Lattice Boltzmann program

# 1. Project aim and participants

We have been using open source library OpenLB to carry out fluid simulation. We found that there are some limitations when simulating multiphase flows: a) Velocity and pressure boundaries only support for one specific fluid. b) program not working properly when simulating more than 2 phase fluids. This project aims to create a LBM program which has better strength in simulating multiphase problems.

**Participants**:

Mr Suo Si, PhD student at the University of Sydney

Mr Zhongzheng Wang, PhD student at the University of Sydney

2. Features

We chose the following features as an initial starting point. More features (with “\*”, or not mentioned) can be implemented in the future

**Lattice Boltzmann models**

* BGK models for fluids
* Multiple relaxation time (MRT)\*
* BGK with Smagorinsky model\*
* MRT with Smagorinsky model\*

**Multiphysics Coupling**

* Entropy
* Shan-Chen two-component fluid

**Lattice structures**

* D2Q9
* D3Q19\*

**Boundary conditions (for straight boundaries)**

* Regularized
* Zou/He\*

3. Program structure

**1st Step: Initialization** The converter between physical and lattice is set. Define where the simulation data is stored and which lattice type is used.

**2nd Step: Prepare Geometry** The geometry is defined. mesh is created and initialized based on geometry, which includes classifying voxels with material numbers. Further, the mesh is distributed over the threads to establish good scaling properties\*.

**3rd Step: Prepare Lattice** According to the material number, the lattice dynamics are set. This step characterizes the collision model and boundary behaviour. (the choice depends on whether a force is acting or not, the use of BGK or MRT, simulation dimension, compressible or not, and the number of voxels chosen). The computing grid, the SuperLattice, is created. The allocation of the required data is done as well.

**4th Step: Main loop with timer** The timer is initialized and started, then a loop over all time steps iT starts the simulation, during which the functions setBoudnaryValues, collideAndStream and getResults (step 5, 6, and 7) are called repeatedly until a maximum of iterations is reached, or the simulation has converged. At the end, the timer is stopped and the summary is printed to the console.

**5th Step: Definition of initial and boundary conditions** Since the boundary is time dependent, this happens in the loop. In some applications, the boundary stays the same during whole simulation and the function doesn’t need to do anything after the very first iteration.

**6th step: Collide and stream execution** Perform the collision and streaming step. If more than one lattice is used, the function is called for each lattice separately.

**7th Step: Computation and output of results** Create console output, figures, or results of certain formats at certain timesteps.