., Conradin, R., Coreixas, C., Petkantchin, R., Raynaud, F., Beny, J., Chopard, B., 2021b. Palabos: Parallel Lattice Boltzmann Solver. Computers & Mathematics with Applications 81, 334-350. https://doi.org/10.1016/j.camwa.2020.03.022

Lee, T., Liu, L. (2010). Lattice Boltzmann simulations of micron-scale drop impact on dry surfaces. Journal of Computational Physics 229, 8045-8063. https://doi.org/10.1016/j.jcp.2010.07.007

Li, Q., Wagner, A.J. (2007). Symmetric free-energy-based multicomponent lattice Boltzmann method. Phys. Rev. E 76, 036701. https://doi.org/10.1103/PhysRevE.76.036701

Liu, H., Kang, Q., Leonardi, C.R., Schmieschek, S., Narváez, A., Jones, B.D., Williams, J.R., Valocchi, A.J., Harting, J. (2016). Multiphase lattice Boltzmann simulations for porous media applications: A review. Comput Geosci 20, 777-805. https://doi.org/10.1007/s10596-015-9542-3

Liu, S., Barati, R., Zhang, C., Kazemi, M. (2023). Coupled Lattice Boltzmann Modeling Framework for Pore-Scale Fluid Flow and Reactive Transport. ACS Omega 8, 13649-13669. https://doi.org/10.1021/acsomega.2c07643

Liu, S., Zou, N., Cui, Y., Wu, W. (2017). Accelerating the Parallelization of Lattice Boltzmann Method by Exploiting the Temporal Locality, in: 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC). Presented at the 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), IEEE, Guangzhou, pp. 1186-1193. https://doi.org/10.1109/ISPA/IUCC.2017.00178

Liu, Y. (2020). Maximizing the CFL number of stable time–space domain explicit finite-difference modeling. Journal of Computational Physics 416, 109501. https://doi.org/10.1016/j.jcp.2020.109501

Massimiani, A., Panini, F., Marasso, S.L., Vasile, N., Quaglio, M., Coti, C., Barbieri, D., Verga, F., Pirri, C.F., Viberti, D. (2023). Design, Fabrication, and Experimental Validation of Microfluidic Devices for the Investigation of Pore-Scale Phenomena in Underground Gas Storage Systems. Micromachines 14, 308. https://doi.org/10.3390/mi14020308

Mazzeo, M.D., Coveney, P.V. (2008). HemeLB: A high performance parallel lattice-Boltzmann code for large scale fluid flow in complex geometries. Computer Physics Communications 178, 894-914. https://doi.org/10.1016/j.cpc.2008.02.013

McNamara, G.R., Zanetti, G. (1988). Use of the Boltzmann Equation to Simulate Lattice-Gas Automata. Phys. Rev. Lett. 61, 2332-2335. https://doi.org/10.1103/PhysRevLett.61.2332

Meskas, J., Bao, Y. (2014). Lattice Boltzmann Method for Fluid Simulations.

Min, C., Gibou, F., Ceniceros, H.D. (2006). A supra-convergent finite difference scheme for the variable coefficient Poisson equation on non-graded grids. Journal of Computational Physics 218, 123-140. https://doi.org/10.1016/j.jcp.2006.01.046

Misaghian, N., Sadeghi, M.A., Lee, K.M., Roberts, E.P.L., Gostick, J.T. (2023). Utilizing Pore Network Modeling for Performance Analysis of Multi-Layer Electrodes in Vanadium Redox Flow Batteries. J. Electrochem. Soc. 170, 070520. https://doi.org/10.1149/1945-7111/ace554

Pan, C., Luo, L.-S., Miller, C.T. (2006). An evaluation of lattice Boltzmann schemes for porous medium flow simulation. Computers & Fluids 35, 898-909. https://doi.org/10.1016/j.compfluid.2005.03.008

Panini, F. (2022). Pore-scale characterization of rock images: geometrical analysis and hydrodynamic simulation. PhD dissertation. Politecnico di Torino.

Panini, F., Ghanbarian, B., Salina Borello, E., Viberti, D. (2024). Estimating geometric tortuosity of saturated rocks from micro-CT images using percolation theory. Transp Porous Med. https://doi.org/10.1007/s11242-024-02085-w

Patil, D.V., Premnath, K.N., Banerjee, S. (2014). Multigrid lattice Boltzmann method for accelerated solution of elliptic equations. Journal of Computational Physics 265, 172-194. https://doi.org/10.1016/j.jcp.2014.01.049

Pellerin, N., Leclaire, S., Reggio, M. (2017). Solving incompressible fluid flows on unstructured meshes with the lattice Boltzmann flux solver. Engineering Applications of Computational Fluid Mechanics 11, 310-327. https://doi.org/10.1080/19942060.2017.1292410

Raeli, A. (2017). Solution Of The Variable Coefficient Poisson Equation On Cartesian Hierarchical Meshes In Parallel: Applications To Phase Changing

Materials (PhD Thesis). MB – Institut de Mathématiques de Bordeaux.

Raeli, A., Bergmann, M., Iollo, A. (2018). A finite-difference method for the variable coefficient Poisson equation on hierarchical Cartesian meshes. Journal of Computational Physics 355, 59-77. https://doi.org/10.1016/j.jcp.2017.11.007

Raeli, A., Salina Borello, E., Panini, F., Serazio, C., Viberti, D. (2024). A parallel programming application of the A* algorithm in digital rock physics. Computers & Geosciences 105578. https://doi.org/10.1016/j.cageo.2024.105578

Salina Borello, E., Peter, C., Panini, F., Viberti, D. (2022). Application of A* algorithm for microstructure and transport properties characterization from 3D rock images. Energy 239, 122151. https://doi.org/10.1016/j.energy.2021.122151

Santos, J.E., Gigliotti, A., Bihani, A., Landry, C., Hesse, M.A., Pyrcz, M.J., Prodanović, M. (2022). MPLBM-UT: Multiphase LBM library for permeable media analysis. SoftwareX 18, 101097. https://doi.org/10.1016/j.softx.2022.101097

Schmieschek, S., Shamardin, L., Frijters, S., Krüger, T., Schiller, U.D., Harting, J., Coveney, P.V. (2017). LB3D: A parallel implementation of the Lattice-Boltzmann method for simulation of interacting amphiphilic fluids. Computer Physics Communications 217, 149-161. https://doi.org/10.1016/j.cpc.2017.03.013

Shan, X., Chen, H. (1994). Simulation of nonideal gases and liquid-gas phase transitions by the lattice Boltzmann equation. Phys. Rev. E 49, 2941-2948. https://doi.org/10.1103/PhysRevE.49.2941

Shan, X., Chen, H. (1993). Lattice Boltzmann model for simulating flows with multiple phases and components. Phys. Rev. E 47, 1815-1819. https://doi.org/10.1103/PhysRevE.47.1815

Shan, X., Doolen, G. (1995). Multicomponent lattice-Boltzmann model with interparticle interaction. J Stat Phys 81, 379-393. https://doi.org/10.1007/BF02179985

Simonis, S., Hafen, N., Jeßberger, J., Dapelo, D., Thäter, G., Krause, M.J. (2023). Homogenized lattice Boltzmann methods for fluid flow through porous media -- part I: kinetic model derivation. https://doi.org/10.48550/ARXIV.2310.14746

Song, S., Wang, S., Le-Clech, P., Shen, Y. (2022). LBM-DEM simulation of particle deposition and resuspension of pre-deposited dynamic membrane. Powder Technology 407, 117637. https://doi.org/10.1016/j.powtec.2022.117637

Succi, S. (2001). The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford University PressOxford. https://doi.org/10.1093/oso/9780198503989.001.0001

Sukop, M.C., Thorne, D.T. (2006). Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-27982-2

Swift, M.R., Orlandini, E., Osborn, W.R., Yeomans, J.M. (1996). Lattice Boltzmann simulations of liquid-gas and binary fluid systems. Phys. Rev. E 54, 5041-5052. https://doi.org/10.1103/PhysRevE.54.5041

Tesser, F. (2018). Parallel solver for the Poisson equation on a hierarchy of superimposed meshes, under a Python framework. Universitè de Bordeaux.

Ting, A.K., Santos, J.E., Guiltinan, E. (2022). Using Machine Learning to Predict Multiphase Flow through Complex Fractures. Energies 15, 8871. https://doi.org/10.3390/en15238871

Venturoli, M., Boek, E.S. (2006). Two-dimensional lattice-Boltzmann simulations of single phase flow in a pseudo two-dimensional micromodel. Physica A: Statistical Mechanics and its Applications 362, 23-29. https://doi.org/10.1016/j.physa.2005.09.006

Viberti, D., Peter, C., Salina Borello, E., Panini, F. (2020). Pore structure characterization through path-finding and Lattice Boltzmann simulation. Advances in Water Resources 141, 103609. https://doi.org/10.1016/j.advwatres.2020.103609

Wang, Y., Wang, S., Xue, S., Adhikary, D. (2015). Numerical modeling of porous flow in fractured rock and its applications in geothermal energy extraction. J. Earth Sci. 26, 20-27. https://doi.org/10.1007/s12583-015-0507-1

Wang, Z., Soomro, M., Peng, C., Ayala, L.F., Ayala, O.M. (2022). Two pressure boundary conditions for multi-component multiphase flow simulations using the pseudo-potential lattice Boltzmann model. Computers & Fluids 248, 105672. https://doi.org/10.1016/j.compfluid.2022.105672

Wu, J., Gan, Y., Shi, Z., Huang, P., Shen, L. (2023). Pore-scale lattice Boltzmann simulation of CO2-CH4 displacement in shale matrix. Energy 278, 127991. https://doi.org/10.1016/j.energy.2023.127991

Xu, A. (2005). Finite-difference lattice-Boltzmann methods for binary fluids. Phys. Rev. E 71, 066706. https://doi.org/10.1103/PhysRevE.71.066706

Xu, F., Liang, S., Zhang, Y., Li, B., Hu, Y. (2021). Numerical study of water–air distribution in unsaturated soil by using lattice Boltzmann method. Computers & Mathematics with Applications 81, 573-587. https://doi.org/10.1016/j.camwa.2019.08.013

Xu, L. (2014). A novel moving boundary condition based on Chapman-Enskog expansion with the lattice Boltzmann method. (PhD Thesis. University of Pittsburgh)..

Xu, M., Liu, H. (2018). Prediction of immiscible two-phase flow properties in a two-dimensional Berea sandstone using the pore-scale lattice Boltzmann simulation. Eur. Phys. J. E 41, 124. https://doi.org/10.1140/epje/i2018-11735-3

Yang, P., Wen, Z., Dou, R., Liu, X. (2017). Permeability in multi-sized structures of random packed porous media using three-dimensional lattice Boltzmann method. International Journal of Heat and Mass Transfer 106, 1368-1375. https://doi.org/10.1016/j.ijheatmasstransfer.2016.10.124

Yin, X., Zhang, J. (2012). An improved bounce-back scheme for complex boundary conditions in latti-

ce Boltzmann method. Journal of Computational Physics 231, 4295-4303. https://doi.org/10.1016/j.jcp.2012.02.014

Zahedi, H., Vakili, M. (2023). The Computational Study of Fluid Diffusion through Complex Porous Media in the Presence of Gravitational Force and at Different Temperatures Using Image Processing Technique and D3Q27 Model of Lattice Boltzmann Method. Iran J Sci Technol Trans Mech Eng 47, 1553-1570. https://doi.org/10.1007/s40997-023-00619-z

Zhang, Y., Yang, Z., Wang, F., Zhang, X. (2021). Comparison of soil tortuosity calculated by different methods. Geoderma 402, 115358. https://doi.org/10.1016/j.geoderma.2021.115358

Zou, Q., He, X. (1997). On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. Physics of Fluids 9, 1591-1598. https://doi.org/10.1063/1.869307

Acknowledgment

Funded by the European Union. Views and opinions expressed within the document are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union or the Commission can be held responsible for them.