Multiphase flow in a porous medium using Lattice Boltzmann Method and grid verification

Assetbek Ashirbekov, B.Eng.

Submitted in fulfillment of the requirements
for the degree of Master of Science
in Mechanical & Aerospace Engineering

NAZARBAYEV UNIVERSITY

School of Engineering and Digital Sciences
Department of Mechanical & Aerospace Engineering
Nazarbayev University

53 Kabanbay Batyr Avenue,
Nur-Sultan city, Kazakhstan, 010000

Supervisor: Prof. Luis R. Rojas-Solórzano
Co-Supervisor: Prof. Konstantinos Kostas

April 2022

DECLARATION

I hereby declare that this manuscript, entitled "Multiphase flow in a porous medium using Lattice Boltzmann Method and grid verification," results from my work except for quotations and citations, which have been duly acknowledged.

I also declare that, to the best of my knowledge and belief, it has not been previously or concurrently submitted, in whole or in part, for any other degree or diploma at Nazarbayev University or any other national or international institution.

Name: Assetbek Ashirbekov

Date: 26.04.2022

Abstract

Lattice Boltzmann Method (LBM) of computational fluid dynamics is well suited for small-scale modeling, operating at mesoscale. LBM with a pseudopotential model and pressure equation of state is used to develop a model of the flow of CO_2 immiscibly displacing water in a porous medium simulating the fundamental problem of CO_2 sequestration. Validation of the model numerical stability is performed with droplet tests. The possibility of a nonunity viscosity ratio is explored, which was found possible, but to an extend limited by increased miscibility. Grid size verification procedure is developed for the model, for the first time in literature.

Acknowledgements

My deepest gratitude to Professor Luis Rojas-Solórzano, without whom my work would not be possible. Luis is not only my supervisor, but my mentor and one of the kindest people I have the privilege to know, with endless support not only towards my achievements, but also during my failures. Thank you for your guidance at the end of my BSc, during my whole MSc degree and beyond.

I want to thank Daniyar, my closest friend with whom I shared my last 7 years in academia and outside of it, and with whom I shared ideas and plans, stresses and achievements among oh so many other things. Thank you for being a friend I hope I deserve.

And thank you, Nazym, for making me happy. The only way to repay I have is to make you as happy as you make me.

Table of Contents

List of Tables

List of Figures

Chapter 1 - Introduction

Numerous physical systems and phenomena are too complex to solve using theoretical analysis or experimental setup. A suitable and powerful approach to solving such problems is computer modeling. Computational Fluid Dynamics (CFD) is a collection of computer modeling techniques used to analyze systems involving fluids and their flow. CFD tools are widely applicable in modeling complex systems and behaviors, yet there are many ways to develop and improve existing methods due to the nontriviality of problems considered. One of the frequently occurring complexities is the mesoscale behavior of the fluid flow, where flow behaves differently at the macro and microscale. Flow in critical energy systems like power plants, fuel cells, soil flow, oil recovery processes, and CO_2 sequestration are governed by this flow behavior.

Finite volume, the volume of fluid, and level-set methods are well-known traditional CFD methods. These methods consider that the fluid is continuous and are usually applied to model the flow on a macroscale, where this assumption holds. However, on the microscale, fluid flows behave differently and are not accurately described by these techniques as on the microscale, the fluid flow behaves more like a collection of discrete particles, defined by the movement of molecules, atoms, and their clusters. Lattice Boltzmann Method (LBM) is a powerful CFD approach that can model the flows at both scales. LBM is modeling fluid as discrete particles moving on the grid, working both microscale parameters and particles interactions while yielding results of macroscale quantities like pressure and velocity fields. Additionally, interface tracking for multi-component flows is computationally expensive in traditional CFD. In LBM, interface tracking does not require additional computational power, as it is preserved automatically by particle species, making it favorable for multi-component flows.

One of the complex systems that this work focuses on is the flow in a porous medium, which is often encountered in energy systems and underground fluid flows. Flow in a fine porous medium is strongly dependent on the interactions between discrete particles and small-scale obstacles. These interactions are well captured by LBM, allowing it to describe and predict the flow behavior accurately. One particular example of mesoscale flow occurs in the sequestration process of CO_2 into the subsurface aquifers. In this process, CO_2 flows into the porous matrix saturated with water, immiscibly displacing it. Atykhan et al. [1] developed the LBM model describing the CO_2 sequestration process; however, their model lacks the critical

state of CO_2 fluid, one of the defining characteristics of the given process. Due to these critical and near-critical states, the actual fluid behavior is expected to undergo significant changes. Kabdenova et al. [2] addressed this issue by developing a crossover enhancement of the LBM formulation. This enhanced model allows describing the fluid properties in critical states more accurately, capturing more physics. Current work aims at developing porous medium immiscible displacement using pressure EoS of LBM formulation to prepare the background work for crossover PR EoS. A significant part of the present work was also published in [3].

The main goal of this work is to model the CO_2-to-water displacement flow in a porous medium with the use of pressure equation of state (EoS) crossover enhancement. To achieve this goal, the following objectives are set:

- Introducing the multi-component LBM formulation.
- Simple analytical case of a water droplet surrounded by CO_2 will be analyzed and used for validation.
- Development of CO_2 front penetration into a medium of uniform obstacle grid saturated with water.
- New grid size scaling procedure will be developed and used to perform domain discretization verification.

Chapter 2 – Methodology

2.1. Lattice Boltzmann Method

For multiphase flow, the interfaces modeled with conventional CFD are computationally expensive. The computational cost rises even further when the porous structure is modeled, adding a solid sub-domain phase into consideration. In LBM, however, the interface between the solid and multiphase fluid phases is conserved, and complex geometries are considered part of the grid, reducing the computational cost.

A standard discretized two-dimensional Lattice Boltzmann Method with the Bhatnagar-Gross-Krook (BGK) collision operator [4] will be utilized as a baseline method.

Particles' states are described using a distribution function as given in Equation (1), where lattice units are assigned for parameters

$$f_i{}^j(\mathbf{x} + \mathbf{e_i}\Delta t, t + \Delta t) - f_i{}^j(\mathbf{x}, t) = -\frac{1}{\tau^j}\left(f_i{}^j(\mathbf{x}, t) - f_i^{j,eq}(\mathbf{x}, t)\right) \quad (1)$$

In Eq. (1), the LHS is the streaming, calculating the advancement of the model between timesteps, and the RHS is the collision step from the Boltzmann equation, derived from the kinetic theory of gases. $f_i{}^j(\mathbf{x}, t)$ is the density distribution function, τ is a relaxation time, i is the direction of movement, j describes phase, and **x** is a grid point occupied by a fluid particle at timestep t. Equation (2) describes the relationship between relaxation time and kinematic viscosity v [5]:

$$v = c_s^2 \sum_j \chi^j(\tau^j - 0.5), \quad (2)$$

where c_s is the lattice sound speed and χ is mass fraction.

Equation (3) gives $f_i^{eq}(\mathbf{x}, t)$ of Eq. (1), which is the equilibrium distribution function.

$$f_i^{j,eq}(\mathbf{x}, t) = \omega_i \rho^j \left[1 + \frac{\mathbf{e_i} \cdot \mathbf{u}^j}{c_s^2} + \frac{(\mathbf{e_i} \cdot \mathbf{u}^j)^2}{2c_s^4} - \frac{(\mathbf{u}^j)^2}{2c_s^2}\right], \quad (3)$$

where ω_i are the weight factors for D2Q9 model. The two-dimensional D2Q9 model with a 9-speed lattice is shown in Figure 1.

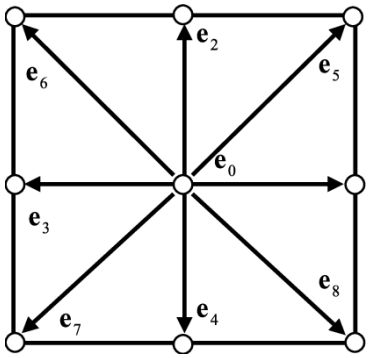


Figure 1: D2Q9, two-dimensional nine-directional discrete velocity model [6]

Lattice Boltzmann method uses dimensionless values for calculations, expressed in Lattice Units (LU). One of the ways to obtain macroscopic values is to calculate moments of the particle distribution f_i:

$$\rho^j = \sum_i f_i{}^j, \tag{4}$$

$$\rho^j \mathbf{u}^j = \sum_i f_i{}^j \, \mathbf{e_i}, \tag{5}$$

where **u** is the macroscopic velocity vector.

2.2. Pseudopotential Lattice Boltzmann model

In the pseudopotential LB model, any interactions between the particles are calculated through interaction potentials, which serve as a density function for each component. For pseudopotential model we define $G(\mathbf{x},\acute{\mathbf{x}})$ - a function that controls the interaction intensity between the particles at grid locations **x** and $\acute{\mathbf{x}}$, ψ is a pseudopotential ("effective mass"), and then the interaction acting on component *j* can be expressed as

$$\mathbf{F}_{int}{}^{j}(\mathbf{x},t) = -G(\mathbf{x},\acute{\mathbf{x}})\,\psi^{j}(\mathbf{x},t)\sum_{i}\omega_i\psi^{j}(\mathbf{x}+\mathbf{e_i}\Delta t,t)\,\mathbf{e_i} \tag{6}$$

Eq. (6) yields interaction force, it can be applied on the nearest neighboring particles. This formulation uses the following pressure EoS:

$$P^{j} = \rho^{j}c_s^2 + \frac{g}{2}c_s^2{\psi^{j}}^2, \tag{7}$$

where the $\rho^{j}c_s^2$ term is the ideal gas EoS, and the $\frac{g}{2}c_s^2{\psi^{j}}^2$ term accounts for repulsive and attractive forces between the fluid particles. Conveniently, for a two-component system that will be considered in this study, Green's function $G(\mathbf{x},\acute{\mathbf{x}})$ for interaction between components is symmetric and reduces to a single parameter g. Only the interactions between nearest neighboring particles are considered in this model, as shown in Equation (8).

$$G(\mathbf{x},\acute{\mathbf{x}}) = \begin{cases} g, |\mathbf{x}-\acute{\mathbf{x}}| \le c, \\ 0, |\mathbf{x}-\acute{\mathbf{x}}| > c, \end{cases} \tag{8}$$

where c is the uniform lattice spacing distance, [7], and g is components' interaction parameter. Pseudopotential definition can be obtained by rearranging Eq. (7):

$$\psi^{j}(\rho^{j}) = \sqrt{\frac{2(P^{j}-\rho^{j}c_s^2)}{gc_s^2}} \tag{9}$$

Yuan and Schaefer incorporated different EoS such as Van der Waals, Peng-Robinson, and Carnahan-Starling into the pseudopotential model through Equation (9) [8].

Interactions of fluid phases with the solid phase, obstacles, and walls, are calculated in a similar fashion using $\mathbf{F}_{wet}$. The surface interaction, i.e. wettability, is modeled using method proposed by Martys and Chen [9], modified for the multi-component form:

$$\mathbf{F}_{wet}{}^{j}(\mathbf{x}) = -g_{wall}{}^{j}\rho^{j}(\mathbf{x})\sum_{i}\omega_i s(\mathbf{x}+\mathbf{e_i})\mathbf{e_i} \tag{10}$$

The g_{wall} is similar to the components' interaction parameter but represents the fluid-wall interaction. With positive values of g_{wall}, the wettability reduces, with $\mathbf{F}_{wet}$ gaining negative values and causing fluid to repel from solid, and negative values of g_{wall} cause wettability increase. $s(\mathbf{x}+\mathbf{e_i})$ is the "switch" function: it takes value of 1 whenever a lattice node **x** is occupied by a solid phase; otherwise, it is 0 [10].

After all interactions are calculated, and the total force acting on a fluid particle can be summarized:

$$\mathbf{F}_{tot}{}^{j} = \mathbf{F}_{wet}{}^{j} + \mathbf{F}_{int}{}^{j} + \mathbf{F}_{b}, \tag{11}$$

where $\mathbf{F}_b$ is some external body force, which can have some conditional or constant value if applied to excite the flow, like modeling the effect of gravity. It can also be given value of 0 if no external body force is modeled.

To apply $\mathbf{F}_{tot}{}^{j}$ to the main function, the velocity shift force scheme was adopted, proposed by Shan and Chen [11]. In velocity shift scheme forces are applied to equilibrium distribution function by updating velocity $\mathbf{u}$, replacing it with $\mathbf{u}_{eq}$ in Eq. (3):

$$\boldsymbol{u}_{eq}{}^{j} = \mathbf{u}^{j} + \frac{\tau^{j}}{\rho^{j}}\mathbf{F}_{tot}{}^{j}\Delta t \tag{12}$$

In this study, the standard definition of LBM will be developed further using pressure EoS proposed in [2] and applied to the porous media geometry [1].

2.3. Grid size scaling

The use of LBM in multiphase flow in porous media is a relatively new direction in the CFD field. In the LBM application used, scaling does not work correctly, as domain size change does not scale the geometry but only the domain creating empty spaces, as shown in Figure 2. Thus, there is a lack of studies to perform the proper domain verification. A new domain verification procedure will be proposed in this study to address this issue.

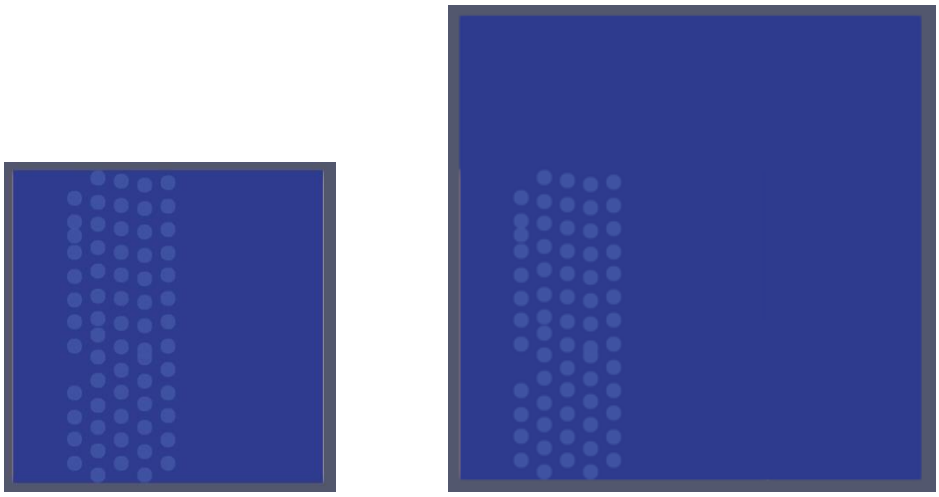


Figure 2: Simple increase of the model grid size from 400x400 (left) to 600x600 (right). Fluid phase in deep blue, solid obstacles in light blue, domain boundary in gray.

Multiplication of grid size cannot solve the issue with scaling, as it cannot change the total amount of occupied grid spaces and creates gaps, as shown in Figure 3.

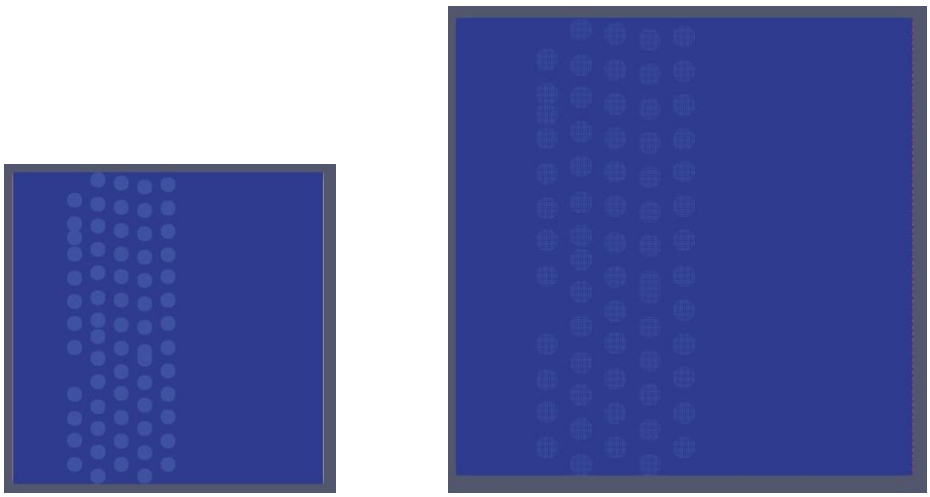


Figure 3: Simple coordinates scaling of the model grid size from 400x400 (left) to 600x600 (right)

Image scaling methods are used to address the issue with scaling in the LBM model. This method uses a pixelated image of a domain, with one pixel corresponding to one grid element and different colors assigned for solids and fluid phases to differentiate between them. Up and down-scaling of the domain is performed using image scaling techniques, as bilinear scaling is shown in Figure 4. Aside from bilinear scaling, nearest neighbor or bicubic scaling techniques can be applied to achieve the same result, depending on the geometry analyzed. This method is applied so scale boundary conditions, geometry, and initialization pattern.

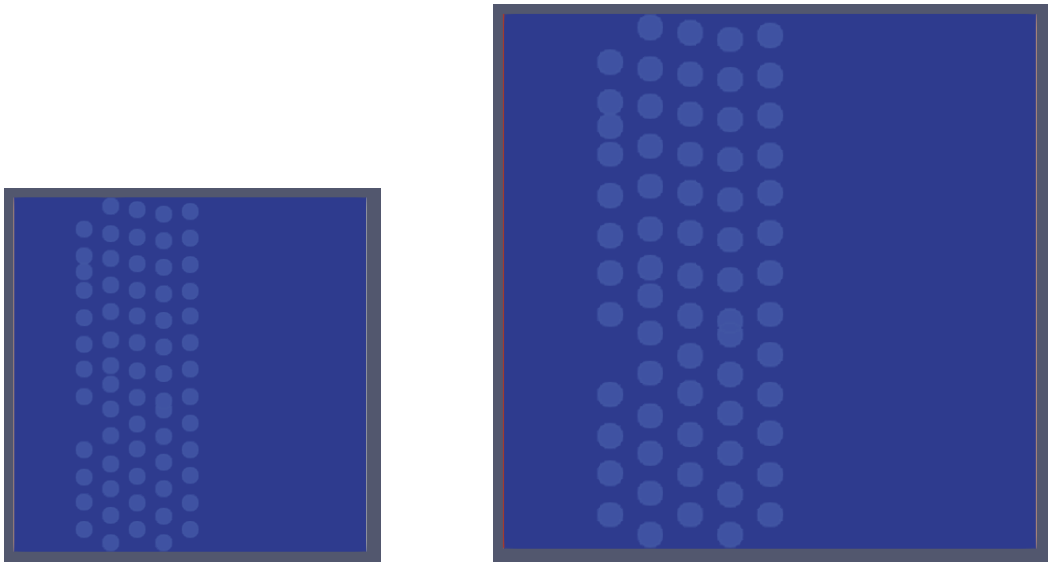


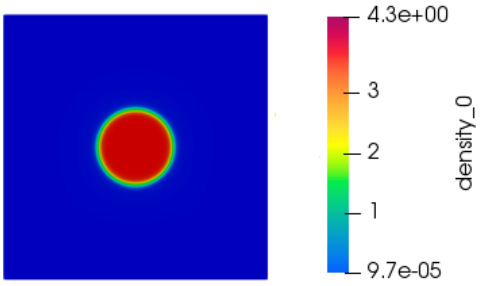
Figure 4: Proper scaling of the model grid size from 400x400 (left) to 600x600 (right) using bilinear scaling

This scaling can be used to perform domain size, or discretization verification. This must be applied carefully, with corresponding changes to boundary conditions, like velocity and pressure, and fluid parameters, like density and viscosity, to keep dimensionless numbers the same between models. All dimensionless numbers usually cannot all be preserved, and the most important for the model should be chosen and preserved. In this study, due to importance of capillary number in porous medium flow, it is chosen for preservation, with changes to inlet velocity and viscosity.

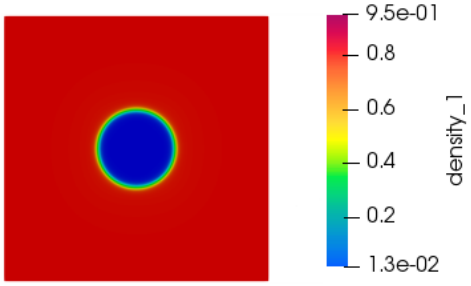
Chapter 3 – Validation tests

3.1. Droplet test

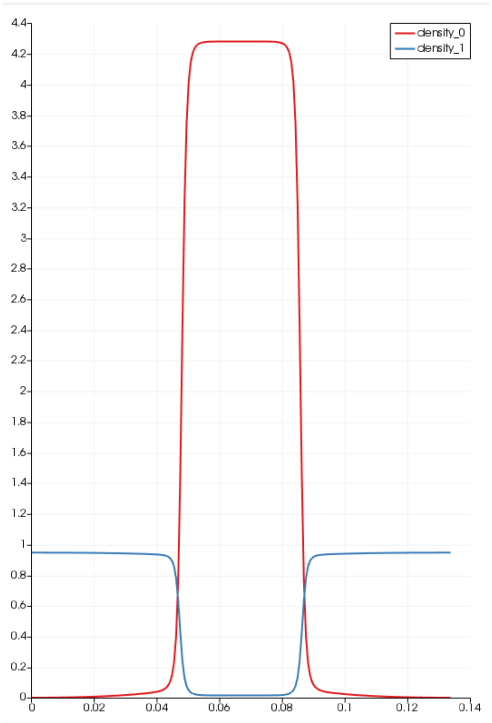
Figure 5 shows a validation performed. Static H_2O droplet with diameter of 30 LU placed in the center of a square domain filled with CO_2 with the size of 201×201 lattice unites (LU^2) and solid boundary at all four sides. The interaction parameter used is $g = 1$, as it gives good stability.



a) H_2O drop in red surrounded by an CO_2 in blue.



b) CO_2 in red surrounding H_2O in blue.



c) Density profile (in LU) at the cross-section of the domain taken at the horizontal centerline (y=100 LU).

Figure 5: Static H_2O droplet at equilibrium within CO_2 saturated medium.

There is a trace amount of fluids mixed in within each other, but the presence of a small quantity of second phase helps reduce the initial gradient at the phase interface, improving the numerical stability.

3.2. Contact angle test

This section presents the contact angle tests, which are essential for properly setting up interactions with the solid phase. It demonstrates the improved model is stable and valid for simulating fluid flows that involve fluid-solid interactions.

Figure 6 shows the multi-component model can create a wide range of contact angles.

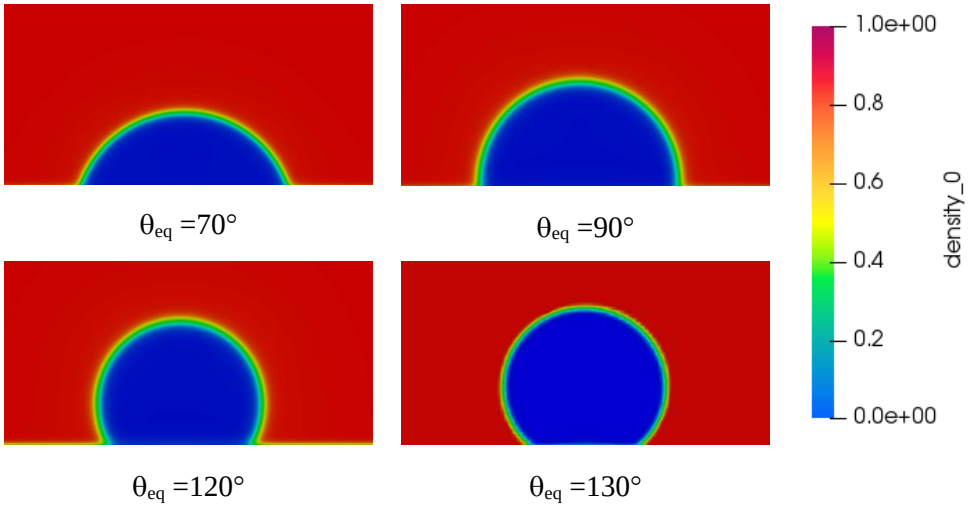


Figure 6: Water droplet contour for different contact angles obtained through adjusting the interaction parameter g_{wall}

In this example, the H_2O is as a wettable fluid, and the CO_2 is non-wettable. g_{wall} values used to perform these tests are shown in Table 1. The results of these tests demonstrate that the pseudopotential model is flexible enough to be used to model a wide range of contact angles. To control the angle the solid-fluid interaction parameters, $g_{CO_2,wall}$ and $g_{H_2O,wall}$ can be adjusted. Possibly, an even larger range of angles may be possible to , but too large of an increase in $|\mathbf{F}_{wet}|$ can lead to numerical instability. Boundaries of possible contact angles can be explored in the future.

Table 1: Values of $g_{CO_2,wall}$ and $g_{H_2O,wall}$ used to achieve different contact angles

θ_{eq}	$g_{CO_2,wall}$	$g_{H_2O,wall}$
70°	0.2	-0.2
90°	0	0
120°	-0.2	0.2
130°	-0.3	0.3

Chapter 4 – Porous medium model and grid verification

4.1. Displacement model

The modeling of CO_2 penetration into a uniform porous media under inlet velocity and outflow outlet was studied using the pressure EoS. The porous media is constructed by placing alternating rows of cylinders into a domain filled with H_2O. The schematic diagram of the porous region used is shown in Figure 7. It features a porosity of 0.78 and a pore throat, distance between two neighboring obstacles of 10 LU. The total domain size is 401×201 $LU^{2,}$ and the horizontal distance between furthers edges of rightmost and leftmost obstacles are 229 LU. For the boundary conditions the top and bottom are set as impermeable wall, the left side is velocity inlet, and the right side is outflow. The density and viscosity ratios are set to 1. The computational domain is subject to an inlet velocity of 0.02 LU/ts to drive the CO2 into the porous medium filled with H2O with a contact angle of 70°, as shown in 3.2. The whole setup can be seen in Figure 8.

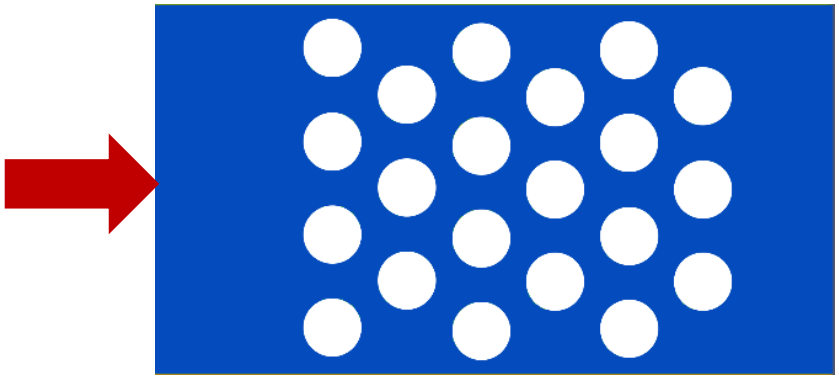


Figure 7: Geometry of the porous media with porosity 0.78

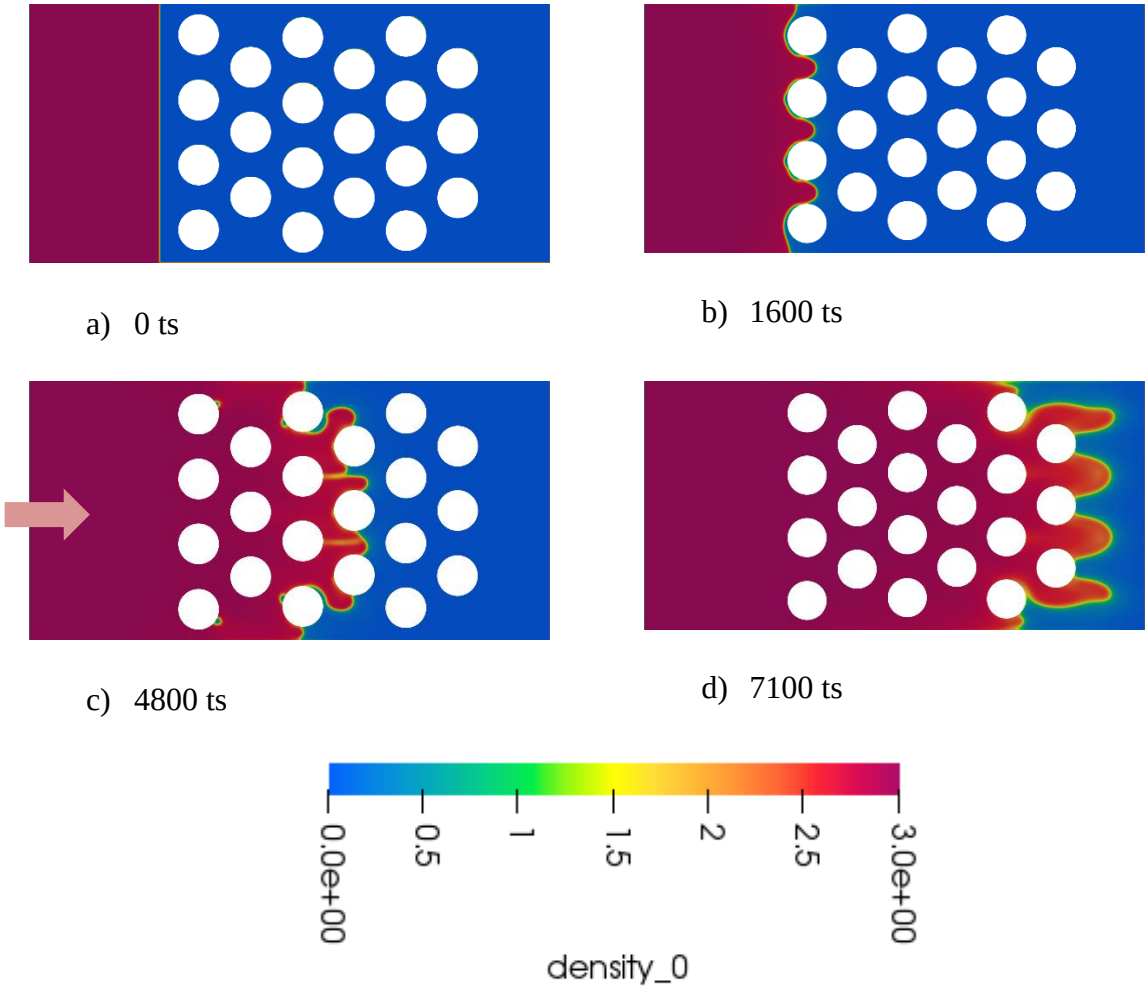


Figure 8: CO_2 penetration pattern with a flat front profile initially at 7 LU from pore network

4.2. Change of the viscosity ratio

As a further improvement, a similar model with an increased viscosity ratio is developed to test the stability of the model. The input parameters were kept the same, but the viscosity ratio was changed to 8.427. This value is suggested by [12] as a realistic value of water-to-CO_2 sequestration conditions at 308 K. This was achieved by changing the relaxation time from 1 to 4.71 for the H_2O as shown in Eq. (2).

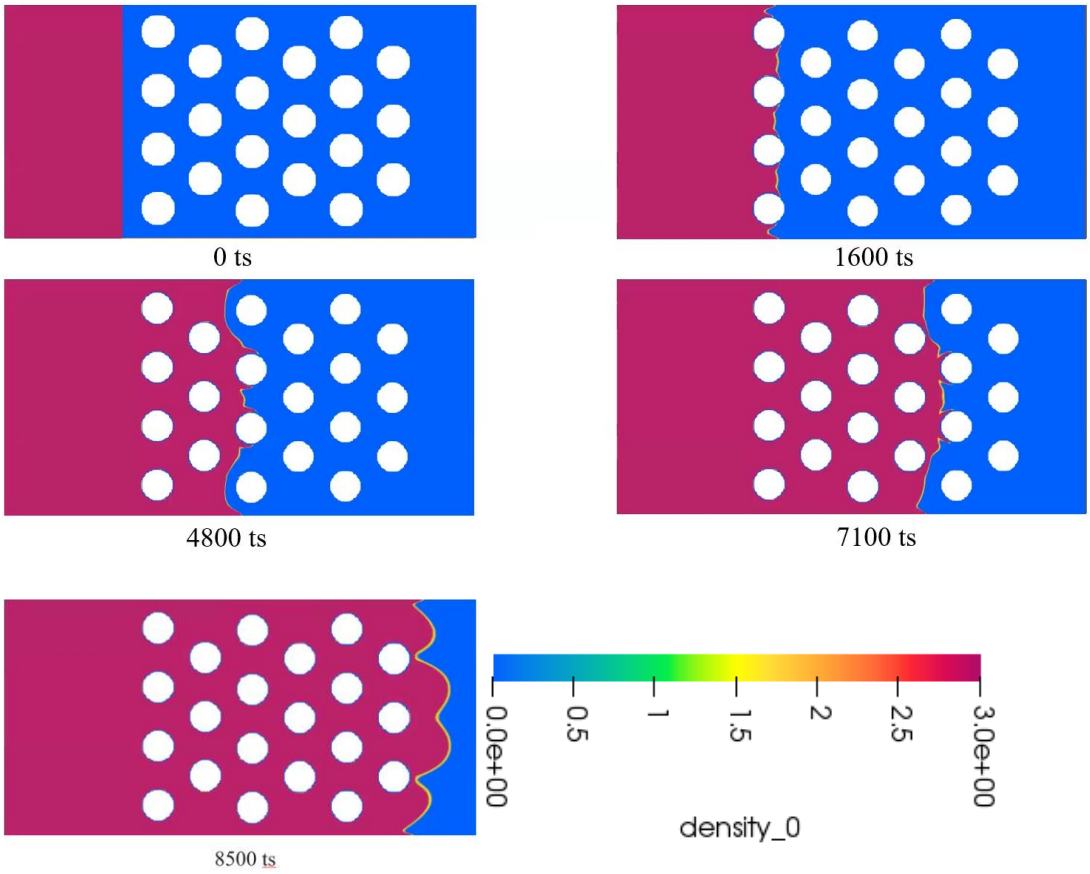


Figure 9: CO_2 penetration pattern with a viscosity ratio raised to 8.427

The issue of this approach is that it increases miscibility and would be unsuitable for even higher viscosity ratios. As shown in Figure 9, the fingering effect becomes much less pronounced. As a result, CO_2 does not reach the latest row of obstacles at 7100ts.

4.3. Grid verification

A grid-independence analysis is proposed by introducing two more refined models of 601×301 LU^2 and 801×401 LU^2, with geometry, inlet speed, viscosity, and timestep span scaled up accordingly, x1.5 and x2, using nearest neighbor scaling with the approach described in 2.3. To compare the simulations, the average velocity magnitude was measured along the line drawn over the last row of obstacles, as shown in Figure 10. Measurements of this velocity probe are shown in Figure 11, tracked from lattice time of 0 to 7100.

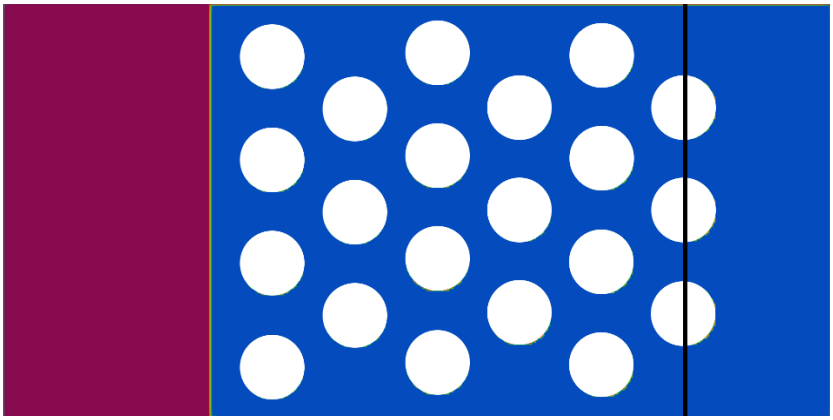


Figure 10: Line probe placement (shown in black) in the grid-independence analysis of porous medium displacement model

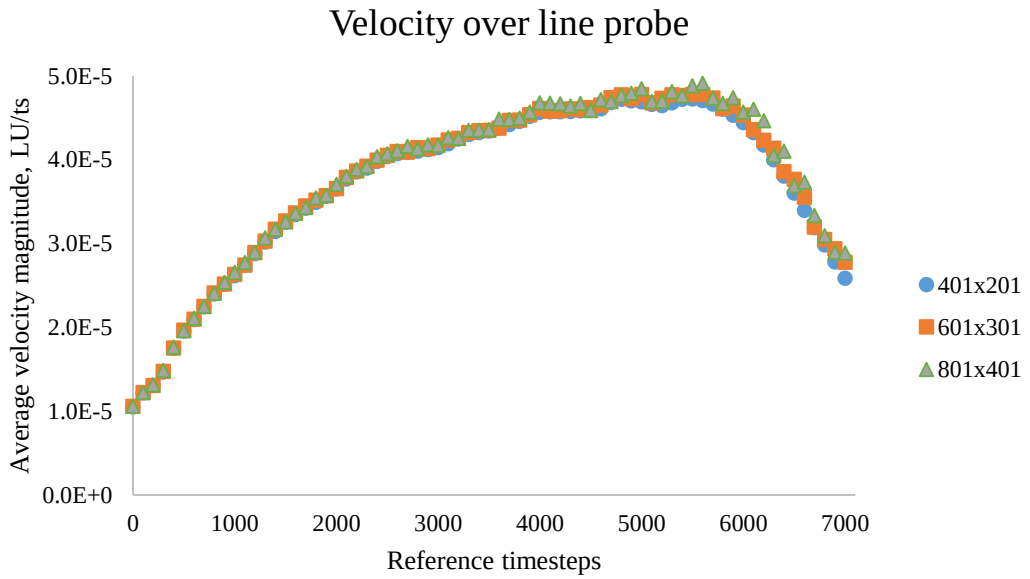


Figure 11: Average velocity magnitude along the probe line (shown in Fig. 10) in the CO_2 penetration LBM model

As shown in Figure 11, the velocity magnitudes are very close for three grid sizes, with only a small difference in the latest timesteps, suggesting the fitness of any of the three grids in our analysis.

For further comparison, CO_2 flux was measured through the probe line through the same time period. The flux was calculated as the fraction of CO_2 particles within all particles that crossed the line multiplied by the time-averaged velocity magnitude, integrated over the probe length. Results are shown in Table 1, highlighting that the error between grid

resolutions is small, with errors dipping below 1%, justifying our initially selected grid for the analysis.

Table 2: Time-space average CO_2 flux over a probe line (Figure 8) integrated over 7100 timesteps

Grid size, LU^2	CO_2 flux, ×10^-7 LU/ts	Relative error (%)
401×201	3.590	-
601×301	3.651	1.7%
801×401	3.679	0.76%

Chapter 5 – Discussion and concluding remarks

In this study, the pseudopotential LBM model was applied to study of immiscible fluid penetration process in capillary different fluid saturated domains, under the motivation of studying CO_2 sequestration. Model's performance was analyzed with two validation tests. Sufficient numerical stability was demonstrated with the static models, showing ability to describe different wettability regimes and stable immiscibility.

Next, the displacement process was analyzed, with the non-wetting fluid is introduced into a homogeneous complex pore network filled with wetting fluid. Results, while preliminary, demonstrate that the model is numerically stable when applied to dynamic models with geometries that are rather complex and with physical flow patterns, as expected in realistic conditions. However, the study is limited to qualitative results in isothermal conditions. Changing the viscosity ratio was explored, but the range of possible viscosities is quite limited in the pseudopotential model. A model size verification procedure was developed and showed acceptable results, but it was complex to set up, requiring considerable manual work from simulation to simulation. For proper domain verification dimensionless values have to preserved, but usually all of them cannot be preserved at once. Future work may include developing verification procedure further, with analysis of preserving different dimensionless numbers.

Future development may include exploring the numerical stability of the more complex formulations, including Peng-Robinson and crossover enhanced P-R EoS for the same model and various density and viscosity ratios, specifically ones described in work of Gooya et al [12], with data on density and viscosity ratios occurring in different temperature regimes of CO_2 sequestration. For variation in viscosity, different models of LBM may be explored, including the color-fluid model, which is notable for achieving high viscosity ratios. Alternatively, modifications of the pseudopotential formulation with a larger domain size may be explored to reduce miscibility.

References

[1] M. Atykhan, E. Monaco, and L. R. Rojas-Solórzano, "Modeling Immiscible Fluid Displacement in a Porous Medium Using Lattice Boltzmann Method," *Fluids,* vol. 6, no. 2, p. 89, 2021.

[2] B. Kabdenova, L. R. Rojas-Solórzano, and E. Monaco, "Lattice Boltzmann simulation of near/supercritical CO2 flow featuring a crossover formulation of the equation of state," *Computers & Fluids,* vol. 216, p. 104820, 2021.

[3] A. Ashirbekov, B. Kabdenova, E. Monaco, and L. R. Rojas-Solórzano, "Equation of State's Crossover Enhancement of Pseudopotential Lattice Boltzmann Modeling of CO2 Flow in Homogeneous Porous Media," *Fluids,* vol. 6, no. 12, p. 434, 2021.

[4] X. He and L.-S. Luo, "Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation," *Physical review E,* vol. 56, no. 6, p. 6811, 1997.

[5] H. Huang, M. Sukop, and X. Lu, "Multiphase lattice Boltzmann methods: Theory and application," 2015.

[6] G. Imani, "Lattice Boltzmann method for conjugate natural convection with heat generation on non-uniform meshes," *Computers & Mathematics with Applications,* vol. 79, no. 4, pp. 1188-1207, 2020.

[7] A. Hu, L. Li, and R. Uddin, "Force method in a pseudo-potential lattice Boltzmann model," *Journal of Computational Physics,* vol. 294, pp. 78-89, 2015.

[8] P. Yuan and L. Schaefer, "Equations of state in a lattice Boltzmann model," *Physics of Fluids,* vol. 18, no. 4, p. 042101, 2006.

[9] N. S. Martys and H. Chen, "Simulation of multicomponent fluids in complex three-dimensional geometries by the lattice Boltzmann method," *Physical Review E,* vol. 53, no. 1, pp. 743-750, 01/01/ 1996.

[10] X. Shan, "Analysis and reduction of the spurious current in a class of multiphase lattice Boltzmann models," *Physical Review E,* vol. 73, no. 4, p. 047701, 04/06/ 2006.

[11] X. Shan and H. Chen, "Lattice Boltzmann model for simulating flows with multiple phases and components," *Physical Review E,* vol. 47, no. 3, pp. 1815-1819, 03/01/ 1993.

[12] R. Gooya, A. Silvestri, A. Moaddel, M. P. Andersson, S. L. S. Stipp, and H. O. Sørensen, "Unstable, Super Critical CO2–Water Displacement in Fine Grained Porous Media under Geologic Carbon Sequestration Conditions," *Scientific Reports,* vol. 9, no. 1, p. 11272, 2019/08/02 2019.

Appendix A: Input files and parameters for the developed simulations

Following files are ".sys" files to be used with DL_MESO 2.6. They will be useful for this work's recreation purposes, and future work.

Droplet test inputs:

space_dimension	2	equilibration_step	0
discrete_speed	9	save_span	10
number_of_fluid	2	dump_span	30
number_of_solute	0	calculation_time	0.0
temperature_scalar	0	noise_intensity	0.0
phase_field	0	sound_speed	540.0
incompressible_fluids	0	kinetic_viscosity	0.001
restart_simulation	0	body_force_0	0.0
collision_type	BGK	body_force_1	0.0
interaction_type	ShanChen	body_force_2	0.0
output_format	VTK	body_force_3	0.0
output_type	Binary	body_force_4	0.0
output_combine_x	0	body_force_5	0.0
output_combine_y	0	oscillating_force_0	0.0
output_combine_z	0	oscillating_force_1	0.0
grid_number_x	201	oscillating_force_2	0.0
grid_number_y	201	oscillating_force_3	0.0
grid_number_z	1	oscillating_force_4	0.0
domain_boundary_width	1	oscillating_force_5	0.0
total_step	3000	oscillating_freq	0.0

boussinesq_force_0	0.0	eos_parameter_a_0	0.0
boussinesq_force_1	0.0	eos_parameter_b_0	0.0
boussinesq_force_2	0.0	acentric_factor_0	0.0
boussinesq_force_3	0.0	wetting_type_0	Potential
boussinesq_force_4	0.0	wall_interaction_0	-0.15
boussinesq_force_5	0.0	potential_type_1	Rho
speed_ini_0	0.0	eos_parameter_a_1	0.0
speed_ini_1	0.0	eos_parameter_b_1	0.0
speed_ini_2	0.0	acentric_factor_1	0.0
density_ini_0	1.0	wetting_type_1	Potential
density_ini_1	0.0	wall_interaction_1	0.15
density_inc_0	0.0	gas_constant	1.0
density_inc_1	1.0	temperature_system	1.0
rheology_fluid_0	Simple	gradient_order	1
rheology_fluid_1	Simple	boundary_type	ZouHe
relaxation_fluid_0	0.0	solute_boundary_type	ZouHe
bulk_relaxation_fluid_0	1.0	thermal_boundary_type	ZouHe
relaxation_fluid_1	1.0	speed_top_0	0.0
bulk_relaxation_fluid_1	1.0	speed_top_1	0.0
interaction_0	1.0	speed_top_2	0.0
interaction_1	1.0	speed_bot_0	0.0
interaction_2	1.0	speed_bot_1	0.0
interaction_3	0.0	speed_bot_2	0.0
potential_type_0	Rho	speed_lef_0	0.0

speed_lef_1	0.0	speed_oscil_rig_1	0.0
speed_lef_2	0.0	speed_oscil_rig_2	0.0
speed_rig_0	0.0	oscillating_freq_rig	0.0
speed_rig_1	0.0	speed_oscil_fro_0	0.0
speed_rig_2	0.0	speed_oscil_fro_1	0.0
speed_fro_0	0.0	speed_oscil_fro_2	0.0
speed_fro_1	0.0	oscillating_freq_fro	0.0
speed_fro_2	0.0	speed_oscil_bac_0	0.0
speed_bac_0	0.0	speed_oscil_bac_1	0.0
speed_bac_1	0.0	speed_oscil_bac_2	0.0
speed_bac_2	0.0	oscillating_freq_bac	0.0
speed_oscil_top_0	0.0	density_top_0	0.0
speed_oscil_top_1	0.0	density_bot_0	0.0
speed_oscil_top_2	0.0	density_lef_0	0.0
oscillating_freq_top	0.0	density_rig_0	0.0
speed_oscil_bot_0	0.0	density_fro_0	0.0
speed_oscil_bot_1	0.0	density_bac_0	0.0
speed_oscil_bot_2	0.0	density_top_1	0.0
oscillating_freq_bot	0.0	density_bot_1	0.0
speed_oscil_lef_0	0.0	density_lef_1	0.0
speed_oscil_lef_1	0.0	density_rig_1	0.0
speed_oscil_lef_2	0.0	density_fro_1	0.0
oscillating_freq_lef	0.0	density_bac_1	0.0
speed_oscil_rig_0	0.0	selectTC	0

Contact angle tests:

space_dimension	2
discrete_speed	9
number_of_fluid	2
number_of_solute	0
temperature_scalar	0
phase_field	0
incompressible_fluids	0
restart_simulation	0
collision_type	BGK
interaction_type	ShanChen
output_format	VTK
output_type	Binary
output_combine_x	0
output_combine_y	0
output_combine_z	0
grid_number_x	101
grid_number_y	101
grid_number_z	1
domain_boundary_width	1
total_step	10000
equilibration_step	0
save_span	500
dump_span	10000
calculation_time	0.0
noise_intensity	0.00
sound_speed	540.0
kinetic_viscosity	0.001
body_force_0	0.0
body_force_1	0.0
body_force_2	0.0
body_force_3	0.0
body_force_4	0.0
body_force_5	0.0
oscillating_force_0	0.0
oscillating_force_1	0.0
oscillating_force_2	0.0
oscillating_force_3	0.0
oscillating_force_4	0.0
oscillating_force_5	0.0
oscillating_freq	0.0
boussinesq_force_0	0.0
boussinesq_force_1	0.0
boussinesq_force_2	0.0
boussinesq_force_3	0.0
boussinesq_force_4	0.0
boussinesq_force_5	0.0
speed_ini_0	0.0
speed_ini_1	0.0
speed_ini_2	0.0
density_ini_0	1.0
density_ini_1	0.000
density_inc_0	0.000

density_inc_1	1.0
rheology_fluid_0	Simple
rheology_fluid_1	Simple
relaxation_fluid_0	1.0
relaxation_fluid_1	1.0
interaction_0	0.0
interaction_1	1.5
interaction_2	1.5
interaction_3	0.0
potential_type_0	Rho
wetting_type_0	Potential
wall_interaction_0	-0.2
potential_type_1	Rho
wetting_type_1	Potential
wall_interaction_1	0.2
gradient_order	1
boundary_type	ZouHe
solute_boundary_type	ZouHe
thermal_boundary_type	ZouHe
speed_top_0	0.0
speed_top_1	0.0
speed_top_2	0.0
speed_bot_0	0.0
speed_bot_1	0.0
speed_bot_2	0.0
speed_lef_0	0.0
speed_lef_1	0.0
speed_lef_2	0.0
speed_rig_0	0.0
speed_rig_1	0.0
speed_rig_2	0.0
speed_fro_0	0.0
speed_fro_1	0.0
speed_fro_2	0.0
speed_bac_0	0.0
speed_bac_1	0.0
speed_bac_2	0.0
speed_oscil_top_0	0.0
speed_oscil_top_1	0.0
speed_oscil_top_2	0.0
oscillating_freq_top	0.0
speed_oscil_bot_0	0.0
speed_oscil_bot_1	0.0
speed_oscil_bot_2	0.0
oscillating_freq_bot	0.0
speed_oscil_lef_0	0.0
speed_oscil_lef_1	0.0
speed_oscil_lef_2	0.0
oscillating_freq_lef	0.0
speed_oscil_rig_0	0.0
speed_oscil_rig_1	0.0
speed_oscil_rig_2	0.0

oscillating_freq_rig	0.0	density_lef_0	0.0
speed_oscil_fro_0	0.0	density_rig_0	0.0
speed_oscil_fro_1	0.0	density_fro_0	0.0
speed_oscil_fro_2	0.0	density_bac_0	0.0
oscillating_freq_fro	0.0	density_top_1	0.0
speed_oscil_bac_0	0.0	density_bot_1	0.0
speed_oscil_bac_1	0.0	density_lef_1	0.0
speed_oscil_bac_2	0.0	density_rig_1	0.0
oscillating_freq_bac	0.0	density_fro_1	0.0
density_top_0	0.0	density_bac_1	0.0
density_bot_0	0.0		

For different values of wall interaction change “wall_interaction_0” and “wall_interaction_1” accordingly.

Porous medium displacement:

space_dimension	2	body_force_2	0.0
discrete_speed	9	body_force_3	0.0
number_of_fluid	2	body_force_4	0.0
number_of_solute	0	body_force_5	0.0
temperature_scalar	0	oscillating_force_0	0.0
phase_field	0	oscillating_force_1	0.0
incompressible_fluids	0	oscillating_force_2	0.0
restart_simulation	0	oscillating_force_3	0.0
collision_type	BGK	oscillating_force_4	0.0
interaction_type	ShanChen	oscillating_force_5	0.0
output_format	VTK	oscillating_freq	0.0
output_type	Binary	boussinesq_force_0	0.0
output_combine_x	0	boussinesq_force_1	0.0
output_combine_y	0	boussinesq_force_2	0.0
output_combine_z	0	boussinesq_force_3	0.0
grid_number_x	401	boussinesq_force_4	0.0
grid_number_y	201	boussinesq_force_5	0.0
grid_number_z	1	speed_ini_0	0.0
domain_boundary_width	1	speed_ini_1	0.0
total_step	10000	speed_ini_2	0.0
equilibration_step	0	density_ini_0	1.0
save_span	100	density_ini_1	1.0
dump_span	300	density_inc_0	0.0
calculation_time	0.0	density_inc_1	0.0
noise_intensity	0.0	rheology_fluid_0	Simple
sound_speed	540.0	rheology_fluid_1	Simple
kinetic_viscosity	0.001	relaxation_fluid_0	1.0
body_force_0	0.0	bulk_relaxation_fluid_0	1.0
body_force_1	0.0	relaxation_fluid_1	1.0

bulk_relaxation_fluid_1	1.0	speed_bot_2	0.0
mrt_relax_freq_0	1.14	speed_lef_0	0.02
mrt_relax_freq_1	1.92	speed_lef_1	0.0
interaction_0	0.0	speed_lef_2	0.0
interaction_1	1.5	speed_rig_0	0.0
interaction_2	1.5	speed_rig_1	0.0
interaction_3	0.0	speed_rig_2	0.0
potential_type_0	Rho	speed_fro_0	0.0
eos_parameter_a_0	0.0	speed_fro_1	0.0
eos_parameter_b_0	0.0	speed_fro_2	0.0
acentric_factor_0	0.0	speed_bac_0	0.0
wetting_type_0	Potential	speed_bac_1	0.0
wall_interaction_0	-0.15	speed_bac_2	0.0
potential_type_1	Rho	speed_oscil_top_0	0.0
eos_parameter_a_1	0.0	speed_oscil_top_1	0.0
eos_parameter_b_1	0.0	speed_oscil_top_2	0.0
acentric_factor_1	0.0	oscillating_freq_top	0.0
wetting_type_1	Potential	speed_oscil_bot_0	0.0
wall_interaction_1	0.15	speed_oscil_bot_1	0.0
gas_constant	1.0	speed_oscil_bot_2	0.0
temperature_system	1.0	oscillating_freq_bot	0.0
gradient_order	1	speed_oscil_lef_0	0.0
boundary_type	ZouHe	speed_oscil_lef_1	0.0
solute_boundary_type	ZouHe	speed_oscil_lef_2	0.0
thermal_boundary_type	ZouHe	oscillating_freq_lef	0.0
speed_top_0	0.0	speed_oscil_rig_0	0.0
speed_top_1	0.0	speed_oscil_rig_1	0.0
speed_top_2	0.0	speed_oscil_rig_2	0.0
speed_bot_0	0.0	oscillating_freq_rig	0.0
speed_bot_1	0.0	speed_oscil_fro_0	0.0

speed_oscil_fro_1	0.0	density_rig_0	0.0
speed_oscil_fro_2	0.0	density_fro_0	0.0
oscillating_freq_fro	0.0	density_bac_0	0.0
speed_oscil_bac_0	0.0	density_top_1	0.0
speed_oscil_bac_1	0.0	density_bot_1	0.0
speed_oscil_bac_2	0.0	density_lef_1	0.0
oscillating_freq_bac	0.0	density_rig_1	0.0
density_top_0	0.0	density_fro_1	0.0
density_bot_0	0.0	density_bac_1	0.0
density_lef_0	0.0	selectTC	1

When prompted for initialization layer thickness, enter “300”.

For the model with increased viscosity ratio, use the same code, but change value of “bulk_relaxation_fluid_1” to 4.71.

For domain verification, 601x301 model:

space_dimension	2	body_force_5	0.0
discrete_speed	9	oscillating_force_0	0.0
number_of_fluid	2	oscillating_force_1	0.0
number_of_solute	0	oscillating_force_2	0.0
temperature_scalar	0	oscillating_force_3	0.0
phase_field	0	oscillating_force_4	0.0
incompressible_fluids	0	oscillating_force_5	0.0
restart_simulation	0	oscillating_freq	0.0
collision_type	BGK	boussinesq_force_0	0.0
interaction_type ShanChen		boussinesq_force_1	0.0
output_format	VTK	boussinesq_force_2	0.0
output_type Binary		boussinesq_force_3	0.0
output_combine_x	0	boussinesq_force_4	0.0
output_combine_y	0	boussinesq_force_5	0.0
output_combine_z	0	speed_ini_0	0.0
grid_number_x	601	speed_ini_1	0.0
grid_number_y	301	speed_ini_2	0.0
grid_number_z	1	density_ini_0	1.0
domain_boundary_width	1	density_ini_1	1.0
total_step	20000	density_inc_0	0.0
equilibration_step	0	density_inc_1	0.0
save_span	200	rheology_fluid_0 Simple	
dump_span	300	rheology_fluid_1 Simple	
calculation_time	0.0	relaxation_fluid_0	1.0
noise_intensity	0.0	bulk_relaxation_fluid_0	1.0
sound_speed	540.0	relaxation_fluid_1	1.0
kinetic_viscosity	0.003	bulk_relaxation_fluid_1	1.0
body_force_0	0.0	mrt_relax_freq_0	1.14
body_force_1	0.0	mrt_relax_freq_1	1.92
body_force_2	0.0	interaction_0	0.0
body_force_3	0.0	interaction_1	1.5
body_force_4	0.0	interaction_2	1.5

interaction_3	0.0
potential_type_0	Rho
eos_parameter_a_0	0.0
eos_parameter_b_0	0.0
acentric_factor_0	0.0
wetting_type_0 Potential	
wall_interaction_0	-0.15
potential_type_1	Rho
eos_parameter_a_1	0.0
eos_parameter_b_1	0.0
acentric_factor_1	0.0
wetting_type_1 Potential	
wall_interaction_1	0.15
gas_constant	1.0
temperature_system	1.0
gradient_order	1
boundary_type	ZouHe
solute_boundary_type	ZouHe
thermal_boundary_type	ZouHe
speed_top_0	0.0
speed_top_1	0.0
speed_top_2	0.0
speed_bot_0	0.0
speed_bot_1	0.0
speed_bot_2	0.0
speed_lef_0	0.03
speed_lef_1	0.0
speed_lef_2	0.0
speed_rig_0	0.0
speed_rig_1	0.0
speed_rig_2	0.0
speed_fro_0	0.0
speed_fro_1	0.0
speed_fro_2	0.0
speed_bac_0	0.0
speed_bac_1	0.0
speed_bac_2	0.0
speed_oscil_top_0	0.0
speed_oscil_top_1	0.0
speed_oscil_top_2	0.0
oscillating_freq_top	0.0
speed_oscil_bot_0	0.0
speed_oscil_bot_1	0.0
speed_oscil_bot_2	0.0
oscillating_freq_bot	0.0
speed_oscil_lef_0	0.0
speed_oscil_lef_1	0.0
speed_oscil_lef_2	0.0
oscillating_freq_lef	0.0
speed_oscil_rig_0	0.0
speed_oscil_rig_1	0.0
speed_oscil_rig_2	0.0
oscillating_freq_rig	0.0
speed_oscil_fro_0	0.0
speed_oscil_fro_1	0.0
speed_oscil_fro_2	0.0
oscillating_freq_fro	0.0
speed_oscil_bac_0	0.0
speed_oscil_bac_1	0.0
speed_oscil_bac_2	0.0
oscillating_freq_bac	0.0
density_top_0	0.0
density_bot_0	0.0
density_lef_0	0.0
density_rig_0	0.0
density_fro_0	0.0
density_bac_0	0.0
density_top_1	0.0

density_bot_1	0.0	density_fro_1	0.0
density_lef_1	0.0	density_bac_1	0.0
density_rig_1	0.0	selectTC	1

When prompted for initialization layer thickness, enter “450”.

And 801x401 model:

space_dimension	2	body_force_5	0.0
discrete_speed	9	oscillating_force_0	0.0
number_of_fluid	2	oscillating_force_1	0.0
number_of_solute	0	oscillating_force_2	0.0
temperature_scalar	0	oscillating_force_3	0.0
phase_field	0	oscillating_force_4	0.0
incompressible_fluids	0	oscillating_force_5	0.0
restart_simulation	0	oscillating_freq	0.0
collision_type	BGK	boussinesq_force_0	0.0
interaction_type ShanChen		boussinesq_force_1	0.0
output_format	VTK	boussinesq_force_2	0.0
output_type Binary		boussinesq_force_3	0.0
output_combine_x	0	boussinesq_force_4	0.0
output_combine_y	0	boussinesq_force_5	0.0
output_combine_z	0	speed_ini_0	0.0
grid_number_x	801	speed_ini_1	0.0
grid_number_y	401	speed_ini_2	0.0
grid_number_z	1	density_ini_0	1.0
domain_boundary_width	1	density_ini_1	1.0
total_step	20000	density_inc_0	0.0
equilibration_step	0	density_inc_1	0.0
save_span	200	rheology_fluid_0 Simple	
dump_span	300	rheology_fluid_1 Simple	
calculation_time	0.0	relaxation_fluid_0	1.0
noise_intensity	0.0	bulk_relaxation_fluid_0	1.0
sound_speed	540.0	relaxation_fluid_1	1.0
kinetic_viscosity	0.004	bulk_relaxation_fluid_1	1.0
body_force_0	0.0	mrt_relax_freq_0	1.14
body_force_1	0.0	mrt_relax_freq_1	1.92
body_force_2	0.0	interaction_0	0.0
body_force_3	0.0	interaction_1	1.5
body_force_4	0.0	interaction_2	1.5

interaction_3	0.0
potential_type_0	Rho
eos_parameter_a_0	0.0
eos_parameter_b_0	0.0
acentric_factor_0	0.0
wetting_type_0 Potential	
wall_interaction_0	-0.15
potential_type_1	Rho
eos_parameter_a_1	0.0
eos_parameter_b_1	0.0
acentric_factor_1	0.0
wetting_type_1 Potential	
wall_interaction_1	0.15
gas_constant	1.0
temperature_system	1.0
gradient_order	1
boundary_type	ZouHe
solute_boundary_type	ZouHe
thermal_boundary_type	ZouHe
speed_top_0	0.0
speed_top_1	0.0
speed_top_2	0.0
speed_bot_0	0.0
speed_bot_1	0.0
speed_bot_2	0.0
speed_lef_0	0.04
speed_lef_1	0.0
speed_lef_2	0.0
speed_rig_0	0.0
speed_rig_1	0.0
speed_rig_2	0.0
speed_fro_0	0.0
speed_fro_1	0.0
speed_fro_2	0.0
speed_bac_0	0.0
speed_bac_1	0.0
speed_bac_2	0.0
speed_oscil_top_0	0.0
speed_oscil_top_1	0.0
speed_oscil_top_2	0.0
oscillating_freq_top	0.0
speed_oscil_bot_0	0.0
speed_oscil_bot_1	0.0
speed_oscil_bot_2	0.0
oscillating_freq_bot	0.0
speed_oscil_lef_0	0.0
speed_oscil_lef_1	0.0
speed_oscil_lef_2	0.0
oscillating_freq_lef	0.0
speed_oscil_rig_0	0.0
speed_oscil_rig_1	0.0
speed_oscil_rig_2	0.0
oscillating_freq_rig	0.0
speed_oscil_fro_0	0.0
speed_oscil_fro_1	0.0
speed_oscil_fro_2	0.0
oscillating_freq_fro	0.0
speed_oscil_bac_0	0.0
speed_oscil_bac_1	0.0
speed_oscil_bac_2	0.0
oscillating_freq_bac	0.0
density_top_0	0.0
density_bot_0	0.0
density_lef_0	0.0
density_rig_0	0.0
density_fro_0	0.0
density_bac_0	0.0
density_top_1	0.0

density_bot_1	0.0	density_fro_1	0.0
density_lef_1	0.0	density_bac_1	0.0
density_rig_1	0.0	selectTC	1

When prompted for initialization layer thickness, enter “600”.

Appendix B: Geometry scaling code

This code was developed in MATLAB and should be saved as ".m" script. Even though only 2D models were considered, this code can work with 3D models as well. Define original size of the domain as `original size` and required scale factor as `scaling`. Scaling factors below 0.5 and 2 are not advisable, may lead to discontinuities depending on geometry. Place geometry ".spa" file in the same directory as this ".m" script, define filename as `name` and run the code.

```
%%Start
originalsize=[401 401]; %original domain size
scaling = 2;          %scaling factor
method = 'bilinear';     %bilinear by default
name = 'lbin.spa';      %name of file to convert

%Reading original file
fileID = fopen(name,'r');
formatSpec = '%i %i %i %i';
file_original = fscanf(fileID,formatSpec,[4 Inf])';
fclose(fileID);

%Initialization
k=1;
file_scaled = zeros(round(size(file_original,1)*scaling^2),4);
type = rot90(unique(file_original(:,4)));

for node = type %Loop for each node type
    %Creating original image
    A = false(originalsize+1);
    for i = 1:size(file_original,1)
        if file_original(i,4) == node
            A(file_original(i,1)+1,file_original(i,2)+1) = true;
        end
    end
    imA = image(A,'CDataMapping','scaled');

    %Scaling
    B = imresize(A, scaling,method);
```

```
    image(B,'CDataMapping','scaled');

    %Scaled file creation
    for i = 1:size(B,1)
        for j = 1:size(B,2)
            if B(i,j)
                file_scaled(k,1)=i;
                file_scaled(k,2)=j;
                file_scaled(k,4)=node;
                k = k+1;
            end
        end
    end
end

%Scaled file writing
trim = ~any(file_scaled,2); %Trimming zero rows
file_scaled(trim,:) = [];

dot_position = strfind(name,'.'); %Creating output file name
name_scaled = name(1:dot_position-1);
extension = name(dot_position:end);

fileID = fopen(append(name_scaled,'_scaled',extension),'w');
fprintf(fileID, '%i %i %i %i\n', (file_scaled)');
fclose(fileID);
```

Similar code and procedure for initialization files:

```
%%Start
originalsize=[401 401]; %original domain size
scaling = 2;          %scaling factor
method = 'nearest';     %bilinear by default
name = 'lbin.init';      %name of file to convert

%Reading original file
fileID = fopen(name,'r');
formatSpec = '%i %i %i %i %i %i %i %i';
file_original = fscanf(fileID,formatSpec,[8 Inf])';
fclose(fileID);

%Initialization
k=1;
file_scaled = zeros(round(size(file_original,1)*scaling^2),8);
type = rot90(unique(file_original(:,8)));

for node = type %Loop for each node type
    %Creating original image
    A = false(originalsize+1);
    for i = 1:size(file_original,1)
            A(file_original(i,1)+1,file_original(i,2)+1) = true;
    end
%    figure(node*2)
    imA = image(A,'CDataMapping','scaled');

    %Scaling
    B = imresize(A, scaling,method);
%    figure(node*2+1)
    image(B,'CDataMapping','scaled');

    %Scaled file creation
    for i = 1:size(B,1)
        for j = 1:size(B,2)
            if B(i,j)
```

```
                file_scaled(k,1)=i;
                file_scaled(k,2)=j;
                file_scaled(k,8)=1;
                for t = 3:7
                    file_scaled(k,t)=0;
                end
                k = k+1;
            end
        end
    end
end

%Scaled file writing
trim = ~any(file_scaled,2); %Trimming zero rows
file_scaled(trim,:) = [];

dot_position = strfind(name,'.'); %Creating output file name
name_scaled = name(1:dot_position-1);
extension = name(dot_position:end);

fileID = fopen(append(name_scaled,'_scaled',extension),'w');
fprintf(fileID, '%i %i %i %i %i %i %i %i\n', (file_scaled)');
fclose(fileID);
```