

Weekly Reports

Mehmani Research Group

Nicolás Bueno Zapata

Spring 2022

Pennsylvania State University

Jan 30, 2022

During this week I was focused on testing my LBM code for a single component and multiphase conditions. For validation, I looked for other available codes from the postdoc, Cheng, and codes available in the Dr. Ayala's research group. I had to redefine some objects in my code to be able to pursue the next modeling objectives as:

- Have different relaxation parameters for every component
- Have more than one force applying to our components
- The components also have the property "pseudopotential"

I was reading how to model two immiscible components through LBM, and I discover that the treatment of Cheng is quite different (valid, though) from the proposed in Kruger's book. I created a new routine with a different Shan-Chen force implementation (each component uses its own pseudopotential) for immiscible displacement, because I want to be able to simulate a case where I inject a single component that displaces the other, although I just realize that I need to specify the composition of my injecting fluid for all the boundary conditions we have specified. In single component, specifying the pressure is equivalent to specify the density. However, for multi-component, we need to provide the amount of each component at that pressure to be able to have a solvable system.

I decided to keep track of all the different options and equations inside the code, and I started to write my own LBM document to explain to myself and future users, what are the equations implemented and the logic behind them.

It has been difficult to balance the time for research, coding, and reading papers where to inspire for research ideas and applications. I am still trying to find the correct point in this semester, as all the courses are kind of related to my thesis, and all deserve the same degree of attention.

Feb 6, 2022

Things that were done

This week started the full implementation of the multi-relaxation-times collision operator for my LBM model. Despite the first coding stage, I had programmed classes for collisionMRT operator and the lbmParameter object, these two were somehow in conflict, as the Fortran polymorphism is somehow more strict than in C++, so I had to arrange the code to account for generalization in the main routines, who do not know in principle, which type of object are they calling to. I read about LAPACK, a library to handle linear systems in Fortran, that could be useful for other problems and applications using the code in development.

Another advance in this matter was the two-phase flash algorithm that is almost working in C++. The purpose of this code is to have my own library for thermodynamic analysis of mixtures (solve stability, n-phase flash, envelopes, bubble points) that can give insights in how to set a simulation case for the LBM code. The novelty with this code is that is connected with Matplotlib, library from Python, so it can be used any plot command from there in C++. I would like to improve the Subsequent Substitution Method by a Newtonian algorithm using the C++ Library, Eigen, to solve linear systems of equations.

Difficulties

This semester, as particularly intense, has been difficult to balance between study and working times. However, I have managed to take advantage of the homework and classes to enforce my knowledge, especially that related with my field of research and expertise: thermodynamic behavior of mixtures, pore-scale phenomena, scientific-programming, and fluid-flow modeling. I also have come across challenges regarding the object-oriented programming with Fortran. There are plenty resources online, but hardly there is one canonical source as there is for C++.

Work for next week

I will validate the MRT collision operator with the Cheng's codes for single component. After this, I will validate for two components. If I reach this point, I will be able to write the paper that he left in progress, and I can focus on particular problems (applications) and the parallelization with OpenMPI. I will start selecting some papers to delimit the

fields where this multiphase model could be applied, as I need to have a goal for the method, so I can take some decisions in advance. As capillary pressure is of our interest, it is worth asking if the 3D implementation is a priority in the research.

Ask for access to the report template.

Feb 7 - Feb 12

What was done last week

This week I finished the coding of the multi-relaxation times (MRT) collision operator. As this operator implies the multiplication of matrices and vectors, I used the Fortran built-in operator for this purpose, although later on a manually multiplication can be coded to gain computational speed (that is not a priority right now, but I keep saving notes for future improvements). At this point, it is worth mentioning the simulations I am running make use of an cubic equation of state, that was validated months ago, before taking a detour to validate single phase problems with analytical solutions, that I presented before going to the Winter break. Now, the validation cases combine single or multi-component cases with BGK or MRT as the collision operator. The results of the previous BGK implementation show a successful match between my model and the Cheng's legacy code (Dr. Ayala's post-doc), for static and oscillating droplets of a single component/multiphase model. The MRT model was successfully implemented, and the static droplet case was validated (Figure 1). However, the case for the oscillating droplet has presented difficulties in terms of model interpretability, and results availability (see difficulties). During the week, I wanted to clarify some ideas that were taking me to different directions. As I am conscious about how important it is to focus on a single task, I tried to discriminate between interesting ideas to pursue later, and the important research aspects that have to be addressed from now. So, apart from coding, I made a preliminary review about current multiphase applications where LBM is a suitable model. I found an interesting topic, called self-propelled droplets, where acceleration is exerted by surface tension gradients caused by non-uniform fields of whether surfactant concentration or temperature. This search is obeying my believe on that concrete applications help to visualize and give shape to the current and future code, at the time it motivates me to be in current hot-spots for research (I want to be close to topics of high applicability). In that sense, I discovered the Marangoni-like flow, I found some papers simulating the phenomena with LBM, and I plan to read a couple during the week and possibly bringing them for the next group meeting.

What will be done next week?

I will finish the MRT validation against Cheng's codes, but this paradigm is becoming more complex every time, as the codes are extensive and each one changes its input parameters. I feel that attach to simpler but extensively reported cases would be better

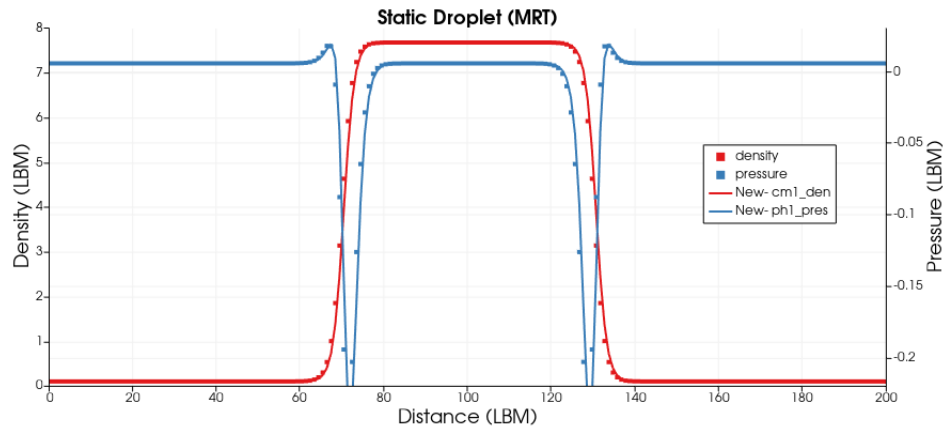


Figure 1: Comparison of pressure and density.

for our purpose. As the MRT code is already running and giving good results (qualitatively), I consider the two-component case is ready to be addressed. The code is already generalized, so the only missing part is to configure the cases and start running and debugging.

Difficulties

- The MRT case does not show several oscillations as the BGK does.
- The BGK has only one parameter that relates to the viscosity of the fluid, so I can control how fast the momentum diffuses through this parameter. However, MRT has more parameters, and sometimes they look like *magic-parameters*, that I am still understanding to produce stable simulations.
- There are many validation codes, and handling that amount of versions is becoming overwhelming. How can I generate enough trust in the model?
- Side note: The phase behavior model I have in C++ started as a class project the last semester for the Advance Programming course, and is the mandatory project for the Thermodynamics course I am taking this semester. That is the reason I am developing the code, and the fact it is in C++ is because I want to improve my proficiency with high-performance programming languages.