Department of Applied Physics
Master Nano, Quantum and Photonics
Research group Fluids and Flows

Lava

Master Final Project

D. Schrijver (1623281)

Supervisors:

Eindhoven, January, 2025

Abstract

Table of Contents

1 Introduction **4**

2 Theory **5**
2.1 Lattice Boltzmann Equation . . . 5
2.2 Collision and streaming . . . 5
2.3 Boundary conditions . . . 6
2.4 Basic algorithm . . . 7
2.5 Sources . . . 8
2.6 Algorithm with sources . . . 8
2.7 Poiseuille flow . . . 8
2.8 Multicomponent Shan-Chen method . . . 9
2.9 Algorithm multicomponent Shan-Chen . . . 11
2.10 Multicomponent Poiseuille flow . . . 11
2.11 Temperature . . . 12
2.12 Algorithm thermal Lattice Boltzmann . . . 13
2.13 Rayleigh-Bénard convection . . . 13
2.14 Phase change . . . 15
2.15 Algorithm phase change . . . 15
2.16 Stefan problem . . . 16
2.17 Phase change in two-component fluid . . . 17
2.18 Algorithm phase change in two-component fluid . . . 17

3 Results **19**

4 Appendix **21**

1 Introduction

2 Theory

The Lattice Boltzmann Equation makes use of the populations indicated by the vector $\boldsymbol{f}$. For two-dimensional simulations, we choose to have nine velocity directions, indicated by D2Q9. This means that the populations $\boldsymbol{f}$ have nine elements. The nine discrete velocity components and their respective weights are given by:

$$\begin{aligned} \{\boldsymbol{c}_i\} &= \{(0,0),\ (\pm1,0),\ (0,\pm1),\ (\pm1,\pm1)\} \\ \{w_i\} &= \{4/9,\ 1/9,\ 1/9,\ 1/9,\ 1/9,\ 1/36,\ 1/36,\ 1/36,\ 1/36\} \end{aligned} \tag{1}$$

For a single fluid component in the absence of forces, the macroscopic quanties can be computed from the populations:

$$\begin{aligned} \rho &= \sum_i f_i \\ \boldsymbol{u} &= \frac{1}{\rho}\sum_i f_i \boldsymbol{c}_i \\ p &= c_s^2 \rho \end{aligned} \tag{2}$$

Where c_s is the speed of sound, with $c_s^2 = 1/3$ for the D2Q9 velocity set.

2.1 Lattice Boltzmann Equation

We can now introduce the Lattice Boltzmann Equation for a single fluid component in the absence of any sources:

$$f_i(\boldsymbol{x}+\boldsymbol{c}_i, t+1) = f_i(\boldsymbol{x},t) + \Omega_{ij}\Big(f_j(\boldsymbol{x},t) - f_j^{\mathrm{eq}}(\boldsymbol{x},t)\Big) \tag{3}$$

Where a sum over the index j is implied. We already choose the timestep and the gridspacing as $\Delta t = \Delta x = 1$ as is often done. If the populations will not change anymore under collisions, they will equation the equilibrium populations given by $\boldsymbol{f}^{\mathrm{eq}}$. This can be computed as:

$$f_i^{\mathrm{eq}} = w_i \rho \left(1 + \frac{\boldsymbol{u}\cdot\boldsymbol{c}_i}{c_s^2} + \frac{(\boldsymbol{u}\cdot\boldsymbol{c}_i)^2}{2c_s^4} - \frac{|\boldsymbol{u}|^2}{2c_s^2}\right) \tag{4}$$

Furthermore, the collision operator is given by the matrix $\boldsymbol{\Omega}$. We choose the simple BGK operator:

$$\boldsymbol{\Omega}_{\mathrm{BGK}} = -\frac{1}{\tau}\boldsymbol{I} \tag{5}$$

Where $\boldsymbol{I}$ is the identity matrix and τ is the relaxation time, which is related to the kinematic viscosity of the fluid as:

$$\nu = c_s^2\left(\tau - \frac{1}{2}\right) \tag{6}$$

2.2 Collision and streaming

The Lattice Boltzmann equation 3 can be split up in a collision step and a streaming step. The collision step is given by:

$$\boldsymbol{f}^*(\boldsymbol{x},t) = \boldsymbol{f}(\boldsymbol{x},t) - \frac{1}{\tau}\Big(\boldsymbol{f}(\boldsymbol{x},t) - \boldsymbol{f}^{\mathrm{eq}}(\boldsymbol{x},t)\Big) \tag{7}$$

And the streaming step is given by:

$$f_i(\boldsymbol{x}, t+1) = f_i^*(\boldsymbol{x} - \boldsymbol{c}_i, t) \tag{8}$$

Which together are equivalent to equation 3.

2.3 Boundary conditions

During the streaming step in equation 8, it is possible that the vector $\boldsymbol{x} - \boldsymbol{c}_i$ will be outside of the domain. When this happens, we will apply the suitable boundary conditions. When the crossed boundary is periodic, the streaming step can be written as:

$$f_i(\boldsymbol{x}, t+1) = f_i^*(\boldsymbol{x} - \boldsymbol{c}_i \mod \boldsymbol{L}, t) \tag{9}$$

Where $\boldsymbol{L}$ is a vector containing the sidelengths of the domain. When the crossed boundary is a no-slip boundary, we will make use of the half-way bounce-back scheme. We assume that the wall is half a gridpoint removed from the outermost gridpoints. During the streaming step, the populations will hit the wall at $t + 1/2$ and return to the original gridpoint at $t + 1$, with inverse velocity:

$$f_i(\boldsymbol{x}, t+1) = f_q^*(\boldsymbol{x}, t) \tag{10}$$

Where the index q is given by:

$$\boldsymbol{c}_q = -\boldsymbol{c}_i \tag{11}$$

We can also have open boundaries. We either impose a velocity profile or a density at these boundaries. A velocity is imposed as:

$$f_i(\boldsymbol{x}, t+1) = f_q^*(\boldsymbol{x}, t) - 2w_q\rho_{\text{in/out}}\frac{\boldsymbol{c}_q \cdot \boldsymbol{u}_{\text{in/out}}}{c_s^2} \tag{12}$$

Note that a density at the inlet/outlet still needs to be chosen or interpolated. We can also impose a pressure difference between the inlet and outlet by choosing the densities to be specific values, because the pressure is directly proportional to the density. This is done using the anti-bounce-back scheme:

$$f_i(\boldsymbol{x}, t+1) = -f_q^*(\boldsymbol{x}, t) + 2w_q\rho_{\text{in/out}}\left(1 + \frac{\left(\boldsymbol{c}_q \cdot \boldsymbol{u}_{\text{in/out}}\right)^2}{2c_s^4} - \frac{|\boldsymbol{u}_{\text{in/out}}|^2}{2c_s^2}\right) \tag{13}$$

Where are not imposing a velocity at the outlet, only a density. This means that we need to find the value of $\boldsymbol{u}_{\text{in/out}}$, which can be done using interpolation of the known velocity at the gridpoints in the fluid. We will assume for simplicity that the velocity of the fluid at the outlet is equal to the velocity of the closest gridpoint.

Another method of imposing a certain velocity or density at the boundaries is by making use of the wet-node approach. The nodes now lie infinitesimally close to the wall, meaning that we directly apply the inlet density or inlet velocity on the nodes themselves.

$$\rho_{\text{in}} = f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8 \tag{14}$$

$$\rho_{\text{in}}u_{\text{in}} = -\left(f_2 + f_6 + f_8\right) + \left(f_1 + f_5 + f_7\right) \tag{15}$$

$$\rho_{\text{in}}v_{\text{in}} = -\left(f_4 + f_6 + f_7\right) + \left(f_3 + f_5 + f_8\right) \tag{16}$$

By assuming that the velocity in the x-direction is zero, we can solve for u:

$$u_{\text{in}} = 1 - \frac{1}{\rho_{\text{in}}}\Big(2\left(f_2 + f_6 + f_8\right) + \left(f_0 + f_3 + f_4\right)\Big) \tag{17}$$

Using a similar procedure, the velocity at the outlet can be calculated as:

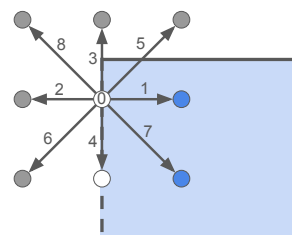


Figure 1: NEBB scheme sketch

$$u_{\text{out}} = -1 + \frac{1}{\rho_{\text{out}}}\Big(2\left(f_1 + f_5 + f_7\right) + \left(f_0 + f_3 + f_4\right)\Big) \tag{18}$$

For the inlet, the NEBB method gives the following values for the unknown populations:

$$\begin{aligned} f_1 &= f_2 + \frac{2}{3}\rho_{\text{in}} u_{\text{in}} \\ f_5 &= f_6 - \frac{1}{2}\left(f_3 - f_4\right) + \frac{1}{6}u_{\text{in}} \\ f_7 &= f_8 + \frac{1}{2}\left(f_3 - f_4\right) + \frac{1}{6}u_{\text{in}}; \end{aligned} \tag{19}$$

And for the outlet:

$$\begin{aligned} f_2 &= f_1 - \frac{2}{3}\rho_{\text{out}} u_{\text{out}} \\ f_6 &= f_5 + \frac{1}{2}\left(f_3 - f_4\right) - \frac{1}{6}u_{\text{out}} \\ f_8 &= f_7 - \frac{1}{2}\left(f_3 - f_4\right) - \frac{1}{6}u_{\text{out}}; \end{aligned} \tag{20}$$

2.4 Basic algorithm

The basic algorithm for a single component without any sources, given by the Lattice Boltzmann equation 3, can be desribed as follows:

(i) Choose $\rho(t = 0)$ and $\boldsymbol{u}(t = 0)$.

(ii) Initialize the populations as $\boldsymbol{f}(t = 0) = \boldsymbol{f}^{\text{eq}}(\rho, \boldsymbol{u})$.

1 Calculate $\boldsymbol{f}^{\text{eq}}(\rho, \boldsymbol{u})$.

2 Collide the populations $\rightarrow \boldsymbol{f}^*$.

3 Stream the populations $\rightarrow \boldsymbol{f}(t + 1)$.

4 Calculate macroscopic quantities $\rho(t + 1)$ and $\boldsymbol{u}(t + 1)$.

5 $t = t + 1$ and return to step 1 while $t < t_{\text{final}}$.

2.5 Sources

When forces are applied on the fluid, the right hand side of the Lattice Boltzmann Equation in equation 3 contains a source term S_i. We compute this source term from the applied force field $\boldsymbol{F}(\boldsymbol{x}, t)$ as:

$$S_i(\boldsymbol{x}, t) = \left(1 - \frac{1}{2\tau}\right) w_i \left(\frac{\boldsymbol{c}_i - \boldsymbol{u}(\boldsymbol{x}, t)}{c_s^2} + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}(\boldsymbol{x}, t)}{c_s^4} \boldsymbol{c}_i\right) \cdot \boldsymbol{F}(\boldsymbol{x}, t) \tag{21}$$

Note that a correction is added for the computation of the velocity field:

$$\boldsymbol{u} = \frac{1}{\rho}\left(\sum_i f_i \boldsymbol{c}_i + \frac{1}{2}\boldsymbol{F}\right) \tag{22}$$

Finally, we choose an initial density and velocity field at the start of the equation, as shown in the basic algorithm 2.4. However, the computation of $\boldsymbol{u}$ requires the force $\boldsymbol{F}$ to be incorporated. The velocity field that will be used during the first step is given by:

$$\boldsymbol{u} = \boldsymbol{u}_0 - \frac{\boldsymbol{F}}{2\rho} \tag{23}$$

2.6 Algorithm with sources

The algorithm for a single component with a sources, given by the Lattice Boltzmann Equation 3 with a source S_i, can be described as follows:

(i) Determine $\boldsymbol{F}(t = 0)$.

(ii) Choose $\rho(t = 0)$ and $\boldsymbol{u}_0$ and compute $\boldsymbol{u}$.

(iii) Initialize the populations as $\boldsymbol{f}(t = 0) = \boldsymbol{f}^{\text{eq}}(\rho, \boldsymbol{u})$.

1 Calculate $\boldsymbol{f}^{\text{eq}}(\rho, \boldsymbol{u})$.

2 Calculate source S_i.

3 Collide the populations $\rightarrow \boldsymbol{f}^*$.

4 Stream the populations $\rightarrow \boldsymbol{f}(t + 1)$.

5 Determine $\boldsymbol{F}(t + 1)$.

6 Calculate macroscopic quantities $\rho(t + 1)$ and $\boldsymbol{u}(t + 1)$.

7 $t = t + 1$ and return to step 1 while $t < t_{\text{final}}$.

2.7 Poiseuille flow

We can test the algorithm given in 2.6 by making use of a two-dimensional Poiseuille flow. We use periodic boundary conditions given in equation 9 in the x-direction and no-slip boundary conditions given in equation 10 in the y-direction. The pressure gradient applied on the fluid is simulated as a body force, such that $\boldsymbol{F}(\boldsymbol{x}, t) = F_p \hat{\boldsymbol{x}}$. The initial density and velocity field are set to unity and zero, respectively. The analytical solution for this Poiseuille flow at asymptotic time is given by:

$$u(y) = \frac{F_p}{2\nu}\left(d^2 - y^2\right) \tag{24}$$

Where we defined the radius of the pipe as $d = N_y/2$ using the number of cells N_y in the y-direction. Furthermore, $y_j = -d + 0.5 + j$. Using $N_x = 32$, $N_y = 128$ and $\tau = 1$, we get the result given in figure 2.

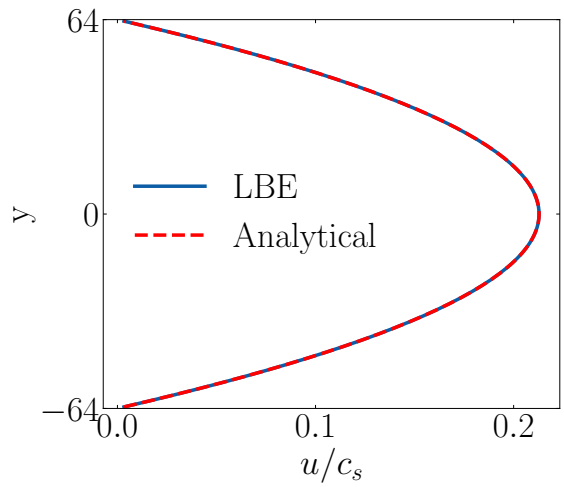


Figure 2: Flow profile of simulated Poiseuille flow at $t = 10^5$ compared to the analytical solution.

It is also possible to create a Poiseuille flow by using the fixed density boundary conditions for the inlet and outlet. For this, we replace F_p in equation 24 by the constant pressure gradient dp/dx. We initialize the system with this constant pressure gradient, using the proportionality between pressure and density. We impose a density of unity at the inlet, and a density of $1 - \frac{dp}{dx} N_x$ at the outlet. The result is given in figure 3. The result does not match up with the analytical solution, as the boundary conditions in the corners are not well defined. For now, we chose to apply the boundary the inlet/outlet boundary conditions at the corners.

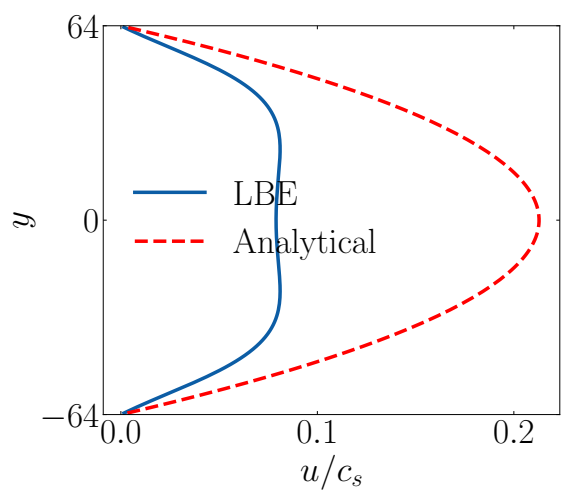


Figure 3: Flow profile of simulated Poiseuille flow at $t = 10^6$ compared to the analytical solution, using fixed density boundary conditions and the anti-bounce-back method.

Using the NEBB method, we get much better results, as shown in figure 4.

2.8 Multicomponent Shan-Chen method

To simulate a multicomponent fluid, we extend the Lattice Boltzmann Equation as follows:

$$f_i^\sigma(\boldsymbol{x} + \boldsymbol{c}_i, t+1) = f_i^\sigma(\boldsymbol{x}, t) - \frac{1}{\tau^\sigma}\Big(f_i^\sigma(\boldsymbol{x}, t) - f_i^{\mathrm{eq},\sigma}(\boldsymbol{x}, t)\Big) + S_i^\sigma \tag{25}$$

The fluids will exert a repelling force on each other, given by the Shan-Chen force:

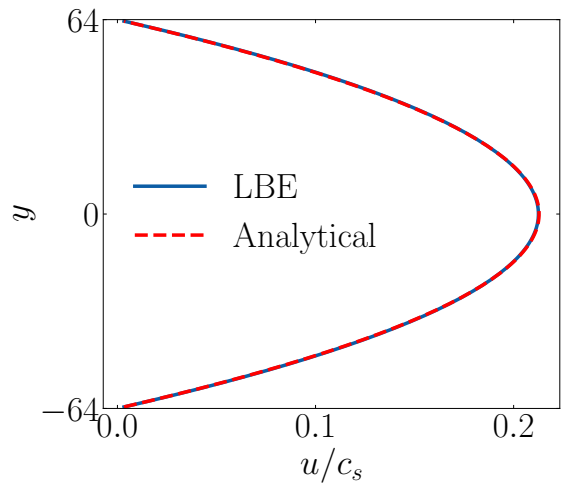


Figure 4: Flow profile of simulated Poiseuille flow at $t = 10^5$ compared to the analytical solution, using fixed density boundary conditions and the NEBB method.

$$\boldsymbol{F}^{\sigma}_{\mathrm{SC}}(\boldsymbol{x}) = -G\psi^{\sigma}(\boldsymbol{x}) \sum_i \left(w_i \psi^{\sigma^*}(\boldsymbol{x} + \boldsymbol{c}_i)\boldsymbol{c}_i \right) \tag{26}$$

Where we assume that the individual fluids are ideal, so the fluids don't exert forces on themselves. The interaction strength is given by G, which should be positive for the fluids to repel each other, making them partially immiscible. Note that σ^* indicates the other fluid component. The pseudopotential is given by ψ, which is most often given by:

$$\psi(\rho) = \rho_0 \left(1 - e^{-\rho/\rho_0}\right) \approx \rho \tag{27}$$

Where ρ_0 is the reference density, chosen by us to be unity in all simulations. The approximation is a first order Taylor expansion often used for the pseudopotential. The boundary conditions for the pressure in equation 26 are given by:

$$\begin{cases} \rho(\boldsymbol{x} + \boldsymbol{c}_i \mod \boldsymbol{L}), & \text{if periodic.} \\ \rho(\boldsymbol{x}), & \text{if no-slip.} \end{cases} \tag{28}$$

After computing the Shan-Chen force in equation 26, we can add this to the body force that we applied to find the total force on the individual fluid components. We need the individual densities and the total density to find the total force on the fluid components:

$$\begin{aligned} \rho^{\sigma} &= \sum_i f_i^{\sigma} \\ \rho &= \sum_{\sigma} \rho^{\sigma} \end{aligned} \tag{29}$$

The total force is computed as:

$$\boldsymbol{F}^{\sigma} = \boldsymbol{F}^{\sigma}_{\mathrm{SC}} + \frac{\rho^{\sigma}}{\rho}\boldsymbol{F}_p \tag{30}$$

The source term S_i^{σ} is calculated as:

$$S_i^{\sigma}(\boldsymbol{x}, t) = \left(1 - \frac{1}{2\tau^{\sigma}}\right) w_i \left(\frac{\boldsymbol{c}_i - \boldsymbol{u}(\boldsymbol{x}, t)}{c_s^2} + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}(\boldsymbol{x}, t)}{c_s^4}\boldsymbol{c}_i \right) \cdot \boldsymbol{F}^{\sigma}(\boldsymbol{x}, t) \tag{31}$$

The velocity of the total fluid is now computed as:

$$\boldsymbol{u} = \frac{1}{\rho} \sum_{\sigma} \left(\sum_{i} (f_i^{\sigma} \boldsymbol{c}_i) + \frac{\boldsymbol{F}^{\sigma}}{2} \right) \tag{32}$$

Where we use the same velocity for both fluid components. For the initial condition, we again choose an initial velocity field and take into account the forcing needed for the equilibrium distributions:

$$\boldsymbol{u} = \boldsymbol{u}_0 - \frac{1}{2\rho} \sum_{\sigma} \boldsymbol{F}^{\sigma} \tag{33}$$

2.9 Algorithm multicomponent Shan-Chen

The algorithm for a multicomponent fluid with a sources, given by the Lattice Boltzmann Equation 25

(i) Choose $\rho(t=0)$ and $\boldsymbol{u}_0$.

(ii) Determine $\boldsymbol{F}_{\mathrm{SC}}^{\sigma}(t=0)$ and $\boldsymbol{F}^{\sigma}(t=0)$.

(iii) Compute $\boldsymbol{u}$.

(iv) Initialize the populations as $\boldsymbol{f}^{\sigma}(t=0) = \boldsymbol{f}^{\mathrm{eq},\sigma}(\rho^{\sigma}, \boldsymbol{u})$.

1 Calculate $\boldsymbol{f}^{\mathrm{eq},\sigma}(\rho^{\sigma}, \boldsymbol{u})$.

2 Calculate source S_i^{σ}.

3 Collide the populations $\rightarrow \boldsymbol{f}^{*,\sigma}$.

4 Stream the populations $\rightarrow \boldsymbol{f}^{\sigma}(t+1)$.

5 Calculate $\rho(t+1)$.

6 Determine $\boldsymbol{F}_{\mathrm{SC}}^{\sigma}(t+1)$ and $\boldsymbol{F}^{\sigma}(t+1)$.

7 Calculate $\boldsymbol{u}(t+1)$.

8 $t = t+1$ and return to step 1 while $t < t_{\mathrm{final}}$.

2.10 Multicomponent Poiseuille flow

We can test the algorithm given in 2.9 by making use of a two-dimensional Multicomponent Poiseuille flow, where we have a high viscosity fluid at $|y| > a$ and a low viscosity fluid at $|y| < a$. The analytical solution is given by:

$$\begin{cases} u(y) = \frac{F_p}{2\nu_{\mathrm{high}}} \left(L_y^2 - y^2\right), & \text{for } |y| > a \\ u(y) = \frac{F_p}{2\nu_{\mathrm{high}}} \left(L_y^2 - a^2\right) + \frac{F_p}{2\nu_{\mathrm{low}}} \left(a^2 - y^2\right), & \text{for } |y| < a \end{cases} \tag{34}$$

We now choose $a = N_y/4$, $\nu_{\mathrm{high}} = 1/6$, $\nu_{\mathrm{high}} = 1/24$ and $G = 4$. This gives the result given in figure 5. The velocity field is underpredicted with respect to the analyitical solution. This can be explained by the fact that the density outside of the interaction zones of the two fluids is larger than unity, to compensate the reduced mass in the interaction zones. The actual solution for the two-component Poiseuille flow contains the dynamic viscosity $\mu = \rho\nu$, which is not equal to the kinematic viscosity ν if the density is not unity.

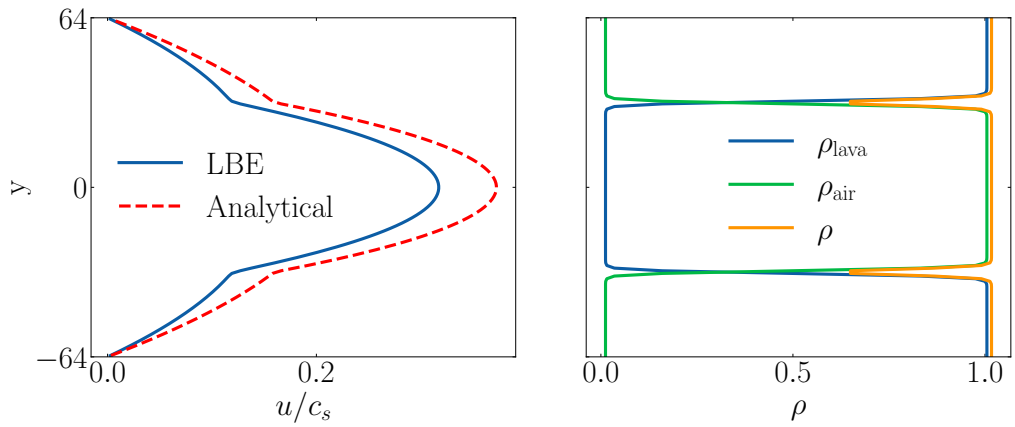


Figure 5: Flow profile of simulated multicomponent Poiseuille flow at $t = 10^5$ compared to the analytical solution.

2.11 Temperature

The advection-diffusion equation for the temperature field is solved by introducing another distribution function $\boldsymbol{g}$, which gives the temperature as:

$$T = \sum_i g_i \tag{35}$$

The distribution function g evolves according to:

$$g_i(\boldsymbol{x} + \boldsymbol{c}_i, t+1) = g_i(\boldsymbol{x}, t) - \frac{1}{\tau_g}\Big(g_i(\boldsymbol{x}, t) - g_i^{\text{eq}}(\boldsymbol{x}, t)\Big) \tag{36}$$

Where the relaxation time is computed as:

$$\tau_g = \frac{1}{\rho}\sum_\sigma \rho^\sigma \tau_g^\sigma \tag{37}$$

Which gives the thermal diffusivity at a specific position in space as:

$$\kappa = c_s^2\left(\tau_g - \frac{1}{2}\right) \tag{38}$$

And the equilibrium distribution $\boldsymbol{g}^{\text{eq}}$ is given by:

$$g_i^{\text{eq}} = w_i T\left(1 + \frac{\boldsymbol{u}\cdot\boldsymbol{c}_i}{c_s^2} + \frac{(\boldsymbol{u}\cdot\boldsymbol{c}_i)^2}{2c_s^4} - \frac{|\boldsymbol{u}|^2}{2c_s^2}\right) \tag{39}$$

The effect of the temperature field on the velocity populations comes from the buoyancy force:

$$\boldsymbol{F}_b = -\alpha\rho_0\left(T - T_0\right)\boldsymbol{g} \tag{40}$$

Where α is the thermal expansion coefficient, T_0 is the reference temperature and ρ_0 is the density where the temperature is given by T_0. Note that the thermal expansion coefficient α can itself by heavily dependent on the temperature T. Lastly, we need to discuss the boundary conditions for the thermal populations g. The periodic boundary conditions are implemented in the same way as described in equation 9. For the no-slip boundaries, we make use of the anti-bounce-back scheme:

$$g_i(\boldsymbol{x}, t+1) = -g_q^*(\boldsymbol{x}, t) + 2w_q T_{\text{wall}} \tag{41}$$

Where the relation between the indices i and q is again given by equation 11.

2.12 Algorithm thermal Lattice Boltzmann

The algorithm for a thermal fluid is given by:

(i) Choose $\rho(t=0)$, $\boldsymbol{u}_0$ and $T(t=0)$.

(ii) Determine $\boldsymbol{F}_b(t=0)$.

(iii) Compute $\boldsymbol{u}$.

(iv) Initialize the populations as $\boldsymbol{f}(t=0) = \boldsymbol{f}^{\mathrm{eq}}(\rho, \boldsymbol{u})$.

(v) Initialize the thermal populations as $\boldsymbol{g}(t=0) = \boldsymbol{g}^{\mathrm{eq}}(T, \boldsymbol{u})$.

1 Calculate $\boldsymbol{g}^{\mathrm{eq}}(T, \boldsymbol{u})$.

2 Collide the thermal populations $\rightarrow \boldsymbol{g}^*$.

3 Stream the thermal populations $\rightarrow \boldsymbol{g}(t+1)$.

4 Calculate $T(t+1)$.

5 Determine $\boldsymbol{F}_b(t+1)$.

6 Calculate $\boldsymbol{f}^{\mathrm{eq}}(\rho, \boldsymbol{u})$.

7 Calculate source S_i.

8 Collide the populations $\rightarrow \boldsymbol{f}^*$.

9 Stream the populations $\rightarrow \boldsymbol{f}(t+1)$.

10 Calculate $\rho(t+1)$.

11 Calculate $\boldsymbol{u}(t+1)$.

12 $t = t+1$ and return to step 1 while $t < t_{\mathrm{final}}$.

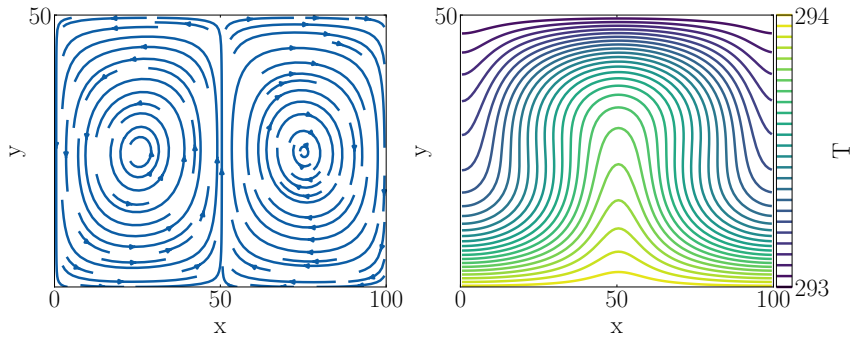
2.13 Rayleigh-Bénard convection

We want to simulate a test case of Rayleigh-Bénard convection in a periodic channel [1]. For this test case, we will go more deeply into the specifics of the unit conversion between real and lattice units. The chosen real length scale will be given by the height of the channel, $\widetilde{L} = \widetilde{L}_y = 1$ m. We use the buoyant velocity as the velocity scale, given by $\widetilde{U} = \sqrt{\widetilde{\alpha}\Delta\widetilde{T}\,\widetilde{g}\widetilde{L}}$. Notice that $\widetilde{\alpha}\Delta\widetilde{T}$ is a nondimensional quantitiy already. For the expansion coefficient, we arbitrarily choose $\widetilde{\alpha} = 2.07 \cdot 10^{-4}$ m^2/s. We set the bottom of the channel to $\widetilde{T}_{\mathrm{bottom}} = 294$ K and the top to $\widetilde{T}_{\mathrm{top}} = 293$ K. The units of the temperature in the real case and in the lattice will be the same. The velocity scale is computed to be $\widetilde{U} \approx 4.5 \cdot 10^{-2}$ m/s. We will also choose the following dimensionless quantities:

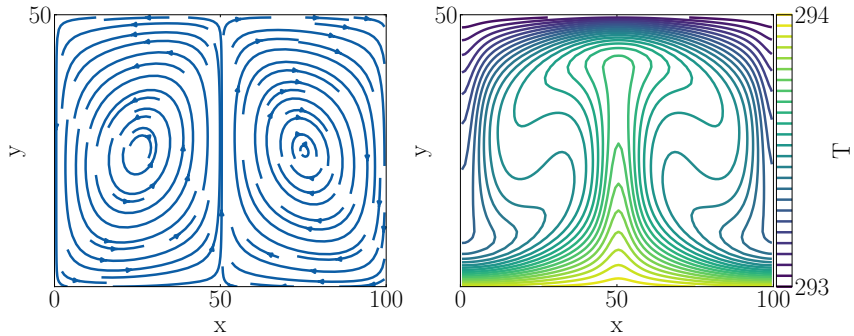
$$Ra = \frac{\widetilde{U}^2\widetilde{L}^2}{\nu\kappa}, \quad Pr = \frac{\nu}{\kappa} \tag{42}$$

We will choose $Pr = 0.71$, and we will vary the Rayleigh number. We will now need the relaxation times and gravitational acceleration in lattice units. To find those, we compute the conversion factors. The spatial conversion factor is computed as $C_l = \Delta\widetilde{x}/\Delta x = \Delta\widetilde{x}$ and the velocity conversion factor is given by $C_u = \widetilde{U}/U$, where we chose the velocity scale in lattice units as $U = 0.01$, such that it is much smaller than the speed of sound $c_s = \sqrt{1/3}$. The conversion factor for the diffusivities is then given by $C_\nu = C_l C_u$, and the conversion factor for acceleration is given by $C_a = C_u^2/C_l$.

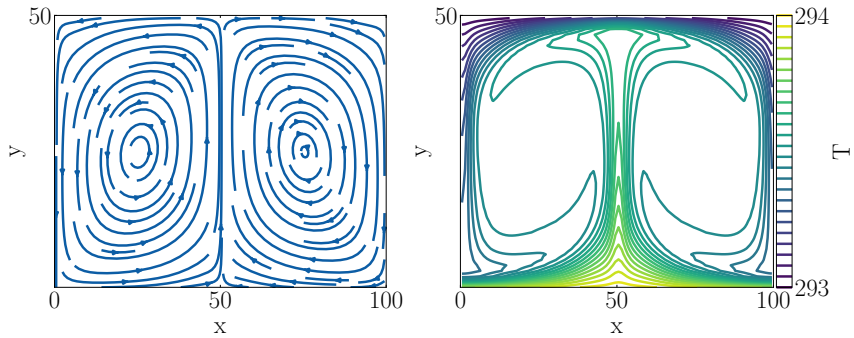
For the grid, we choose $N_x = 100$ and $N_y = 50$, which gives $g \approx 9.7 \cdot 10^{-3}$. We do simulations for $Ra = 5000$, $Ra = 50000$ and $Ra = 10^6$, where the resulting streamlines of the velocity field and contours for the temperature fields are shown in figure 6.



(a) $Ra = 5000$



(b) $Ra = 50000$



(c) $Ra = 10^6$

Figure 6

2.14 Phase change

We want to be able to have a change of phase between liquid and solid. This is done by making use of the Enthalpy:

$$H(t) = cT(t) + L_f \phi_l(t-1) \tag{43}$$

Where H is the enthalpy, c is the heat capacity, L_f is the latent heat and ϕ_l is the fraction of liquid on a grid point. The heat capacity of the solid and liquid phase differs, so the term $cT(t)$ can be changed to:

$$cT(t) = (1 - \phi_l(t-1))\, c_{\text{solid}} T + \phi_l(t-1)\Big(c_{\text{liquid}}\,(T - T_m) + c_{\text{solid}} T_m\Big) \tag{44}$$

The liquid fraction at time t can now be computed as:

$$\phi_l(t) = \begin{cases} 0, & \text{for } H(t) < H_s = c_{\text{solid}} T_m \\ \frac{H(t) - H_s}{L_f}, & \text{for } H_s \leq H(t) \leq H_s + L_f \\ 1, & \text{for } H(t) > H_s + L_f \end{cases} \tag{45}$$

Where T_m is the melting temperature. The thermal populations at time $t+1$ can then be computed by using equation 36 with an added source:

$$g_i(\boldsymbol{x} + \boldsymbol{c}_i, t+1) = g_i(\boldsymbol{x}, t) - \frac{1}{\tau_g}\Big(g_i(\boldsymbol{x}, t) - g_i^{\text{eq}}(\boldsymbol{x}, t)\Big) - w_i \frac{L_f}{c_{\text{solid}}}\Big(\phi_l(\boldsymbol{x}, t) - \phi_l(\boldsymbol{x}, t-1)\Big) \tag{46}$$

To make sure that the solid parts of the fluid stay stationary, we add a penalization force depending on the solid fraction:

$$\boldsymbol{F}_{\text{pen}} = -\left(1 - \phi_l^2\right) \rho \boldsymbol{u} \tag{47}$$

For the initialization fo the liquid fraction, we choose $\phi_l(t=0)$ and set $\phi_l(t=-1) = \phi_l(t=0)$. Finally, we have different thermal diffusivities for the solid and liquid phase:

$$\tau_g = \phi_l \tau_g^{\text{liquid}} + (1 - \phi_l)\, \tau_g^{\text{solid}} \tag{48}$$

2.15 Algorithm phase change

The algorithm for a fluid with phase change is given by:

(i) Choose $\rho(t=0)$, $\boldsymbol{u}_0$, $T(t=0)$ and $\phi_l(t=0)$.

(ii) Set $\phi_l(t=-1) = \phi_l(t=0)$.

(iii) Determine $\boldsymbol{F}_b(t=0)$ and $\boldsymbol{F}_{\text{pen}}(t=0)$.

(iv) Compute $\boldsymbol{u}$.

(v) Initialize the populations as $\boldsymbol{f}(t=0) = \boldsymbol{f}^{\text{eq}}(\rho, \boldsymbol{u})$.

(vi) Initialize the thermal populations as $\boldsymbol{g}(t=0) = \boldsymbol{g}^{\text{eq}}(T, \boldsymbol{u})$.

1 Calculate $\boldsymbol{g}^{\text{eq}}(T, \boldsymbol{u})$.

2 Collide the thermal populations $\rightarrow \boldsymbol{g}^*$.

3 Stream the thermal populations $\rightarrow \boldsymbol{g}(t+1)$.

4 Calculate $T(t+1)$.

5 Determine $\boldsymbol{F}_b(t+1)$ and $\boldsymbol{F}_{\text{pen}}(t+1)$.

6 Calculate $\boldsymbol{f}^{\text{eq}}(\rho, \boldsymbol{u})$.

7 Calculate source S_i.

8 Collide the populations $\rightarrow \boldsymbol{f}^*$.

9 Stream the populations $\rightarrow \boldsymbol{f}(t+1)$.

10 Calculate $\rho(t+1)$.

11 Calculate $\boldsymbol{u}(t+1)$.

12 Compute $H(t+1)$ and $\phi_l(t+1)$.

13 $t = t+1$ and return to step 1 while $t < t_{\text{final}}$.

2.16 Stefan problem

To test the solidification, we apply the algorithm in 2.15 to a stationary flow where we have a cold plate on the bottom below the melting temperature. This means that the interface of solidified fluid grows upwards. We define the Stefan number as:

$$St = \frac{c\Delta T}{L_f} \tag{49}$$

The position of the interface over time is given by:

$$y_i(t) = 2\lambda\sqrt{\kappa t} \tag{50}$$

Where the constant λ depends on the Stefan number as:

$$\lambda e^{\lambda^2} \text{erf}(\lambda) = \frac{St}{\sqrt{\pi}} \tag{51}$$

We choose $c_p^{\text{liquid}} = c_p^{\text{solid}} = 0.95$, $L_f = 1$, $St = 1$, $T_m = 0$, $T_{\text{bottom}} = -St/c_p^{\text{solid}}$, $T_{\text{top}} = T_m$ and $\kappa^{\text{liquid}} = \kappa^{\text{solid}} = 0.00166$. We solve equation 51 by using a rootfinding algorithm. The result is given in figure 7.

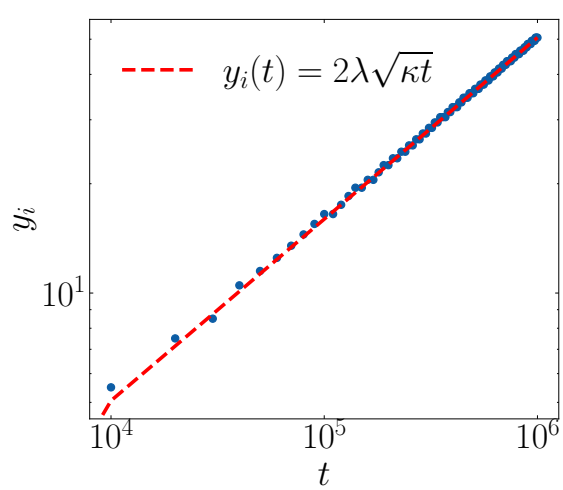


Figure 7: Position of interface over time, compared to the analytical solution.

2.17 Phase change in two-component fluid

Only the lava can change phase. We use one distribution function for the temperature, where the relaxation time is computed as:

$$\tau_g = \frac{\rho_{\text{lava}}}{\rho}\left(\phi_{l,\text{lava}}\tau_{g,\text{lava}}^{\text{liquid}} + (1-\phi_{l,\text{lava}})\,\tau_{g,\text{lava}}^{\text{solid}}\right) + \frac{\rho_{\text{air}}}{\rho}\tau_{g,\text{air}} \tag{52}$$

2.18 Algorithm phase change in two-component fluid

We define the following:

$$\hat{\boldsymbol{u}}(t) = \boldsymbol{u}(t) - \frac{1}{2\rho(t)}\sum_{\sigma}\boldsymbol{F}^{\sigma}(t) \tag{53}$$

$$\boldsymbol{u}^*(t+1) = \frac{1}{\rho(t+1)}\sum_{\sigma}\sum_{i} f_i^{\sigma}(t+1)\boldsymbol{c}_i \tag{54}$$

$$\boldsymbol{u}(t+1) = \boldsymbol{u}^*(t+1) + \frac{1}{2\rho(t+1)}\sum_{\sigma}\boldsymbol{F}^{\sigma}(t+1) \tag{55}$$

Furthermore, the Buoyancy force in a two-component fluid is given by:

$$\begin{aligned}\boldsymbol{F}_{\text{buoy}} &= (\rho(T)-\rho_0)\,\boldsymbol{g}\\ &\approx \left(\sum_{\sigma}\left(\rho_0^{\sigma}\left(1-\alpha^{\sigma}\left(T-T_0^{\sigma}\right)\right)\right) - \sum_{\sigma}\rho_0^{\sigma}\right)\boldsymbol{g}\\ &= -\sum_{\sigma}\rho_0^{\sigma}\alpha^{\sigma}\left(T-T_0^{\sigma}\right)\boldsymbol{g}\end{aligned} \tag{56}$$

With $\rho_0 \equiv \rho(T_0)$. The component forces are computed by multiplying this force with the density fraction. Because only the lava can experience phase change, there will only be a force on the lava component due to penalization:

$$F_{\text{pen}}^{\text{lava}} = -\left(1-\left(\phi_l^{\text{lava}}\right)^2\right)\rho^{\text{lava}}\boldsymbol{u} \tag{57}$$

Furthermore, for the source in the thermal collision step, we multiply it by $\frac{\rho^{\text{lava}}}{\rho}$. We can define a general algorithm with 2 components, where one can change phase, as:

(i) Choose $\rho^{\text{lava}}(t_0)$, $\rho^{\text{air}}(t_0)$, $\boldsymbol{u}(t_0)$, $T(t_0)$ and $\phi_l^{\text{lava}}(t_0)$,

(ii) $\rho(t_0) = \rho^{\text{lava}}(t_0) + \rho^{\text{air}}(t_0)$,

(iii) $\phi_l^{\text{lava}}(t_0 - 1) = \phi_l^{\text{lava}}(t_0)$,

(iv) Compute $\boldsymbol{F}^{\sigma}_{\text{body}}(t_0)$ using $\rho^{\sigma}(t_0)$.

(v) Compute $\boldsymbol{F}^{\sigma}_{\text{buoy}}(t_0)$ using $\rho^{\sigma}(t_0)$ and $T(t_0)$.

(vi) Compute $\boldsymbol{F}^{\sigma}_{\text{SC}}(t_0)$ using $\rho^{\sigma}(t_0)$.

(vii) $\boldsymbol{f}^{\sigma}(t_0) = \boldsymbol{f}^{\text{eq},\sigma}\big(\rho^{\sigma}(t_0), \hat{\boldsymbol{u}}(t_0)\big)$.

(viii) $\boldsymbol{g}(t_0) = \boldsymbol{g}^{\text{eq}}\big(T(t_0), \hat{\boldsymbol{u}}(t_0)\big)$

(ix) $t = t_0$.

1. Finalize when $t = t_{\text{final}}$.
2. Compute $\boldsymbol{g}^{\text{eq}}(t)$ using $T(t)$ and $\boldsymbol{u}(t)$.
3. Collide $\boldsymbol{g}^{*}(t)$ using $\boldsymbol{g}(t)$, $\boldsymbol{g}^{\text{eq}}(t)$, $\rho(t)$, $\phi_l^{\text{lava}}(t-1)$ and $\phi_l^{\text{lava}}(t)$.
4. Stream $\boldsymbol{g}(t+1)$ using $\boldsymbol{g}^{*}(t)$ and $\phi_l^{\text{lava}}(t)$.
5. Compute $T(t+1)$ using $\boldsymbol{g}(t+1)$.
6. Compute $\boldsymbol{f}^{\text{eq},\sigma}(t)$ using $\rho^{\sigma}(t)$ and $\boldsymbol{u}(t)$.
7. Compute $S_i^{\sigma}(t)$ using $\boldsymbol{u}(t)$ and $\boldsymbol{F}^{\sigma}(t)$.
8. Collide $\boldsymbol{f}^{*,\sigma}(t)$ using $\boldsymbol{f}^{\sigma}(t)$, $\boldsymbol{f}^{\text{eq},\sigma}(t)$, $\rho(t)$ and $S_i^{\sigma}(t)$.
9. Stream $\boldsymbol{f}^{\sigma}(t+1)$ using $\boldsymbol{f}^{*,\sigma}(t)$.
10. Compute $\rho^{\sigma}(t+1)$, $\rho(t+1)$ and $\boldsymbol{u}^{*}(t+1)$ using $\boldsymbol{f}^{\sigma}(t+1)$.
11. Compute $\boldsymbol{F}^{\sigma}_{\text{body}}(t+1)$ using $\rho^{\sigma}(t+1)$.
12. Compute $\boldsymbol{F}^{\sigma}_{\text{buoy}}(t+1)$ using $\rho^{\sigma}(t+1)$ and $T(t+1)$.
13. Compute $\boldsymbol{F}^{\sigma}_{\text{SC}}(t+1)$ using $\rho^{\sigma}(t+1)$.
14. Compute $\boldsymbol{u}(t+1)$ using $\boldsymbol{u}^{*}(t+1)$, $\rho(t+1)$ and $\boldsymbol{F}^{\sigma}(t+1)$.
15. Compute $H^{\text{lava}}(t+1)$ using $T(t+1)$, $\phi_l^{\text{lava}}(t)$.
16. Compute $\phi_l^{\text{lava}}(t+1)$ using $H^{\text{lava}}(t+1)$.
17. $t \to t+1$, return to step 1.

3 Results

References

[1] Y. Wei, Z. Wang, J. Yang, H.-S. Dou, and Y. Qian, Computers & Fluids **118**, 167 (2015).

4 Appendix