# Lattice Boltzmann Method for Fluid Simulations

Yuanxun Bill Bao & Justin Meskas

April 14, 2011

## 1 Introduction

In the last two decades, the Lattice Boltzmann method (LBM) has emerged as a promising tool for modelling the Navier-Stokes equations and simulating complex fluid flows. LBM is based on microscopic models and mesoscopic kinetic equations. In some perspective, it can be viewed as a finite difference method for solving the Boltzmann transport equation. Moreover the Navier-Stokes equations can be recovered by LBM with a proper choice of the collision operator. In Section 2 and 3, we first introduce this method and describe some commonly used boundary conditions. In Section 4, the validity of this method is confirmed by comparing the numerical solution to the exact solution of the steady plane Poiseuille flow and convergence of solution is established. Some interesting numerical simulations, including the lid-driven cavity flow, flow past a circular cylinder and the Rayleigh-Bénard convection for a range of Reynolds numbers, are carried out in Section 5, 6 and 7. In Section 8, we briefly highlight the procedure of recovering the Navier-Stokes equations from LBM. A summary is provided in Section 9.

## 2 Lattice Boltzmann Model

The Lattice Boltzmann method [1, 2, 3] was originated from Ludwig Boltzmann's kinetic theory of gases. The fundamental idea is that gases/fluids can be imagined as consisting of a large number of small particles moving with random motions. The exchange of momentum and energy is achieved through particle streaming and billiard-like particle collision. This process can be modelled by the Boltzmann transport equation, which is

$$\frac{\partial f}{\partial t} + \vec{u} \cdot \nabla f = \Omega \tag{1}$$

where $f(\vec{x}, t)$ is the particle distribution function, $\vec{u}$ is the particle velocity, and $\Omega$ is the collision operator. The LBM simplifies Boltzmann's original idea of gas dynamics by reducing the number of particles and confining them to the nodes of a lattice. For a two dimensional model, a particle is restricted to stream in a possible of 9 directions, including the one staying at rest. These velocities are referred to as the *microscopic velocities* and denoted by $\vec{e}_i$, where $i = 0, \ldots, 8$. This model is commonly known as the D2Q9 model as it is two dimensional and involves 9 velocity vectors. Figure 1 shows a typical lattice node of D2Q9 model with 9 velocities $\vec{e}_i$ defined by

$$\vec{e}_i = \begin{cases} (0,0) & i = 0 \\ (1,0), (0,1), (-1,0), (0,-1) & i = 1,2,3,4 \\ (1,1), (-1,1), (-1,-1), (1,-1) & i = 5,6,7,8 \end{cases} \tag{2}$$

For each particle on the lattice, we associate a discrete probability distribution function $f_i(\vec{x}, \vec{e}_i, t)$ or simply $f_i(\vec{x}, t)$, $i = 0 \ldots 8$, which describes the probability of streaming in one particular direction.

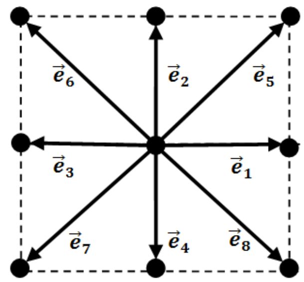


Figure 1: Illustration of a lattice node of the D2Q9 model

The *macroscopic fluid density* can be defined as a summation of microscopic particle distribution function,

$$\rho(\vec{x}, t) = \sum_{i=0}^{8} f_i(\vec{x}, t) \tag{3}$$

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

$$\vec{u}(\vec{x}, t) = \frac{1}{\rho} \sum_{i=0}^{8} c f_i \vec{e}_i \tag{4}$$

The key steps in LBM are the streaming and collision processes which are given by

$$\underbrace{f_i(\vec{x} + c\vec{e}_i \Delta t, t + \Delta t) - f_i(\vec{x}, t)}_{\text{Streaming}} = \underbrace{-\frac{[f_i(\vec{x}, t) - f_i^{eq}(\vec{x}, t)]}{\tau}}_{\text{Collision}} \tag{5}$$

In the actual implementation of the model, streaming and collision are computed separately, and special attention is given to these when dealing with boundary lattice nodes. Figure 2 shows graphically how the streaming step takes place for the interior nodes.

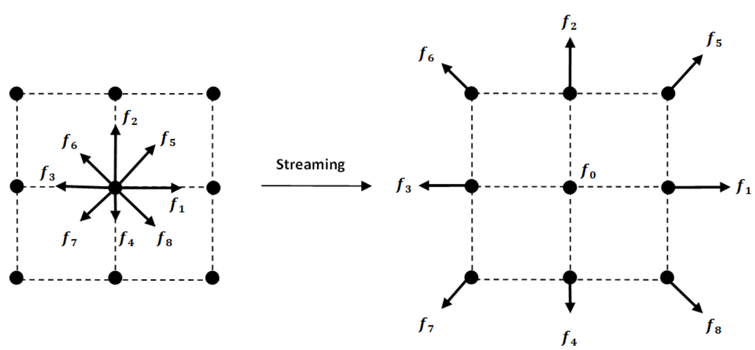


Figure 2: Illustration of the streaming process of a lattice node

In the collision term of (5), $f_i^{eq}(\vec{x}, t)$ is the equilibrium distribution, and $\tau$ is considered as the relaxation time towards local equilibrium. For simulating single phase flows, it suffices to use Bhatnagar-Gross-Krook (BGK) collision, whose equilibrium distribution $f_i^{eq}$ is defined by

$$f_i^{eq}(\vec{x}, t) = w_i \rho + \rho s_i(\vec{u}(\vec{x}, t)) \tag{6}$$

where $s_i(\vec{u})$ is defined as

$$s_i(\vec{u}) \quad = \quad w_i \left[ 3\frac{\vec{e}_i \cdot \vec{u}}{c} + \frac{9}{2}\frac{(\vec{e}_i \cdot \vec{u})^2}{c^2} - \frac{3}{2}\frac{\vec{u} \cdot \vec{u}}{c^2} \right], \tag{7}$$

and $w_i$, the weights,

$$w_i \quad = \quad \begin{cases} 4/9 & i = 0 \\ 1/9 & i = 1,2,3,4 \\ 1/36 & i = 5,6,7,8 \end{cases} \tag{8}$$

and $c = \dfrac{\Delta x}{\Delta t}$ is the lattice speed. The fluid kinematic viscosity $\nu$ in the D2Q9 model is related to the relaxation time $\tau$ by

$$\nu \quad = \quad \frac{2\tau - 1}{6} \frac{(\Delta x)^2}{\Delta t} \tag{9}$$

The algorithm can be summarized as follows:

1. Initialize $\rho$, $\vec{u}$, $f_i$ and $f_i^{eq}$
2. Streaming step: move $f_i \longrightarrow f_i^*$ in the direction of $\vec{e}_i$
3. Compute macroscopic $\rho$ and $\vec{u}$ from $f_i^*$ using (3) and (4)
4. Compute $f_i^{eq}$ using (6)
5. Collision step: calculate the updated distribution function $f_i = f_i^* - \frac{1}{\tau}(f_i^* - f_i^{eq})$ using (5)
6. Repeat step 2 to 5

Notice that numerical issues can arise as $\tau \to 1/2$. During the streaming and collision step, the boundary nodes require some special treatments on the distribution functions in order to satisfy the imposed macroscopic boundary conditions. We discuss these in details in Section 3.

# 3 Boundary Conditions

Boundary conditions (BCs) are central to the stability and the accuracy of any numerical solution. For the lattice Boltzmann method, the discrete distribution functions on the boundary have to be taken care of to reflect the macroscopic BCs of the fluid. In this project, we explore two of the most widely used BCs: Bounce-back BCs [4] and Zou-He velocity and pressure (density) BCs [5].

## 3.1 Bounce-back BCs

Bounce-back BCs are typically used to implement no-slip conditions on the boundary. By the so-called bounce-back we mean that when a fluid particle (discrete distribution function) reaches a boundary node, the particle will scatter back to the fluid along with its incoming direction. Bounce-back BCs come in a few variants and we focus on two types of implementations: the on-grid and the mid-grid bounce-back [4].

The idea of the on-grid bounce-back is particularly simple and preserves a decent numerical accuracy. In this configuration, the boundary of the fluid domain is aligned with the lattice points (see Figure 3). One can use a boolean mask for the boundary and the interior nodes. The incoming directions of the distribution functions are reversed when encountering a boundary node. This implementation

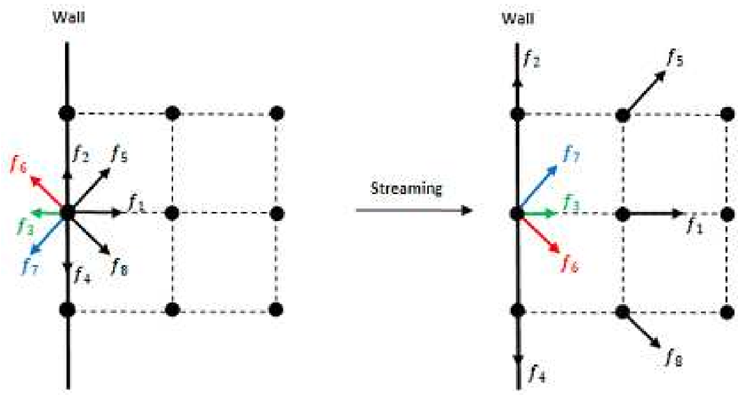


Figure 3: Illustration of on-grid bounce-back

does not distinguish the orientation of the boundaries and is ideal for simulating fluid flows in complex geometries, such as the porous media flow.

The configuration of the mid-grid bounce-back introduces fictitious nodes and places the boundary wall centered between fictitious nodes and boundary nodes of the fluid (see Figure 4). At a given time step $t$, the distribution functions with directions towards the boundary wall would leave the domain. Collision process is then applied and directions of these distribution functions are reversed and they bounce back to the boundary nodes. We point out that the distribution functions at the end of bounce-back in this configuration is the post-collision distribution functions.

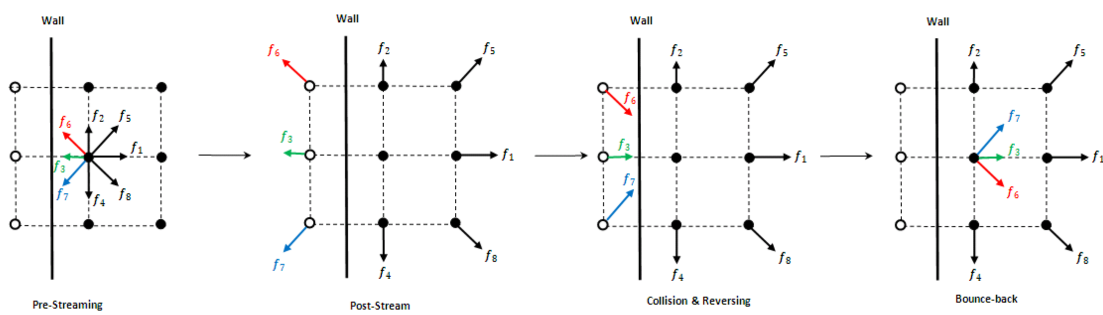


Figure 4: Illustration of mid-grid bounce-back

Although the on-grid bounce-back is easy to implement, it has been verified that it is only first-order accurate due to its one-sided treatment on streaming at the boundary. However the centered nature of the mid-grid bounce-back leads to a second order of accuracy at the price of a modest complication.

## 3.2 Zou-He Velocity and Pressure BCs

In many physical situations, we would like to model flows with prescribed velocity or pressure (density) at the boundary. This particular velocity/pressure BC we discuss here was originally developed by Zou and He in [5]. For illustration, we consider that the velocity $\vec{u}_L = (u, v)$ is given on the left boundary. After streaming, $f_0, f_2, f_3, f_4, f_6$ and $f_7$ are known. What's left undetermined are $f_1, f_5, f_8$ and $\rho$ (see Figure 5).

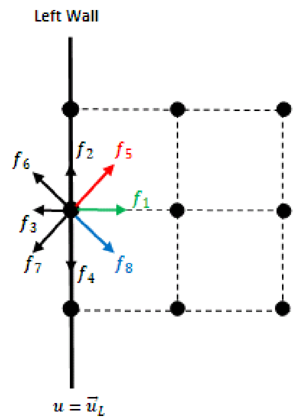


Figure 5: Illustration of Zou-He velocity BC

The idea of Zou-He BCs is to formulate a linear system of $f_1, f_5, f_8$ and $\rho$ using (3) and (4). After rearranging:

$$
\begin{aligned}
f_1 + f_5 + f_8 &= \rho - (f_0 + f_2 + f_4 + f_3 + f_6 + f_7) && (10)\\
f_1 + f_5 + f_8 &= \rho u + (f_3 + f_6 + f_7) && (11)\\
f_5 - f_8 &= \rho v - f_2 + f_4 - f_6 + f_7 && (12)
\end{aligned}
$$

By considering (10) and (11), we can determine

$$
\rho = \frac{1}{1-u}\left[(f_0 + f_2 + f_4 + 2(f_3 + f_6 + f_7)\right] \quad (13)
$$

However, we need a fourth equation to close the system and solve for $f_1$, $f_5$ and $f_8$. The assumption made by Zou and He is that the bounce-back rule still holds for the non-equilibrium part of the particle distribution normal to the boundary. In this case, the fourth equation is

$$
f_1 - f_1^{eq} = f_3 - f_3^{eq} \quad (14)
$$

With $f_1$ solved by (6) and (14), $f_5$, $f_8$ are subsequently determined:

$$
\begin{aligned}
f_1 &= f_3 + \frac{2}{3}\rho v && (15)\\
f_5 &= f_7 - \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho u + \frac{1}{2}\rho v && (16)\\
f_8 &= f_6 + \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho u - \frac{1}{2}\rho v && (17)
\end{aligned}
$$

A similar procedure is taken if a given pressure (density) is imposed on the boundary. Here we notice that this type of BC depends on the orientation of the boundary and thus is hard to generalize for complex geometries.