MULTIPHASE MODELS

A. Kuzmin

Department of Chemical and Materials Engineering
University of Alberta
Canada

August 24,2011 / LBM Workshop

SHAN-CHEN MODEL

Implement the Shan-Chen model in Matlab:

EXERCISE 1

- Implement the Shan-Chen force:

$$\boldsymbol{F}(\boldsymbol{x}) = -G\psi(\boldsymbol{x}) \sum_i w_i \psi(\boldsymbol{x} + \boldsymbol{c_i})\boldsymbol{c_i}$$

- Obtain the decomposition of the critical fluid on liquid and gas. Initialize the density with the critical density perturbed by the random number $\rho = \rho_{crit} + 0.1rand()$. The temperature-like parameter G is $G = -5.0$.
- Use the C++ code provided as far as matlab simulation takes a quite long time. One needs to check the Laplace law:

$$\Delta P = \frac{\sigma}{R},$$

where σ is the surface tension. The file is "shanchen_ex1.cpp". We imposed $G = -5.0$ with the initial densities $\rho_{liq} = 1.95$ and $\rho_{gas} = 0.15$. Perform simulations with $R = 15$, $R = 20$, $R = 25$, $R = 30$ (Hint: change line 26 in the code). For those who didn't use the Linux C++ compiler use "g++ -O3 shanchen_ex1.cpp -o main.out" to compile and "./main.out" to run the program. The output of the program will give you ΔP. Visualize ΔP vs. $1/R$ with Matlab.

Gas and liquid separation

The purpose of the following exercise is to show that the simulation gas and liquid densities versus G are almost the same as calculated ones (File: shanchen_ex2.m).

Exercise 2

1. Open the "ghandl.jpg" to see dependancy of the gas and liquid densities on G. Notice that increasing G increases the gas-liquid separation.
2. Use different G parameters:
   - $G = -5.0$,$\rho_{liq} = 1.9$, $\rho_{gas} = 0.15$. Check the equilibrium densities.
   - $G = -6.0$, $\rho_{liq} = 1.9$, $\rho_{gas} = 0.15$. See that it is unstable.
   - $G = -6.0$, $\rho_{liq} = 2.7$, $\rho_{gas} = 0.05$. See the stable equilibrium densities.
   - Try to obtain stable situation by changing ρ_{liq} and ρ_{gas} with $G = -8.0$.

The message: Need to play with parameters to obtain a stable situation.

Advanced exercise

Look at the provided examples with walls. Try to change the contact angle by playing with the wall density. There are two files provided: “shanchen_walls.m” and “shanchen_walls.cpp”. One can visualize results of C++ program with “visualize.m” Matlab script.

BINARY-LIQUID MODEL

We implement the binary-liquid model in the most spread way by including the force inside the eqilibrium function:

$$
\begin{aligned}
f_i^{eq} = \quad & w_i\rho\Big(3\big(\frac{\rho}{3} + A(-\frac{\phi^2}{2} + \frac{3}{4}\phi^4) - k\phi\Delta\phi\big) + 3c_{i\alpha}u_\alpha^{eq} \\
& + \frac{9}{2}\big(c_{i\alpha}c_{i\beta} - \frac{1}{3}\big)u_\alpha^{eq}u_\beta^{eq}\Big) + k\big(w_i^{xx}(\partial_x\phi)^2 + w_i^{yy}(\partial_y\phi)^2 \\
& + w_i^{xy}\partial_x\phi\partial_y\phi\big), \text{ for } 1 \leq i \leq 8 \\
f_0^{eq} = \quad & \rho - \sum_{i=1} 8 f_i^{eq}
\end{aligned}
$$

where additional weights w_i^{xx},w_i^{xy} and w_i^{yy} are responsible for the distribution of the forcing through all equilibrium functions:

$$
\begin{aligned}
w_i^{xx} &= \{0, \frac{1}{3}, -\frac{1}{6}, \frac{1}{3}, -\frac{1}{6}, -\frac{1}{24}, -\frac{1}{24}, -\frac{1}{24}, -\frac{1}{24}\} \\
w_i^{yy} &= \{0, -\frac{1}{6}, \frac{1}{3}, -\frac{1}{6}, \frac{1}{3}, -\frac{1}{24}, -\frac{1}{24}, -\frac{1}{24}, -\frac{1}{24}\} \\
w_i^{xy} &= \{0, 0, 0, 0, 0, \frac{1}{4}, -\frac{1}{4}, \frac{1}{4}, -\frac{1}{4}\}
\end{aligned}
$$

PHASE EQUILIBRIUM DISTRIBUTION FUNCTION

Phase equilibrium functions are:

$$g_i = w_i\Big(3\Gamma\mu + \phi 3u_\alpha c_{i\alpha} + \phi\big(c_{i\alpha}c_{i\beta} - \frac{1}{3}\big)\Big), \text{ for } 1 \le i \le 8$$

$$g_0 = \phi - \sum_{i=1}^{8} g_i,$$

where the chemical potential $\mu = \Gamma(A(-\phi + \phi^3) - k\Delta\phi)$.

$$\Delta = \begin{bmatrix} \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \\ \frac{4}{6} & \frac{-20}{6} & \frac{4}{6} \\ \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \end{bmatrix}; \quad \partial_x = \begin{bmatrix} -\frac{1}{12} & 0 & \frac{1}{12} \\ -\frac{4}{12} & 0 & \frac{4}{12} \\ -\frac{1}{12} & 0 & \frac{1}{12} \end{bmatrix}; \quad \partial_y = \begin{bmatrix} \frac{1}{12} & \frac{4}{12} & \frac{1}{12} \\ 0 & 0 & 0 \\ -\frac{1}{12} & -\frac{4}{12} & -\frac{1}{12} \end{bmatrix}$$

Exercise 1

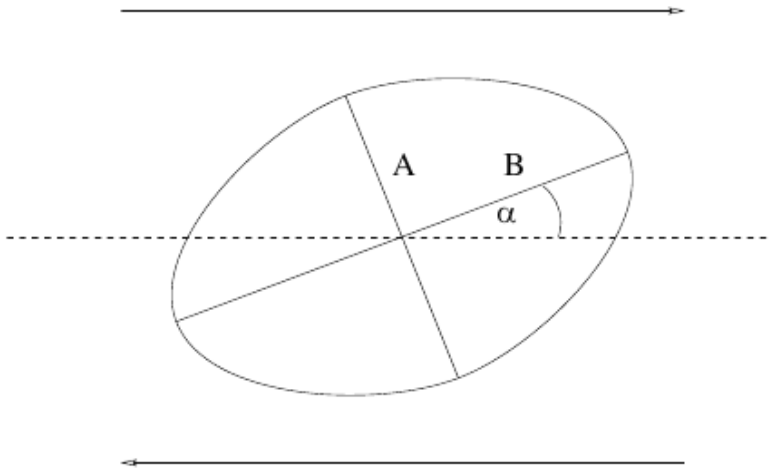
Code the binary-liquid model:

Exercise 1

1. Look at the implementation of the initialization. It contains all the necessary elements for the equilibrium distribution functions. Especially is important to understand the change of the equilibrium functions and calculation of the laplacians.
2. Notice that the calculation of the macroscopic quantities are taken out of the main loop? Why is that?
3. Transfer this part of the code to the main body.
4. Change τ_ρ with the function $\tau_\rho(\phi, \tau_{gas}, \tau_{liq})$.
5. Obtain the stable picture

SHEAR FLOW

Because Matlab is a pretty slow solver, we coded the next physical example in the C++ code. Please take a look at the provided file “binary_ex2.cpp”. The purpose of this example is to show that we simulate a real physical system. When two walls are moving with certain velocity u_{wall} then the droplet in the center deforms into the ellipse. The ellipse has two axes a and b. The deformation of the ellipse $D = \frac{A-B}{A+B}$ (figure of courtesy A. Wagner, et.al) is proportional to the capillary number $Ca \propto u_{wall}$.



Exercise 2

Moving walls and droplet exercise:

Exercise 2

- Look at the implementation of the walls. The wall phases are prescribed by the wall gradient, which is responsible for the contact angle. For simplicity we take the wall gradient as zero (lines 241-248)
- Look at the function “update_bounce_back” which updates the walls populations. Notice that the bounce-back is able to prescribe velocity boundary conditions as well for the density probability distribution function.
- Compile with g++: “g++ -O3 binary_ex2.cpp -o main.out”. Run the example “./main.out” and visualize it with Matlab (file: ellipse.m). The file will give you as well the properties of the ellipse.
- Examine velocities $u_{wall} = 0.05$, $u_{wall} = 0.075$,$u_{wall} = 0.1$ (line 23).
- Plot deformation parameter D versus the wall velocity u_{wall}. See the linear dependence?

CONTACT ANGLE

This example is to show that the system captures the contact angle within 5 degrees accuracy. As we indicated in the lecture the contact angle is simulated by the change of the wall gradient near the wall $\partial_\perp \phi$. The contact angle of the surface can be predicted by solving the equation:

$$\partial_\perp \phi = \sqrt{\frac{2A}{k}} \sqrt{\cos\left(\frac{\alpha}{3}\right)\left(1 - \cos\left(\frac{\alpha}{3}\right)\right)},$$

where parameter α is connected with the contact angle $\alpha = \cos^{-1}(\sin^2(\theta))$.

EXERCISE 3

- The simulations are conducted through the code "binary_ex3.cpp". Please open it and see how the simulation is done. The wall gradient is involved in lines 246-250.
- Please compile it and run with the wall gradient $\partial_\perp = -0.05, -0.1, -0.15$.
- Examine results with the provided Matlab script "contactangle.m" which calculates the analytical angle and extracts the simulation angle. Compare them.