Computational Fluid Dynamics: Lattice Boltzmann Method

Mark Alexander Kaltenborn

June 15, 2015

Introduction to LBM

Lattice Boltzmann Method (LBM) has emerged as a very popular and reliable tool for modeling the Navier-Stokes equations in computational fluid dynamics (CFD) fluid simulations. The LBM is based on the microscopic models for fluid dynamics. It can be seen as a finite difference method for solving the Boltzmann transport equation. The Navier-Stokes equations are easily recoverable with this model by fine tuning the collision function that is used. By simulating colliding and streaming the finite number of particles, realistic viscous fluid flows can be realized. First, the method will be introduced in general. Then the boundary conditions will be discussed. Following the establishment of the mechanism, the driven cavity simulation will be explain. Results gathered from the driven cavity will then be discussed.

The Lattice Boltzmann method has multiple advantages for fluid simulations over many finite difference methods. LBM is very apt for simulating multiphase or multicomponent flows. As well as, LBM is ideal for handling complex boundaries, such as moving boundaries, with particular bounce-back mechanisms. So LBM is very useful to simulate flows with complex geometries, like porous media flows.

Implementation

LBM is a useful simulation technique that has emerged and grown in popularity with computational physicists. Unlike previous CFD methods, which solved for macroscopic values numerically, LBM models consist of fictitious particles, composed of distribution functions. These fictitious particles perform collision and translation processes over a discrete lattice. LBM has several advantages over the other traditional CFD methods.

First we establish our lattice. For this method, the D2Q9 model, we utilize a square lattice. Each node on the lattice is connected with the other eight neighboring nodes. With a properly defined grid we then define our macroscopic variables.

The distribution function at each node is defined at by $f_k(\vec{x},t)$. The macroscopic fluid density, $\rho(\vec{x},t)$ can be defined as a summation of microscopic particle distribution functions,

$$\rho(\vec{x},t) = \sum_{k=0}^{8} f_k(\vec{x},t) \tag{1}$$

Accordingly, the macroscopic velocity, $\vec{u}(\vec{x},t)$ is an average of microscopic velocities $\vec{e}_k$ weighted by the distribution functions $f_k(\vec{x},t)$,

$$\vec{e}_k = \{(0,0),(1,0),(0,1),(-1,0),(0,-1),(1,1),(-1,1),(-1,-1),(1,-1)\}$$

$$\vec{u}(\vec{x},t) = \frac{1}{\rho(\vec{x},t)} \sum_{k=0}^{8} f_k(\vec{x},t)\vec{e}_k. \tag{2}$$

In the particular LBM, the collision and translation steps are defined as follows:

$$\begin{aligned}\text{Collision step: } & f_k(\vec{x},t) = f_k(\vec{x},t) + \tfrac{1}{\tau}(f_k^{eq}(\vec{x},t) - f_k(\vec{x},t)) \\ \text{Translation step: } & f_k(\vec{x}+\vec{e}_k\delta_t, t+\delta_t) = f_k(\vec{x}, t+\delta_t)\end{aligned}$$

Here, $f_k^{eq}(\vec{x},t)$ is the equilibrium distribution, and τ is considered as the relaxation time towards local equilibrium. In this single-phase flow method, the Bhatnagar-Gross-Krook (BGK) collision is used, whose equilibrium distribution f^{eq} is defined by

$$f_k^{eq}(\vec{x},t) = w_k\rho_k(\vec{x},t)(1 + 3(\vec{e}_k \cdot \vec{u}(\vec{x},t)) + \frac{9}{2}(\vec{e}_k \cdot \vec{u}(\vec{x},t))^2 - \frac{3}{2}\vec{u}(\vec{x},t) \cdot \vec{u}(\vec{x},t)), \tag{3}$$

where $t_0 = \frac{4}{9}$, $t_k = \frac{1}{9}$ for $k = 1{:}4$, $t_k = \frac{1}{36}$ for $k = 5{:}8$.

Viscosity, ν in this model are related to τ by,

$$\nu = \frac{2\tau - 1}{6} \tag{4}$$

The algorithm for this method was implemented as follows: First, Initialize the lattice, ρ, $\vec{u}$, f_k and f_k^{eq}. Second, compute macroscopic ρ and $\vec{u}$ from f_k . Third, collision step: calculate the updated distribution function. Fourth, translation step: move $f \to f'$ in the direction of $\vec{e}_k$ Fifth, bounce back: using a simple bounce back method at the regular walls to rebound distributions sent into boundary. Sixth, Zou-He velocity, moving wall: using this method to simulate the moving lid of the cavity.

Repeat the second through sixth steps.

Boundary Conditions

The boundary conditions implemented in this method are simple bounce back; however, they remain critical to maintain accuracy of the solutions. The bounce back condition is typically used to implement no-slip conditions on the edges. This bounce back condition means that when a fluid particle (discrete distribution function) reaches a boundary node, the particle will rebound back to the node whence it came with opposite momenta.

Zou-He Velocity Method

At the moving boundary of this model, the Zou-He velocity method was used to introduce velocity to the system. This is conceptually simple. Particle distributions that are translated onto the moving boundary are not simply bounced back, like at the other boundaries. At each moving boundary node, the incoming particle distributions are summed up and redistributed back towards the cavity such that they impose a velocity, $U_T = (u, v)$ into the cavity.

For example, if the top wall were to be considered the moving wall of the system, after streaming f_0, f_1, f_2, f_3, f_5 and f_6 are known. What must be calculated are f_4, f_7, f_8 and ρ.

$$\begin{aligned} f_4 + f_7 + f_8 &= \rho - (f_0 + f_1 + f_2 + f_3 + f_5 + f_6) \\ -f_7 + f_8 &= \rho u - f_1 - f_5 + f_3 + f_6 \\ f_4 + f_7 + f_8 &= \rho v + f_2 + f_3 + f_6 \end{aligned} \tag{5}$$

From these equations we can calculate an equation for ρ. Utilizing a fourth equation, $f_4 - f_4^{eq} = f_2 - f_2^{eq}$, we can solve for our sought after distributions.

$$\begin{aligned} \rho &= \frac{1}{1-v}(2(f_2 + f_5 + f_6) + f_0 + f_1 + f_3) \\ f_4 &= f_2 - \frac{2}{3}v\rho \\ f_7 &= \frac{-1}{2}\rho u + \frac{5}{6}\rho v + \frac{1}{2}(f_1 - f_3) + f_5 \\ f_8 &= \frac{1}{2}\rho u + \frac{5}{6}\rho v + \frac{1}{2}(f_3 - f_1) + f_6. \end{aligned} \tag{6}$$

Driven Cavity

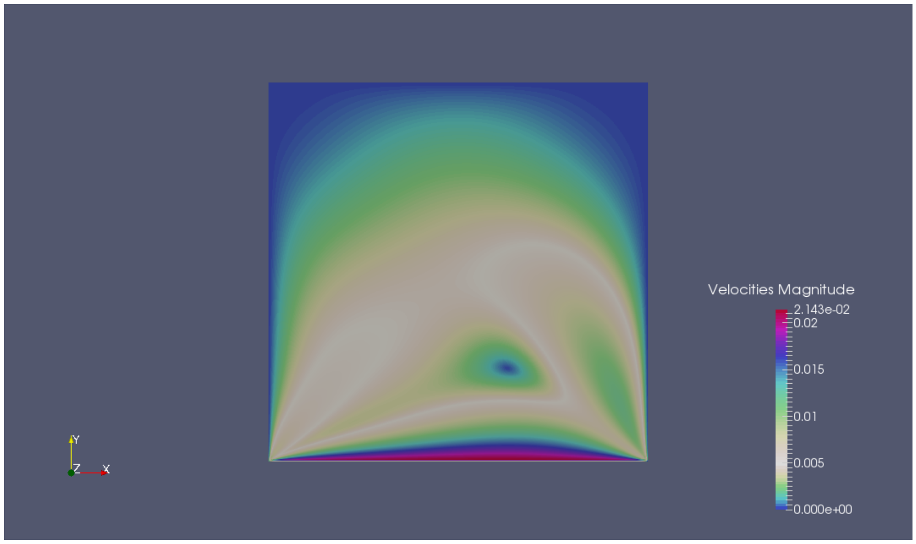
In this simulation of a 2D flow, we have a moving lid at the bottom with a speed of $U_T = (u, 0)$. The other three walls were set to have simple bounce back conditions. The moving lid has the Zou-He condition, discussed above. The initial conditions are that the velocity field is zero everywhere and the initial distribution function is set by the weights, $f_k = w_k$. This results in an initial condition that $\rho = 1$. The only exception is the velocity of the fluid on the top is set to be $U_T = (u, 0)$. The Reynold's number for the simulation is determined by,

$$Re = \frac{uL}{\nu}, \tag{7}$$

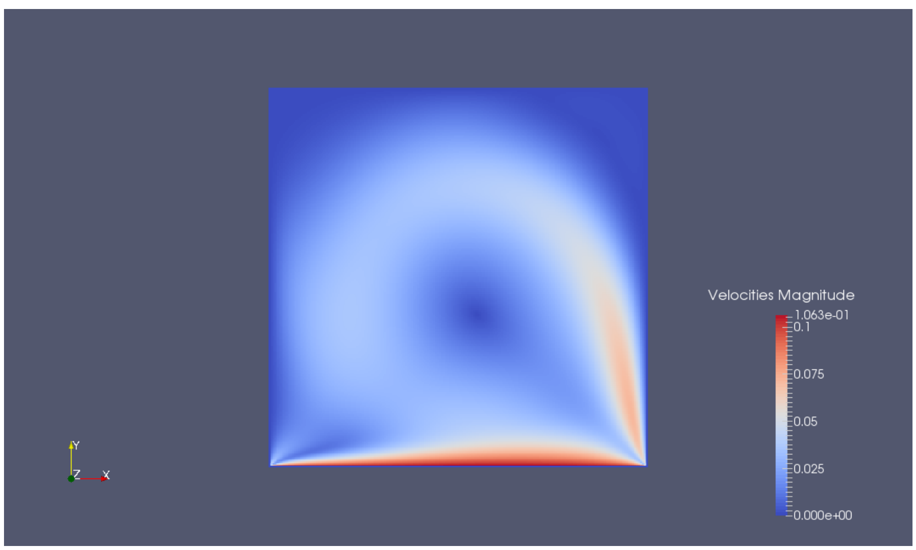
where L is the length of the lattice.

Results

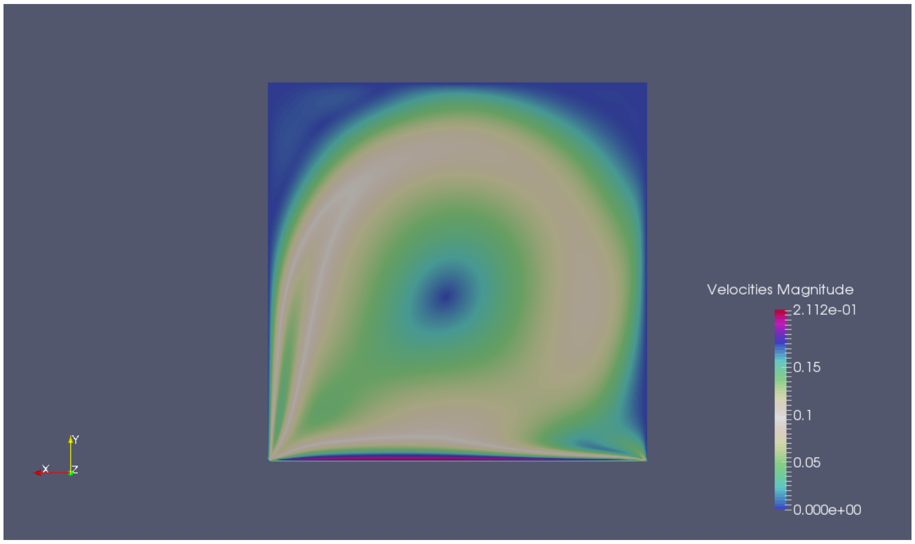
Here we include images obtained using paraview to visualize the lattice once it reaches equilibrium.



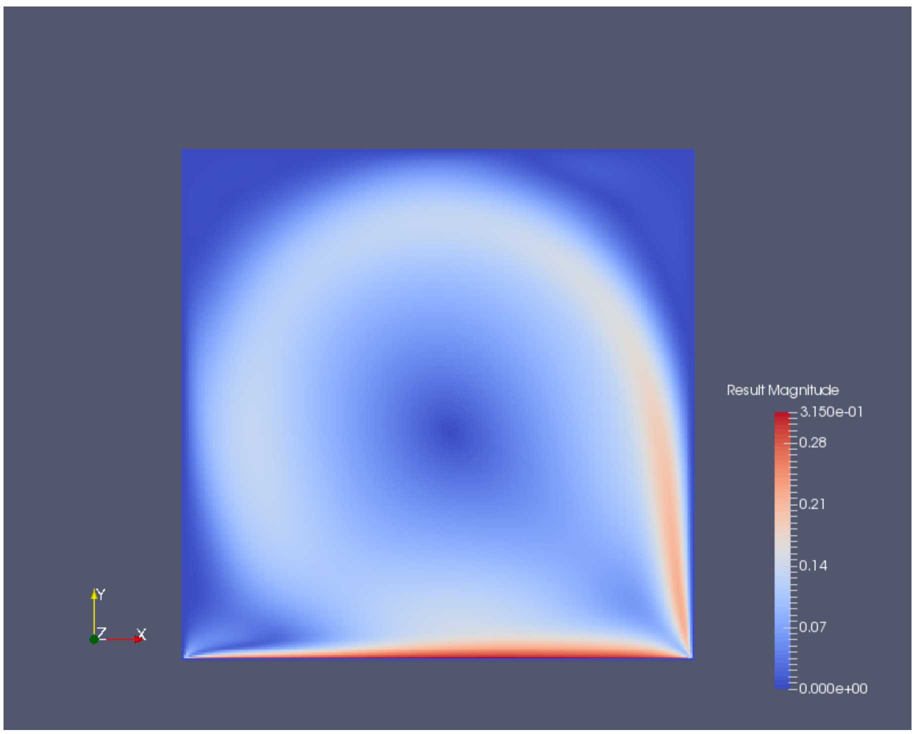
(Fig. 1 Grid 256X256, $Re = 100, u = 0.02170, \tau = \frac{2}{3}$).



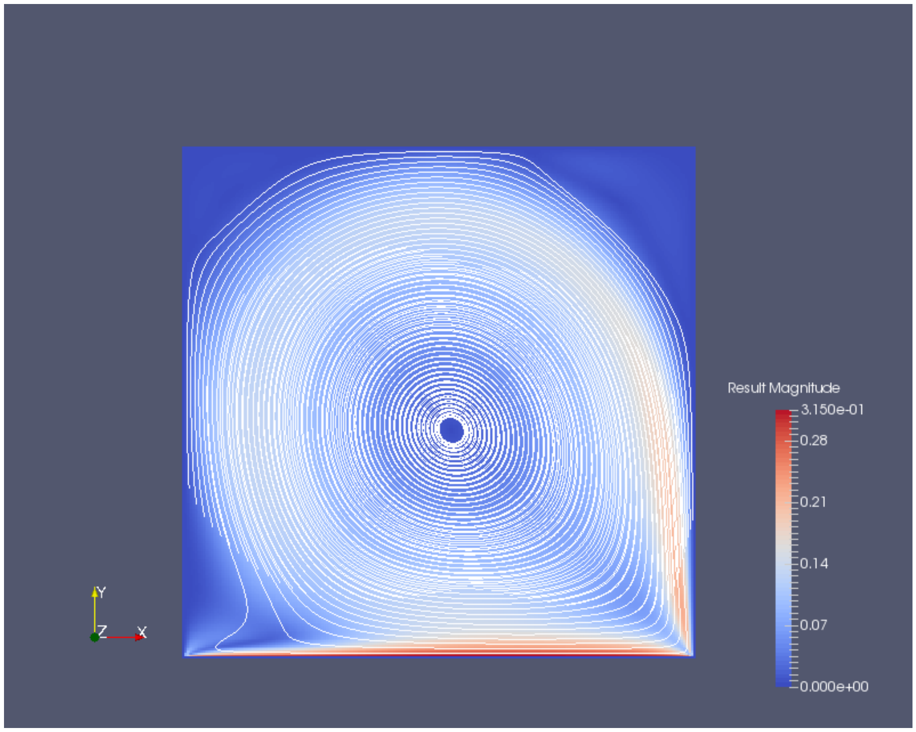
(Fig. 2 Grid 256X256, $Re = 500, u = 0.10850, \tau = \frac{2}{3}$).



(Fig. 3 Grid 256X256, $Re = 1000, u = 0.21701, \tau = \frac{2}{3}$).



(Fig. 4 Grid 256X256, $Re = 1500, u = 0.32552, \tau = \frac{2}{3}$).



(Fig. 5 Grid 256X256, $Re = 1500, u = 0.32552, \tau = \frac{2}{3}$).

Conclusion

In this project, I managed to program a functional Lattice Boltzmann method with proper boundary conditions. The validity of my model has been verified by comparing to results obtained from other LBMs lid-driven cavity models. The relation between LBM and Navier-Stokes has not been fully analyzed in this report.