3
Methodology

This chapter provides a detailed description of the methodology used in the present thesis work to simulate various fluid flow problems using LBM. First, the formulation for governing equations of the particular SRT-LBM is introduced in detail. Secondly, the incorporation of LES turbulence model in LBE is explained. Later, a comprehensive overview of the different boundary conditions is also described in detail. Lastly, an explanation of the CUDA programming model for parallelizing LBM code to make it suitable to run on multi-threaded GPU platform is presented.

The remainder of the chapter is organized as follows: Section 3.1 provides a detailed description to the formulation of LBM, followed by the explanation of incorporating the LES turbulence model in the LBM framework. The details of the boundary conditions are presented in Section 3.2. Section 3.3 includes the information of the GPU architecture and the implementation of the LBM algorithm on the GPU platform. Section 3.4 summarizes the chapter.

3.1 SIMULATION PROCEDURE

In this section, a comprehensive description of the methodology used in the thesis work has been presented. The section starts with the detailed introduction of the SRT-LBM collision model of LBM, used in the present thesis work to discretize the fluid domain, followed by the explanation for incorporating the LES turbulence model in the LBE.

3.1.1 Single-Relaxation-Time (SRT) model

This subsection discussed the single relaxation time (SRT) model of LBM. SRT is the most commonly used model of LBM for solving fluid flow problems [Qian *et al.*, 1992]. SRT-LBM is described here for D_2Q_9, D_3Q_{15}, D_3Q_{19}, and D_3Q_{27} discrete velocity models of LBM [Yasuda *et al.*, 2017]. Figure 2.2 shows the lattice structure of D_2Q_9, D_3Q_{15}, D_3Q_{19}, and D_3Q_{27} discrete velocity models. The governing LBE for the SRT-LBM can be written as:

$$f_k(\vec{x}+\vec{e}_k\Delta t, t+\Delta t) = f_k(\vec{x},t) + \frac{1}{\tau}[f_k^{eq}(\vec{x},t) - f_k(\vec{x},t)] \tag{3.1}$$

where, f_k and f_k^{eq} are the particle and equilibrium distribution function along direction k at $(\vec{x},t)$, $\vec{e}_k$ is the particle velocity vector in the k^{th} direction, Δt is the time step size, and τ is the single-relaxation time parameter that controls the rate of approach to equilibrium [Perumal and Dass, 2015]. The above equation for SRT-LBM consists of two essential steps, namely streaming and collision steps.

Collision step: In the step particle collides locally, The step relaxes the particle distribution function f_k to the equilibrium distribution value f_k^{eq} at each lattice location [Hecht and Harting, 2010]. The mathematical expression for the step can be written as:

$$f_k^{'}(\vec{x},t) = f_k(\vec{x},t) - \frac{1}{\tau}[f_k(\vec{x},t) - f_k^{eq}(\vec{x},t)] \tag{3.2}$$

Streaming step: The step is not local, and each of the particle distribution f_k propagate to

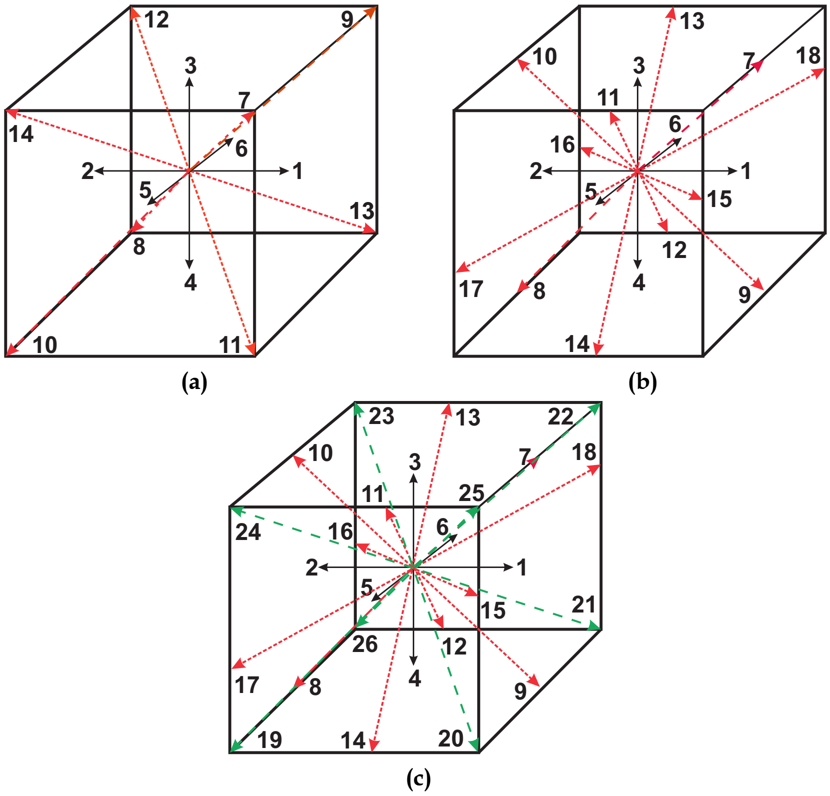

Figure 3.1 : Lattice structure of (a) D_2Q_9, (b) D_3Q_{15}, (c) D_3Q_{19}, and (d) D_3Q_{27} discrete velocity models.

neighboring lattice site in the assigned direction of particle velocity vector $\vec{e}_k$.

$$f_k(\vec{x}+\vec{e}_k\Delta t, t+\Delta t) = f_k^{'}(\vec{x},t) \tag{3.3}$$

where f_k and $f_k^{'}$ represent the particle distribution function at the pre-collision and post-collision states, respectively [Perumal and Dass, 2015]. The corresponding equation to obtain equilibrium distribution function f_k^{eq} is given as:

$$f_k^{eq} = w_k\rho\left[1+\frac{\vec{e}_k.\vec{u}}{c_s^2}+\frac{(\vec{e}_k\cdot\vec{u})^2}{2c_s^4}-\frac{\vec{u}\cdot\vec{u}}{2c_s^2}\right] \tag{3.4}$$

where, w is the weight factor, ρ is the fluid density , $\vec{u}$ = (u_x, u_y, u_z) is the macroscopic fluid velocity vector, and c_s is the speed of sound which is related to lattice speed, c, and equal to $c_s = \frac{c}{\sqrt{3}}$ for D_3Q_{15}, D_3Q_{19}, and D_3Q_{27} discrete velocity models. Also, the lattice speed, $c = \frac{\Delta x}{\Delta t}$, and Δx represent the size of the lattice and Δt is the time-step size. The corresponding discrete velocity vector $\vec{e}_k$ and weights w_k for the D_3Q_{15} lattice structure are given as:

$$\vec{e}_k = \begin{cases} c(\pm1,0,0), c(0,\pm1,0), c(0,0,\pm1), & k=1....6 \\ c(\pm1,\pm1,\pm1), & k=7....14 \\ c(0,0,0), & k=15 \end{cases} \tag{3.5}$$

$$w_k = \begin{cases} \frac{1}{9}, & k=1....6 \\ \frac{1}{72}, & k=7....14 \\ \frac{2}{9}, & k=15 \end{cases} \tag{3.6}$$

For D_3Q_{19} lattice structure, the discrete velocity vector $\vec{e}_k$ and weights w_k are expressed as:

$$\vec{e}_k = \begin{cases} c(\pm1,0,0), c(0,\pm1,0), c(0,0,\pm1), & k = 1....6 \\ c(\pm1,\pm1,0), c(\pm1,0,\pm1), c(0,\pm1,\pm1), & k = 7....18 \\ c(0,0,0), & k = 19 \end{cases} \tag{3.7}$$

$$w_k = \begin{cases} \frac{1}{18}, & k = 1....6 \\ \frac{1}{36}, & k = 7....18 \\ \frac{1}{3}, & k = 19 \end{cases} \tag{3.8}$$

For D_3Q_{27} lattice structure, the discrete velocity vector $\vec{e}_k$ and weights w_k are expressed as:

$$\vec{e}_k = \begin{cases} c(\pm1,0,0), c(0,\pm1,0), c(0,0,\pm1), & k = 1....6 \\ c(\pm1,\pm1,0), c(\pm1,0,\pm1), c(0,\pm1,\pm1), & k = 7....18 \\ c(\pm1,\pm1,\pm1), & k = 19....26 \\ c(0,0,0), & k = 27 \end{cases} \tag{3.9}$$

$$w_k = \begin{cases} \frac{2}{27}, & k = 1....6 \\ \frac{1}{54}, & k = 7....18 \\ \frac{1}{216}, & k = 19....26 \\ \frac{8}{27}, & k = 27 \end{cases} \tag{3.10}$$

The relaxation time parameter in Eqn. (3.1) depends on the kinematic viscosity of the fluid is given as:

$$\nu = \left(\frac{2\tau - 1}{6}\right) \frac{(\Delta x)^2}{\Delta t} \tag{3.11}$$

The hydrodynamic fields i.e. density ρ and momentum density $\rho\vec{u}$ can be calculated from the particle distribution function f_k as follows:

$$\rho = \sum_k f_k = \sum_k f_k^{eq}, \; \rho\vec{u} = \sum_k \vec{e}_k f_k = \sum_k \vec{e}_k f_k^{eq} \tag{3.12}$$

3.1.2 LES Turbulent Model in LBM

An essential aspect of the present thesis work is the modeling of turbulent eddies, as the Reynolds number used in the simulations indicates a robust, turbulent flow. The direct simulation is not feasible due to the limitation of the computational resources. DNS would require a fine meshing of the domain and an enormous number of time steps to model the turbulent flow. Thus, LES is chosen in the present thesis work for the modeling of turbulent flow. LES has been widely accepted by researchers for the simulation of turbulent flows because it is flexible for implementation on complex geometries, provides comprehensive information for different turbulent statistics, and requires less computational power as compared to DNS [Ayranci *et al.*, 2012]. The idea behind the approach is to decompose the flow variables into large and small-scale structures by specifying a filtering procedure [Stephen B., 2000]. In LES, the process assumed that small-scale flow structures are universal and of isotropic nature (i.e., independent of the flow geometry). As a result, it provides an advantage to the method for modeling turbulent flow with relative ease [Derksen and Van den Akker, 1999]. The procedure enables large-scale structures to be resolved explicitly in grid calculation and uses the SGS model for the small-scale, turbulent structures. The general filtering operation is given as:

$$\overline{\phi}(x) = \int \phi(x') G(x, x') dx' \tag{3.13}$$

where ϕ is the spatial dependent quantity and G is the kernel function and integral is extended over the entire domain [Hou *et al.*, 1994; Koda and Lien, 2015]. The filtered SRT-LBM equation can be written as:

$$\overline{f}_k(\vec{x}+\vec{e}_k\Delta t, t+\Delta t) = \overline{f}_k(\vec{x},t) - \frac{1}{\tau_t}[\overline{f}_k(\vec{x},t) - \overline{f}_k^{eq}(\vec{x},t)] \tag{3.14}$$

where $\overline{f}_k^{eq}$ and $\overline{f}_k$ and the filtered quilibrium distribution function and particle distribution function, respectively. In the present thesis work, the SGS model (also known as the eddy-viscosity model) is adopted to resolve small-scale structures. The eddy viscosity term ν_{SGS} can be obtained from filter width Δ (with a size equal to grid spacing Δx) and characteristic filtered rate of strain $\overline{S}$. The expression for the same is given as:

$$\begin{aligned} \nu_{SGS} &= (C_s\Delta)^2\overline{S} \\ \overline{S} &= \sqrt{2\sum_{ij} S_{ij}S_{ij}} \end{aligned} \tag{3.15}$$

where C_s is the Smagorinsky constant, $S_{ij} = (\partial_j\vec{u}_i + \partial_i\vec{u}_j)$ is the filtered strain rate tensor. The subgrid closure can be implemented directly in the LBM equation by simply replacing physical kinematic viscosity ν, with the total effective viscosity ν_t in collision step [Koda and Lien, 2015]. The expression for total effective viscosity ν_t is given as:

$$\nu_t = \nu + \nu_{SGS} \tag{3.16}$$

In LBM, the viscosity of fluid is related to the relaxation time parameter as given in Eq.(3.11). Thus,

$$\nu_t = \left(\frac{2\tau_t - 1}{6}\right)\frac{(\Delta x)^2}{\Delta t} \tag{3.17}$$

on solving Eqn.(3.11),(3.15)-(3.17), we get:

$$\tau_t = \tau + \frac{3\Delta t(C_s\Delta)^2\overline{S}}{(\Delta x)^2} \tag{3.18}$$

where τ_t is total value of the relaxation time. The filtered strain rate tensor S_{ij} can be obtained directly from the non-equilibrium momentum flux tensor $\overline{\Pi}_{ij}$ given as:

$$\overline{\Pi}_{ij} = \sum_k \vec{e}_{ki}\vec{e}_{kj}(\overline{f}_i - \overline{f}_i^{eq}) = \frac{-2\overline{\rho}\tau_t(\Delta x)^2 S_{ij}}{3\Delta t} \tag{3.19}$$

Also,

$$Q^{1/2} = \sqrt{\overline{\Pi}_{ij} : \overline{\Pi}_{ij}} = \frac{\sqrt{2}\overline{\rho}\tau_t(\Delta x)^2\overline{S}}{3\Delta t} \tag{3.20}$$

Substituting Eq. (3.18) in Eq. (3.20) gives the expression for the characteristic filtered rate of strain $\overline{S}$:

$$\overline{S} = \frac{-\tau\overline{\rho}c\Delta x + \sqrt{(\tau\overline{\rho}c\Delta x)^2 + 18\sqrt{2}(C_s\Delta)^2 Q^{1/2}}}{6\overline{\rho}(C_s\Delta)^2} \tag{3.21}$$

Finally, on substituting Eq. (3.21) in Eq. (3.18) gives:

$$\tau_t = \frac{\tau}{2} + \frac{\sqrt{(\tau\overline{\rho}c\Delta x)^2 + 18\sqrt{2}(C_s\Delta)^2 Q^{1/2}}}{2\overline{\rho}c} \tag{3.22}$$

3.2 BOUNDARY CONDITIONS

Boundary conditions play a vital role in lattice Boltzmann simulation. An individual used distribution functions (f_k) to implement boundary conditions in LBM instead of using hydrodynamics variables [Mussa *et al.*, 2009]. Therefore, the appropriate mathematical expressions need to be determined to obtain the distribution functions at the boundary nodes. The researchers developed various methods to implement constant velocity inlet, no-slip, free-slip boundary conditions in the frame of LBM. The details of some of the boundary conditions used in the present thesis work are described below:

3.2.1 On-site Velocity Boundary Condition

Zou and He [1997] was the first who proposed the on-site velocity boundary condition for the D_2Q_9 lattice structure in order to simulate 2-D flow. The work presented by Zou and He [1997] also provides an overview of the on-site velocity boundary condition for the D_3Q_{15} lattice model for 3-D flow simulation. The work was further studied by Kutay *et al.* [2006] to demonstrate the velocity boundary condition for the D_3Q_{19} lattice model. The boundary condition proposed by Kutay *et al.* [2006] suffers from the limitation of inflow being perpendicular to the boundary plane. Later, Hecht and Harting [2010] modified the on-site velocity boundary condition with variable influx directions for the D_3Q_{19} lattice structure. The present section discussed the on-site velocity boundary conditions for the different three-dimension lattice structures of LBM. The geometry for the D_3Q_{15}, D_3Q_{19}, and D_3Q_{27} lattice structure are shown in Figure 3.1. The benchmark case of incompressible laminar flow through a 3-D square duct, as shown in Figure 3.2, has been chosen to derive the boundary condition. The mathematical expression for the on-site velocity boundary for each of the D_3Q_{15}, D_3Q_{19}, and D_3Q_{27} lattice structure can be written as:

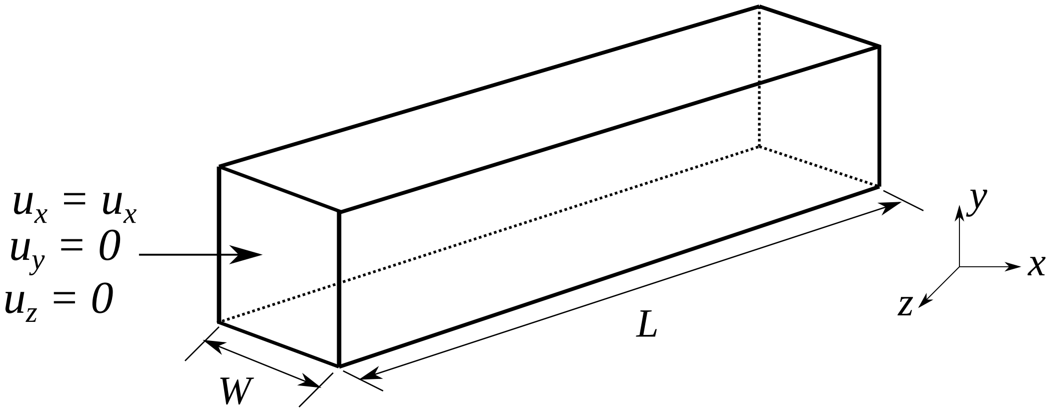

Figure 3.2 : Schematic representation of flow in 3D square duct

- **For D_3Q_{15} lattice structure**: The on-site velocity boundary condition for the D_3Q_{15} lattice structure can be expressed as follows:

  1. Consider the left boundary (i.e x = 0) as inlet as shown in Figure 3.2. The inlet is on the *yz* plane. The density ρ for the D_3Q_{15} lattice structure from Eqn.(3.12) can be written as:

$$\begin{aligned}\rho = f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8 + f_9 + f_{10} \\ + f_{11} + f_{12} + f_{13} + f_{14} + f_{15}\end{aligned} \tag{3.23}$$

$$\begin{aligned}\rho u_x = f_1 + f_7 + f_9 + f_{11} + f_{13} \\ - (f_2 + f_8 + f_{10} + f_{12} + f_{14})\end{aligned} \tag{3.24}$$

After the streaming steps in the LBE, the distribution functions f_k, ($k = 2, 3, 4, 5, 6, 8, 10, 12, 14, 15$) are known. The velocity components $u_y = u_z = 0$ are fixed at inlet of the square duct. The velocity component u_x and unknown fluid populations, f_k, $(k = 1, 7, 9, 11, 13)$ needs to be determined. These can be obtained as follows:

On subtracting Eqn.(3.24) from Eqn.(3.23) we get

$$\rho = \frac{1}{1-u_x}\big[f_3+f_4+f_5+f_6+f_{15} \\ +2(f_2+f_8+f_{10}+f_{12}+f_{14})\big] \tag{3.25}$$

or

$$u_x = 1-\frac{1}{\rho}\big[f_3+f_4+f_5+f_6+f_{15} \\ +2(f_2+f_8+f_{10}+f_{12}+f_{14})\big] \tag{3.26}$$

Considering the bounce-back rule suggested by Zou and He [1997] for the non-equilibrium part of the particle distribution function f_k (k = 1, 7, 9, 11, 13) we have:

$$f_1 - f_1^{eq} = f_2 - f_2^{eq} \tag{3.27}$$

It gives:

$$f_1 = f_2 + \frac{2}{3}\rho u_x \tag{3.28}$$

Also,

$$f_7 = f_8 + +\frac{1}{12}\rho u_x - \frac{1}{4}\big[(f_3-f_4)+(f_5-f_6)\big] \tag{3.29}$$

$$f_9 = f_1 0 + +\frac{1}{12}\rho u_x - \frac{1}{4}\big[(f_3-f_4)-(f_5-f_6)\big] \tag{3.30}$$

$$f_1 1 = f_1 2 + +\frac{1}{12}\rho u_x - \frac{1}{4}\big[-(f_3-f_4)+(f_5-f_6)\big] \tag{3.31}$$

$$f_1 3 = f_1 4 + +\frac{1}{12}\rho u_x - \frac{1}{4}\big[-(f_3-f_4)-(f_5-f_6)\big] \tag{3.32}$$

The on-site velocity boundary conditions for the other sides of the geometry can be obtained using the same approach and similarly, we can derive the expressions for the unknown distribution functions in the D_3Q_{19} and D_3Q_{27} lattice structure. These are given as follows:

- **For D_3Q_{19} lattice structure**: Hecht and Harting [2010] presented the on-site velocity boundary condition for the D_3Q_{19} lattice structure. The mathematical expressions are given as:

  1. **At the left boundary (i.e x = 0) as inlet**:

$$\rho = \frac{1}{1-u_x}\big[f_3+f_4+f_5+f_6+f_{15}+f_{16}+f_{17}+f_{18}+f_{19} \\ +2(f_2+f_{11}+f_{12}+f_{13}+f_{14})\big] \tag{3.33}$$

or

$$u_x = 1-\frac{1}{\rho}\big[f_3+f_4+f_5+f_6+f_{15}+f_{16}+f_{17}+f_{18}+f_{19} \\ +2(f_2+f_{11}+f_{12}+f_{13}+f_{14})\big] \tag{3.34}$$

and

$$f_1 = f_2 + \frac{1}{3}\rho u_x \tag{3.35}$$

Similarly, we get

$$f_8 = f_{11} + \frac{\rho}{6}(u_x - u_y) + N_y^x \tag{3.36}$$

$$f_7 = f_{12} + \frac{\rho}{6}(u_x + u_y) - N_y^x \tag{3.37}$$

$$f_9 = f_{14} + \frac{\rho}{6}(u_x + u_z) - N_z^x \tag{3.38}$$

$$f_{10} = f_{13} + \frac{\rho}{6}(u_x - u_z) + N_z^x \tag{3.39}$$

where

$$N_y^x = \frac{1}{2}\left[f_3 + f_{15} + f_{16} - (f_4 + f_{17} + f_{18})\right] - \frac{1}{3}\rho u_y \tag{3.40}$$

$$N_z^x = \frac{1}{2}\left[f_5 + f_{11} + f_{15} - (f_6 + f_{16} + f_{18})\right] - \frac{1}{3}\rho u_z \tag{3.41}$$

2. **For the right boundary ((i.e. x = n_x) as outlet**:

$$\begin{aligned}\rho = \frac{1}{u_x + 1}&\left[f_3 + f_4 + f_5 + f_6 + f_{15} + f_{16} + f_{17} + f_{18} + f_{19}\right.\\ &\left. + 2(f_1 + f_7 + f_8 + f_9 + f_{10})\right]\end{aligned} \tag{3.42}$$

or

$$\begin{aligned}u_x = -1 + \frac{1}{\rho}&\left[f_3 + f_4 + f_5 + f_6 + f_{15} + f_{16} + f_{17} + f_{18} + f_{19}\right.\\ &\left. + 2(f_1 + f_7 + f_8 + f_9 + f_{10})\right]\end{aligned} \tag{3.43}$$

and

$$f_2 = f_1 - \frac{1}{3}\rho u_x \tag{3.44}$$

$$f_{11} = f_8 + \frac{\rho}{6}(u_y - u_x) - N_y^x \tag{3.45}$$

$$f_{12} = f_7 + \frac{\rho}{6}(-u_x - u_y) + N_y^x \tag{3.46}$$

$$f_{14} = f_9 + \frac{\rho}{6}(-u_x - u_z) + N_z^x \tag{3.47}$$

$$f_{13} = f_{10} + \frac{\rho}{6}(u_z - u_x) - N_z^x \tag{3.48}$$

3. **For the front boundary ((i.e. y = 0) as inlet**:

$$\begin{aligned}\rho = \frac{1}{1 - u_y}&\left[f_1 + f_2 + f_5 + f_6 + f_9 + f_{10} + f_{13} + f_{14} + f_{19}\right.\\ &\left. + 2(f_4 + f_8 + f_{12} + f_{17} + f_{18})\right]\end{aligned} \tag{3.49}$$

or

$$u_y = 1 - \frac{1}{\rho}\big[f_1 + f_2 + f_5 + f_6 + f_9 + f_{10} + f_{13} + f_{14} + f_{19} \\ + 2(f_4 + f_8 + f_{12} + f_{17} + f_{18})\big] \tag{3.50}$$

and

$$f_3 = f_4 + \frac{1}{3}\rho u_y \tag{3.51}$$

$$f_7 = f_{12} + \frac{\rho}{6}(u_y + u_x) - N_x^y \tag{3.52}$$

$$f_{11} = f_8 + \frac{\rho}{6}(u_y - u_x) + N_x^y \tag{3.53}$$

$$f_{15} = f_{18} + \frac{\rho}{6}(u_y + u_z) - N_z^y \tag{3.54}$$

$$f_{16} = f_{17} + \frac{\rho}{6}(u_y - u_z) + N_z^y \tag{3.55}$$

where

$$N_x^y = \frac{1}{2}\big[f_1 + f_9 + f_{10} - (f_2 + f_{13} + f_{14})\big] - \frac{1}{3}\rho u_x \tag{3.56}$$

$$N_z^y = \frac{1}{2}\big[f_5 + f_9 + f_{13} - (f_6 + f_{10} + f_{14})\big] - \frac{1}{3}\rho u_z \tag{3.57}$$

4. **At the back boundary ((i.e. y = n_y) as outlet**:

$$\rho = \frac{1}{u_y + 1}\big[f_1 + f_2 + f_5 + f_6 + f_9 + f_{10} + f_{13} + f_{14} + f_{19} \\ + 2(f_3 + f_7 + f_{11} + f_{15} + f_{16})\big] \tag{3.58}$$

or

$$u_y = \frac{1}{1 - \rho}\big[f_1 + f_2 + f_5 + f_6 + f_9 + f_{10} + f_{13} + f_{14} + f_{19} \\ + 2(f_3 + f_7 + f_{11} + f_{15} + f_{16})\big] \tag{3.59}$$

and

$$f_4 = f_3 - \frac{1}{3}\rho u_y \tag{3.60}$$

$$f_{12} = f_7 + \frac{\rho}{6}(-u_y - u_x) + N_x^y \tag{3.61}$$

$$f_8 = f_{11} + \frac{\rho}{6}(-u_y + u_x) - N_x^y \tag{3.62}$$

$$f_{18} = f_{15} + \frac{\rho}{6}(-u_y - u_z) + N_z^y \tag{3.63}$$

$$f_{17} = f_{16} + \frac{\rho}{6}(-u_y + u_z) - N_z^y \tag{3.64}$$

5. **For the bottom boundary ((i.e. z = 0) as inlet**:

$$\begin{aligned}\rho = &\frac{1}{1-u_z}\big[f_1+f_2+f_3+f_4+f_7+f_8+f_{11}+f_{12}+f_{19}\\ &+2(f_6+f_{10}+f_{14}+f_{16}+f_{18})\big]\end{aligned} \tag{3.65}$$

or

$$\begin{aligned}u_z = 1 - &\frac{1}{\rho}\big[f_1+f_2+f_3+f_4+f_7+f_8+f_{11}+f_{12}+f_{19}\\ &+2(f_6+f_{10}+f_{14}+f_{16}+f_{18})\big]\end{aligned} \tag{3.66}$$

and

$$f_5 = f_6 + \frac{1}{3}\rho u_z \tag{3.67}$$

$$f_9 = f_{14} + \frac{\rho}{6}(u_z+u_x) - N_x^z \tag{3.68}$$

$$f_{13} = f_{10} + \frac{\rho}{6}(u_z-u_x) + N_x^z \tag{3.69}$$

$$f_{15} = f_{18} + \frac{\rho}{6}(u_z+u_y) - N_y^z \tag{3.70}$$

$$f_{17} = f_{16} + \frac{\rho}{6}(u_z-u_y) + N_y^z \tag{3.71}$$

where

$$N_x^z = \frac{1}{2}\big[f_1+f_7+f_8-(f_2+f_{11}+f_{12})\big] - \frac{1}{3}\rho u_x \tag{3.72}$$

$$N_y^z = \frac{1}{2}\big[f_3+f_7+f_{11}-(f_4+f_8+f_{12})\big] - \frac{1}{3}\rho u_y \tag{3.73}$$

6. **For the top boundary ((i.e. z = n_z) as outlet**:

$$\begin{aligned}\rho = &\frac{1}{1-u_z}\big[f_1+f_2+f_3+f_4+f_7+f_8+f_{11}+f_{12}+f_{19}\\ &+2(f_5+f_9+f_{13}+f_{15}+f_{17})\big]\end{aligned} \tag{3.74}$$

or

$$\begin{aligned}u_z = -1 + &\frac{1}{\rho}\big[f_1+f_2+f_3+f_4+f_7+f_8+f_{11}+f_{12}+f_{19}\\ &+2(f_5+f_9+f_{13}+f_{15}+f_{17})\big]\end{aligned} \tag{3.75}$$

and

$$f_6 = f_5 - \frac{1}{3}\rho u_z \tag{3.76}$$

$$f_{10} = f_{13} + \frac{\rho}{6}(-u_z+u_x) + N_x^z \tag{3.77}$$

$$f_{14} = f_9 + \frac{\rho}{6}(-u_z-u_x) - N_x^z \tag{3.78}$$

$$f_{16} = f_{17} + \frac{\rho}{6}(-u_z+u_y) + N_y^z \tag{3.79}$$

$$f_{18} = f_{15} + \frac{\rho}{6}(-u_z-u_y) - N_y^z \tag{3.80}$$

- **For D_3Q_{27} lattice structure**:

1. **At the left boundary (i.e x = 0) as inlet**:

$$\begin{aligned}\rho = \frac{1}{1-u_x}\big[&f_3+f_4+f_5+f_6+f_{15}\\ &+f_{16}+f_{17}+f_{18}+f_{19}+2(f_2+f_{11}+f_{12}\\ &+f_{13}+f_{14}+f_{21}+f_{23}+f_{25}+f_{27})\big]\end{aligned} \tag{3.81}$$

or

$$\begin{aligned}u_x = 1-\frac{1}{\rho}\big[&f_3+f_4+f_5+f_6+f_{15}\\ &+f_{16}+f_{17}+f_{18}+f_{19}+2(f_2+f_{11}+f_{12}\\ &+f_{13}+f_{14}+f_{21}+f_{23}+f_{25}+f_{27})\big]\end{aligned} \tag{3.82}$$

$$f_1 = f_2 + \frac{4}{9}\rho u_x \tag{3.83}$$

Similarly, we get

$$f_7 = f_{12} + \frac{\rho}{9}(u_x+u_y) - N_y^x \tag{3.84}$$

$$f_8 = f_{11} + \frac{\rho}{9}(u_x-u_y) + N_y^x \tag{3.85}$$

$$f_9 = f_{14} + \frac{\rho}{9}(u_x+u_z) - N_z^x \tag{3.86}$$

$$f_{10} = f_{13} + \frac{\rho}{9}(u_x-u_z) + N_z^x \tag{3.87}$$

$$f_{20} = f_{21} + \frac{\rho}{36}\big[u_x+(u_y+u_z)\big] - N_y^x - N_z^x \tag{3.88}$$

$$f_{22} = f_{23} + \frac{\rho}{36}\big[u_x+(u_y-u_z)\big] - N_y^x + N_z^x \tag{3.89}$$

$$f_{24} = f_{25} + \frac{\rho}{36}\big[u_x-(u_y-u_z)\big] + N_y^x - N_z^x \tag{3.90}$$

$$f_{26} = f_{27} + \frac{\rho}{36}\big[u_x-(u_y+u_z)\big] + N_y^x + N_z^x \tag{3.91}$$

where

$$N_y^x = \frac{1}{6}\big[f_3+f_{15}+f_{16}-(f_4+f_{17}+f_{18})\big] - \frac{1}{9}\rho u_y \tag{3.92}$$

$$N_z^x = \frac{1}{6}\big[f_5+f_{15}+f_{17}-(f_6+f_{16}+f_{18})\big] - \frac{1}{9}\rho u_z \tag{3.93}$$

Similar to the left boundary, the on-site velocity boundary conditions on other sides for the D_3Q_{27} lattice structure is given as:

2. **At the right boundary (i.e. x = n_x) as outlet**: we get

$$\begin{aligned}\rho = \frac{1}{u_x+1}\big[&f_3+f_4+f_5+f_6+f_{15}+f_{16}+f_{17}+f_{18}+f_{19}\\ &+2(f_1+f_7+f_8+f_9+f_{10}+f_{20}+f_{22}+f_{24}+f_{26})\big]\end{aligned} \tag{3.94}$$

gives

$$u_x = -1 + \frac{1}{\rho}\big[f_3 + f_4 + f_5 + f_6 + f_{15} + f_{16} + f_{17} + f_{18} + f_{19} \\ + 2(f_1 + f_7 + f_8 + f_9 + f_{10} + f_{20} + f_{22} + f_{24} + f_{26})\big] \tag{3.95}$$

And, the unknown distributions are:

$$f_2 = f_1 - \frac{4}{9}\rho u_x \tag{3.96}$$

And,

$$f_{12} = f_7 - \frac{\rho}{9}(u_y + u_x) + N_y^x \tag{3.97}$$

$$f_{11} = f_8 + \frac{\rho}{9}(u_y - u_x) - N_y^x \tag{3.98}$$

$$f_{14} = f_9 - \frac{\rho}{9}(u_z + u_x) + N_z^x \tag{3.99}$$

$$f_{13} = f_{10} + \frac{\rho}{9}(u_z - u_x) - N_z^x \tag{3.100}$$

$$f_{21} = f_{20} - \frac{\rho}{36}\big[u_x + (u_y + u_z)\big] + N_y^x + N_z^x \tag{3.101}$$

$$f_{23} = f_{22} - \frac{\rho}{36}\big[u_x + (u_y - u_z)\big] + N_y^x - N_z^x \tag{3.102}$$

$$f_{25} = f_{24} - \frac{\rho}{36}\big[u_x - (u_y - u_z)\big] - N_y^x + N_z^x \tag{3.103}$$

$$f_{27} = f_{26} - \frac{\rho}{36}\big[u_x - (u_y + u_z)\big] - N_y^x - N_z^x \tag{3.104}$$

3. **At the front boundary (i.e. y = 0) as inlet**: we obtain

$$\rho = \frac{1}{1 - u_y}\big[f_1 + f_2 + f_5 + f_6 + f_9 + f_{10} + f_{13} + f_{14} + f_{19} \\ + 2(f_4 + f_8 + f_{12} + f_{17} + f_{18} + f_{21} + f_{23} + f_{24} + f_{26})\big] \tag{3.105}$$

gives

$$u_y = 1 - \frac{1}{\rho}\big[f_1 + f_2 + f_5 + f_6 + f_9 + f_{10} + f_{13} + f_{14} + f_{19} \\ + 2(f_4 + f_8 + f_{12} + f_{17} + f_{18} + f_{21} + f_{23} + f_{24} + f_{26})\big] \tag{3.106}$$

And, the unknown density distribution functions are:

$$f_3 = f_4 + \frac{4}{9}\rho u_y \tag{3.107}$$

Also,

$$f_7 = f_{12} + \frac{\rho}{9}(u_y + u_x) - N_x^y \tag{3.108}$$

$$f_{11} = f_8 + \frac{\rho}{9}(u_y - u_x) + N_x^y \tag{3.109}$$

$$f_{15} = f_{18} + \frac{\rho}{9}(u_y + u_z) - N_z^y \tag{3.110}$$

$$f_{16} = f_{17} + \frac{\rho}{9}(u_y - u_z) + N_z^y \tag{3.111}$$

$$f_{20} = f_{21} + \frac{\rho}{36}\big[u_y + (u_x + u_z)\big] - N_x^y - N_z^y \tag{3.112}$$

$$f_{22} = f_{23} + \frac{\rho}{36}\big[u_y + (u_x - u_z)\big] - N_x^y + N_z^y \tag{3.113}$$

$$f_{25} = f_{24} + \frac{\rho}{36}\big[u_y - (u_x + u_z)\big] + N_x^y + N_z^y \tag{3.114}$$

$$f_{27} = f_{26} + \frac{\rho}{36}\big[u_y - (u_x - u_z)\big] + N_x^y - N_z^y \tag{3.115}$$

where

$$N_x^y = \frac{1}{6}\big[f_1 + f_9 + f_{10} - (f_2 + f_{13} + f_{14})\big] - \frac{1}{9}\rho u_x \tag{3.116}$$

$$N_z^y = \frac{1}{6}\big[f_5 + f_9 + f_{13} - (f_6 + f_{10} + f_{14})\big] - \frac{1}{9}\rho u_z \tag{3.117}$$

4. **At the back boundary (i.e. y =** n_y**) as outlet**: we have

$$\begin{aligned}\rho = \frac{1}{u_y + 1}\big[&f_1 + f_2 + f_5 + f_6 + f_9 + f_{10} + f_{13} + f_{14} + f_{19} \\ &+ 2(f_3 + f_7 + f_{11} + f_{15} + f_{16} + f_{20} + f_{22} + f_{25} + f_{27})\big]\end{aligned} \tag{3.118}$$

gives

$$\begin{aligned}u_y = -1 + \frac{1}{\rho}\big[&f_1 + f_2 + f_5 + f_6 + f_9 + f_{10} + f_{13} + f_{14} + f_{19} \\ &+ 2(f_3 + f_7 + f_{11} + f_{15} + f_{16} + f_{20} + f_{22} + f_{25} + f_{27})\big]\end{aligned} \tag{3.119}$$

And, the unknown fluid populations are:

$$f_4 = f_3 - \frac{4}{9}\rho u_y \tag{3.120}$$

Also,

$$f_{12} = f_7 - \frac{\rho}{9}(u_x + u_y) + N_x^y \tag{3.121}$$

$$f_8 = f_{11} + \frac{\rho}{9}(u_x - u_y) - N_x^y \tag{3.122}$$

$$f_{18} = f_{15} - \frac{\rho}{9}(u_z + u_y) + N_z^y \tag{3.123}$$

$$f_{17} = f_{16} + \frac{\rho}{9}(u_z - u_y) - N_z^y \tag{3.124}$$

$$f_{21} = f_{20} - \frac{\rho}{36}\big[u_y + (u_x + u_z)\big] + N_x^y + N_z^y \tag{3.125}$$

$$f_{23} = f_{22} - \frac{\rho}{36}\big[u_y + (u_x - u_z)\big] + N_x^y - N_z^y \tag{3.126}$$

$$f_{24} = f_{25} - \frac{\rho}{36}\big[u_y - (u_x + u_z)\big] - N_x^y - N_z^y \tag{3.127}$$

$$f_{26} = f_{27} - \frac{\rho}{36}\big[u_y - (u_x - u_z)\big] - N_x^y + N_z^y \tag{3.128}$$

5. **At the bottom boundary (i.e. z = 0) as inlet**:

   The density ρ expression is given by

$$\rho = \frac{1}{1-u_z}\big[f_1+f_2+f_3+f_4+f_7+f_8+f_{11}+f_{12}+f_{19} \\ +2(f_6+f_{10}+f_{14}+f_{16}+f_{18}+f_{21}+f_{22}+f_{25}+f_{26})\big] \tag{3.129}$$

   gives

$$u_z = 1-\frac{1}{\rho}\big[f_1+f_2+f_3+f_4+f_7+f_8+f_{11}+f_{12}+f_{19} \\ +2(f_6+f_{10}+f_{14}+f_{16}+f_{18}+f_{21}+f_{22}+f_{25}+f_{26})\big] \tag{3.130}$$

   Thus,

$$f_5 = f_6+\frac{4}{9}\rho u_z \tag{3.131}$$

   Also,

$$f_9 = f_{14}+\frac{\rho}{9}(u_z+u_x)-N_x^z \tag{3.132}$$

$$f_{13} = f_{10}+\frac{\rho}{9}(u_z-u_x)+N_x^z \tag{3.133}$$

$$f_{15} = f_{18}+\frac{\rho}{9}(u_z+u_y)-N_y^z \tag{3.134}$$

$$f_{17} = f_{16}+\frac{\rho}{9}(u_z-u_y)+N_y^z \tag{3.135}$$

$$f_{20} = f_{21}+\frac{\rho}{36}\big[u_z+(u_x+u_y)\big]-N_x^z-N_y^z \tag{3.136}$$

$$f_{23} = f_{22}+\frac{\rho}{36}\big[u_z-(u_x+u_y)\big]+N_x^z+N_y^z \tag{3.137}$$

$$f_{24} = f_{25}+\frac{\rho}{36}\big[u_z+(u_x-u_y)\big]-N_x^z+N_y^z \tag{3.138}$$

$$f_{27} = f_{26}+\frac{\rho}{36}\big[u_z-(u_x-u_y)\big]+N_x^z-N_y^z \tag{3.139}$$

   where

$$N_x^z = \frac{1}{6}\big[f_1+f_7+f_8-(f_2+f_{11}+f_{12})\big]-\frac{1}{9}\rho u_x \tag{3.140}$$

$$N_y^z = \frac{1}{6}\big[f_3+f_7+f_{11}-(f_4+f_8+f_{12})\big]-\frac{1}{9}\rho u_y \tag{3.141}$$

6. **At the top (i.e. z = n_z) boundary as outlet**: we have

$$\rho = \frac{1}{u_z+1}\big[f_1+f_2+f_3+f_4+f_7+f_8+f_{11}+f_{12}+f_{19} \\ +2(f_5+f_9+f_{13}+f_{15}+f_{17}+f_{20}+f_{23}+f_{24}+f_{27})\big] \tag{3.142}$$

This can be rewritten as:

$$u_z = -1 + \frac{1}{\rho}\big[f_1 + f_2 + f_3 + f_4 + f_7 + f_8 + f_{11} + f_{12} + f_{19} \\ + 2(f_5 + f_9 + f_{13} + f_{15} + f_{17} + f_{20} + f_{23} + f_{24} + f_{27})\big] \tag{3.143}$$

The unknown density distribution functions can be computed as:

$$f_6 = f_5 - \frac{4}{9}\rho u_z \tag{3.144}$$

Also,

$$f_{14} = f_9 - \frac{\rho}{9}(u_x + u_z) + N_x^z \tag{3.145}$$

$$f_{10} = f_{13} + \frac{\rho}{9}(u_x - u_z) - N_x^z \tag{3.146}$$

$$f_{18} = f_{15} - \frac{\rho}{9}(u_y + u_z) + N_y^z \tag{3.147}$$

$$f_{16} = f_{17} + \frac{\rho}{9}(u_y - u_z) - N_y^z \tag{3.148}$$

$$f_{21} = f_{20} - \frac{\rho}{36}\big[u_z + (u_x + u_y)\big] + N_x^z + N_y^z \tag{3.149}$$

$$f_{22} = f_{23} - \frac{\rho}{36}\big[u_z - (u_x + u_y)\big] - N_x^z - N_y^z \tag{3.150}$$

$$f_{25} = f_{24} - \frac{\rho}{36}\big[u_z + (u_x - u_y)\big] + N_x^z - N_y^z \tag{3.151}$$

$$f_{26} = f_{27} - \frac{\rho}{36}\big[u_z - (u_x - u_y)\big] - N_x^z + N_y^z \tag{3.152}$$

3.2.2 Bounce-Back Boundary Conditions

The BB boundary condition is the most commonly used boundary condition in LBM to obtain a no-slip condition at the boundary walls. The key idea is to BB fluid particles in the opposite direction once the fluid particles reach the boundary. There are several ways to implement the BB schemes at the boundary. These are explained as below:

3.2.2 Full way bounce-back

The scheme is also referred to as the standard BB method [Guo and Shu, 2013]. The full way BB method is the easiest to implement. In this method, the lattice points are located on the fluid boundary, and after the streaming process, the unknown fluid populations at the boundary nodes take the value of the fluid populations opposite in direction, as shown in Figure 3.3.

The method sees no collision operation on the fluid node at the wall boundary during its implementation. The method provides the first-order accuracy to the solution. Despite the popularity of this scheme due to its simplicity, the scheme is not flawless since it is not truly mass-conservative. The scheme can be defined mathematically as:

$$f_k(\vec{x}, t + \Delta t) = f_{\bar{k}}(\vec{x}, t + \Delta t) \tag{3.153}$$

where, f is the post-streaming state of the distribution function, and $(k, \bar{k})$ are the distribution functions (fluid population) in the direction opposite to each other.

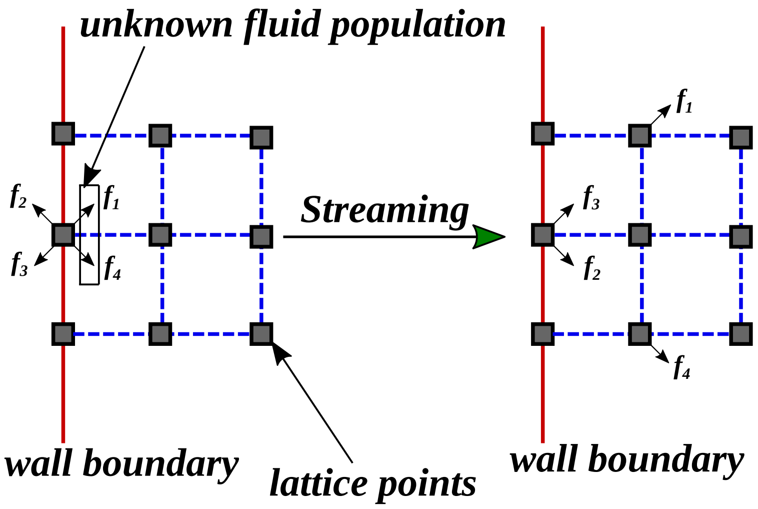

Figure 3.3 : 2D representation of the full way bounce-back method before (left) and after (right) streaming process.

3.2.2 Half way bounce-back

In this approach, wall boundary is simply placed at a half distance between two lattice sites. In other words, the wall boundary is located in between the ghost nodes and fluid nodes (or lattice points) as shown in Figure 3.4. The collision operation is performed on all lattice sites. The method is also called as bounce-back on the link (BBL) [Behrend, 1995]. Moreover, the significant features of this boundary condition that it can provide second order accuracy and also conserve the mass at the boundary wall. The scheme is defined with the expression given as:

$$f_k(\vec{x}, t + \Delta t) = f_{\bar{k}}(\vec{x}, t^+) \tag{3.154}$$

where $f_{\bar{k}}(\vec{x}, t^+)$ represents the post-collision distribution function.

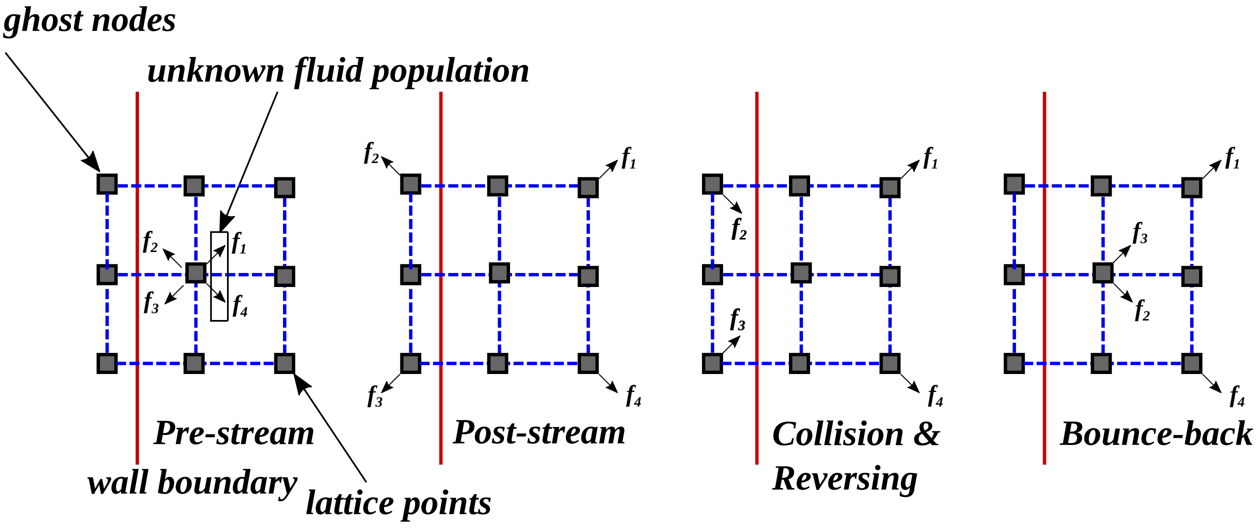

Figure 3.4 : 2D representation of the half way bounce-back method

3.2.2 Modified bounce-back

The implementation of the modified BB boundary condition is similar to the full-way bounce-back method. It differs only in the sense that it includes collision operation at the wall nodes during the process.

3.2.3 Free-Slip Boundary Condition

The boundary condition is also known as specular reflection. The boundary condition is generally used to the situation of a smooth boundaries of zero friction. In the case of free-slip boundary condition, the implementation is similar to that used in the full-way BB boundary condition. However, instead of performing the complete reflection, the outgoing particle distribution functions are reoriented via specular reflection as shown in Figure 3.5.

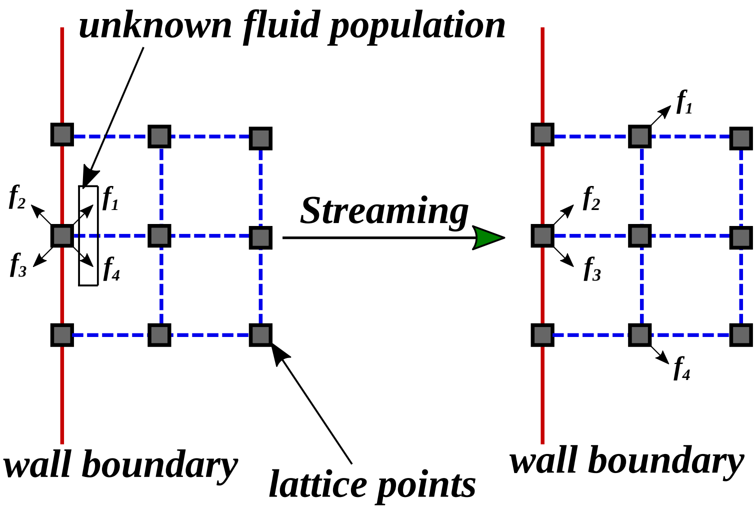

Figure 3.5 : 2D representation of the free-slip boundary condition before (left) and after (right) streaming process.

The mathematical formulation for the scheme can be written as:

$$f_k(\vec{x}, t + \Delta t) = f_{\bar{\bar{k}}}(\vec{x}, t + \Delta t) \tag{3.155}$$

where, $f_{\bar{\bar{k}}}$ is the specular reflection of the unknown fluid populations at the boundary nodes.

3.2.4 Immersed Boundary Method

The IB method can be described as a non-body conformal grid method for implementing no-slip conditions at the boundary surface by adding a forcing term to the governing flow equation either explicitly or implicitly. The concept behind the method is that the boundary nodes are represented as Lagrangian markers independent of the mesh used in flow solver [Peng and Luo, 2008] as shown in Figure 3.6. Additionally, the IB method needs interface schemes since the boundary nodes generally do not correspond to the Eulerian grid nodes. The most popular interface scheme for the IB method is the diffuse interface scheme. In the diffuse interface scheme, the Lagrangian markers are located on the boundary surface at which the boundary force is evaluated, and the Eulerian grid nodes represent the flow field. Hence, interpolation between the Lagrangian markers and Eulerian nodes is required to interpolate the fluid velocity at the neighboring Eulerian nodes on boundary nodes for calculating the force exerted by the flow on the boundary surface and for the distribution of boundary force on the neighboring Eulerian nodes. The distributed forces are added as an external force term in the governing fluid flow equations.

The governing LBE for the SRT-LBM in the presence of an external force, can be rewritten as [Kang and Hassan, 2011]:

$$f_k(\vec{x}+\vec{e}_k\Delta t, t+\Delta t) = f_k(\vec{x},t) - \frac{1}{\tau}[f_k(\vec{x},t) - f_k^{eq}(\vec{x},t)] + F_k(\vec{x},t)\Delta t \tag{3.156}$$

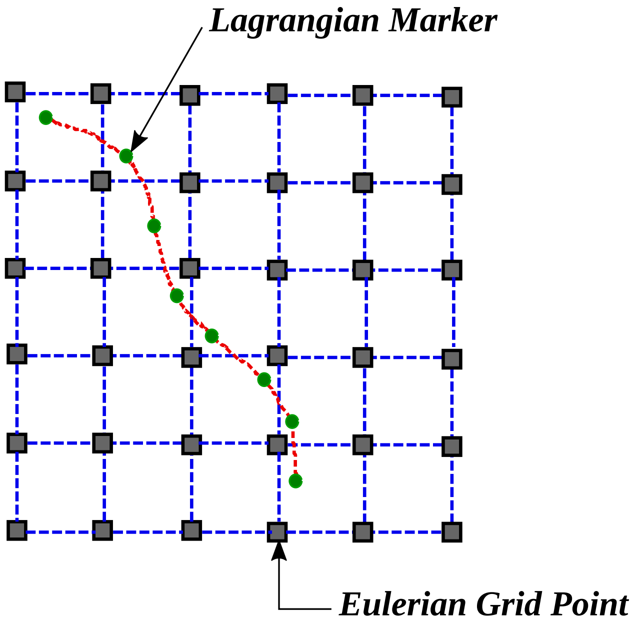

Figure 3.6 : Illustration of immersed boundary method.

where F_k is the discrete force distribution function along direction k, that can be computed with the given expression [Kang and Hassan, 2011]:

$$F_k(\vec{x},t) = \left(1-\frac{1}{2\tau}\right)w_k\left[\frac{\vec{e}_k - \vec{u}(\vec{x},t)}{c_s^2} + \frac{\vec{e}_k\cdot\vec{u}(\vec{x},t)}{c_s^4}\vec{e}_k\right]\cdot\mathbf{F}(\vec{x},t) \tag{3.157}$$

Moreover, the following steps are used in the numerical calculations to solve the LBE with a forcing term [Guo *et al.*, 2002; Kang and Hassan, 2011].

- **First-forcing step**:

$$\rho(\vec{x},t)\vec{u}(\vec{x},t) = \sum_k \vec{e}_k f_k(\vec{x},t) + \frac{\Delta t}{2}\mathbf{F}(\vec{x},t) \tag{3.158}$$

- **Collision step**:

$$f_k^{'}(\vec{x},t) = f_k(\vec{x},t) - \frac{1}{\tau}[f_k(\vec{x},t) - f_k^{eq}(\vec{x},t)] \tag{3.159}$$

- **Second-forcing step**:

$$f_k^{''}(\vec{x},t) = f_k^{'}(\vec{x},t) + \Delta t F_k(\vec{x},t) \tag{3.160}$$

- **Streaming step**:

$$f_k(\vec{x}+\vec{e}_k\Delta t, t+\Delta t) = f_k^{''}(\vec{x},t) \tag{3.161}$$

where $f_k^{'}$ and $f_k^{''}$ are defined as the post-collision and post-forcing distribution functions, respectively. However, the forcing term on the right hand side of Eq.(3.157) is the boundary force distribution on the neighboring Eulerian nodes. This can be computed with the following procedure:

i. Firstly, the unforced velocity $\vec{U}_b^{uf}$ at the boundary nodes $\vec{x}_b$ is obtained by interpolating the unforced fluid velocity $\vec{u}^{uf}$ from the neighboring Eulerian nodes. This can be given as:

$$\vec{U}_b^{uf} = \sum_{i,j,k} \vec{u}_{ijk}^{uf} D_f(\vec{x}_{ijk} - \vec{x}_b)\Delta x \Delta y \Delta z \tag{3.162}$$

where, D_f is the cheap-clipped polynomial mapping function [Deen *et al.*, 2004], $\vec{x}$ is the position vector of the Eulerian fluid nodes, and $\vec{x}_b$ represent the position vector of the control points on the boundary surface of the reactor components. The mathematical expression for the cheap-clipped polynomial mapping function can be written as:

$$D_f(\vec{x}_{ijk} - \vec{x}_b) = d\Big(x_i - x_b\Big) d\Big(y_j - y_b\Big) d\Big(z_k - z_b\Big) \tag{3.163}$$

And,

$$d(x_i - x_b) = \begin{cases} \frac{15}{16}\left[\frac{(x_i-x_b)^5}{n^5} - 2\frac{(x_i-x_b)^2}{n^3} + \frac{1}{n}\right] & -n \leq (x_i - x_b) \leq +n \\ 0, & otherwise \end{cases} \tag{3.164}$$

ii. Then, the forces on the boundary nodes $\vec{F}_b$ are obtained using the expression given below:

$$\vec{F}_b = 2\rho \frac{\vec{U}_b - \vec{U}_b^{uf}}{\Delta t} \tag{3.165}$$

where $\vec{U}_b$ is the velocity vector on the lagrangian markers at boundary surface.

iii. Finally, forcing term **F** on the right-hand side of equation Eq.(3.157) for the distribution of the boundary forces $\vec{F}_b$ on the neighboring Eulerian nodes can be computed as:

$$\mathbf{F} = \sum_b \vec{F}_b D_f(\vec{x}_{ijk} - \vec{x}_b)\Delta x \tag{3.166}$$

3.3 GPU PROGRAMMING

3.3.1 Architecture Details of NVIDIA Tesla GP100 GPU

The simulations for the present thesis work have been carried out on a Tesla P100 accelerator of NVIDIA GP100 GPU card.

As shown in Figure 3.7, the NVIDIA GP100 architecture contains Graphical Processing Clusters (GPCs), Texture Processing Clusters (TPCs), Streaming Multiprocessors (SMs), and memory controllers. A complete GP100 is made up of 6 GPCs, 30 TPCs, 60 Pascal SMs. It consists of eight memory controllers of 512 bits each. (total of 4096 bits). Each GPC comprises five TPCs, each of which includes two SMs. Each SM consists of 32 double-precision CUDA cores that is half of the single-precision CUDA cores and four texture units as illustrated in Figure 3.8. A complete GP100 GPU features has 1920 double-precision CUDA cores (also 3840 single precision CUDA cores) and 240 texture units over 60 SMs. Each memory controller has 512 KB of L2 cache linked to it. Morever, each HBM2 DRAM stack is managed by two memory controllers. The entire GPU is comprised of a total of 4096 KB of L2 cache. The Tesla P100 accelerator employs 56 SMs. For more details about the architecture of the Tesla GP100 card, one can refer to NVIDIA [2016]. The technical specifications of the NVIDIA Tesla GP100 GPU are given in Table 3.1.

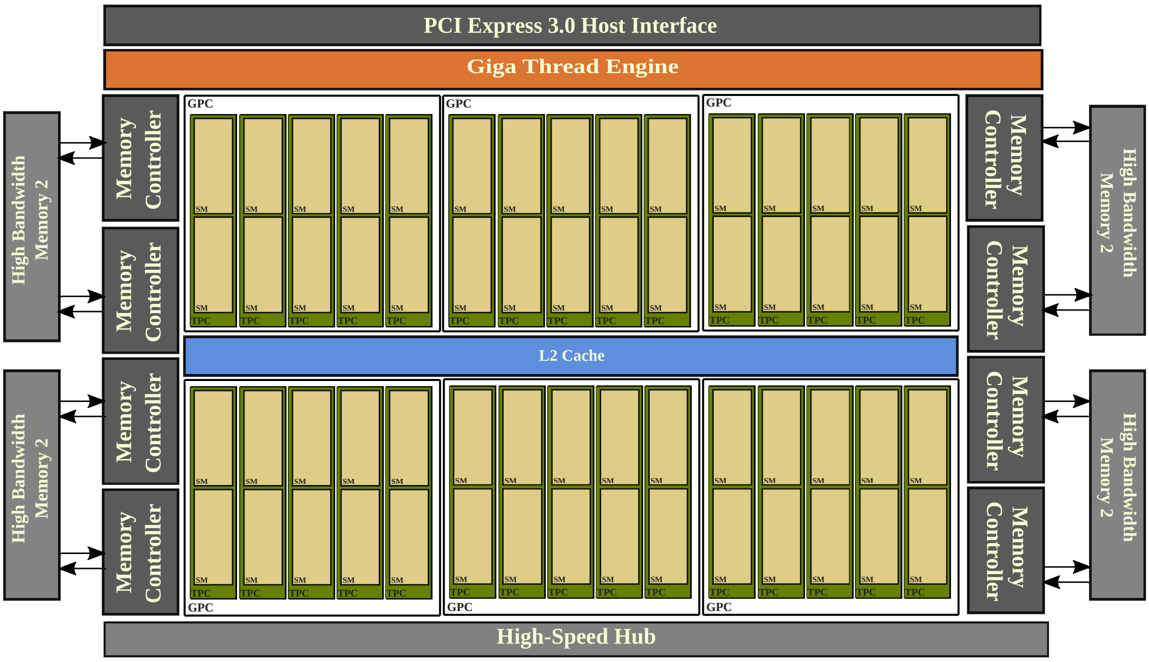

Figure 3.7 : Architecture of Tesla GP100.

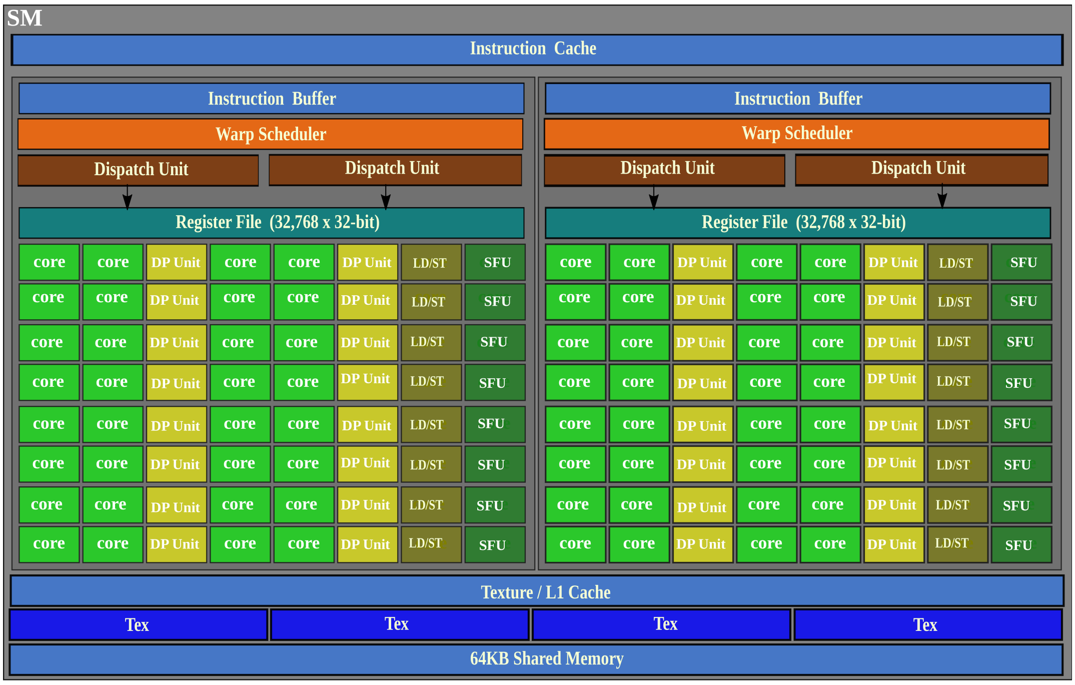

Figure 3.8 : Streaming Multiprocessor.

3.3.2 Structure of CUDA Programming

The CUDA programming model developed by NVIDIA has been used to develop the code suitable for parallel computation on GPU architecture [Guide, 2013; Cheng *et al.*, 2014]. The CUDA programming model represents CPU architecture as a host unit and GPU architecture as a device unit. CUDA programming model uses the kernel function for the execution of the program. The

Table 3.1 : NVIDIA Tesla GP100 specifications

GPU Architecture	NVIDIA Pascal
NVIDIA CUDA Cores	3840
Memory Size	16 GB
Memory Bandwidth	732 GB/s
Double-Precision Performance	4.7 TeraFLOPS
Single-Precision Performance	9.3 TeraFLOPS
Half-Precision Performance	18.7 TeraFLOPS

kernel function in the CUDA programming model is a call function. It is called from the host (i.e., CPU) and runs on the device (i.e., GPU), as shown in Figure 3.9. It executes the complete domain grid in the GPU, which has been divided into a large number of GPU blocks. Each GPU block consists of a large number of threads for parallel computation.

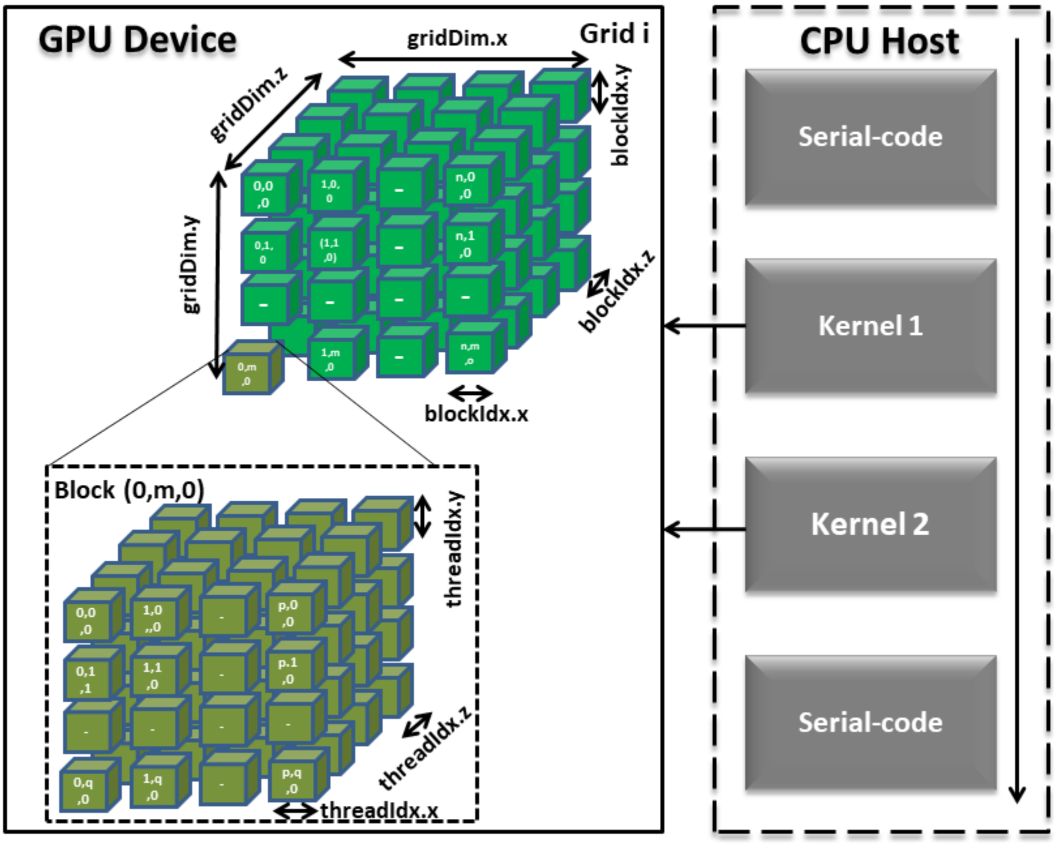

Figure 3.9 : Illustration of CUDA programming model in 3D.

Moreover, as illustrated in Figure 3.9, the meaning of the following terminologies are:

- *gridDim.x*, *gridDim.y*, and *gridDim.z* are the in-built commands used to define the number of blocks in the directions *x*, *y*, and *z*, respectively.

- *blockDim.x*, *blockDim.y*, and *blockDim.z* are the in-built commands used to provide block dimension, which means the number of threads each block contains in the direction *x*, *y*, and *z*, respectively.

- *blockIdx.x*, *blockIdx.y*, and *blockIdx.z* are the in-built commands that represents the block index (ID) in the *x*, *y*, and *z* direction, respectively.

- *threadIdx.x*, *threadIdx.y*, and *threadIdx.z* are the in-built commands for defining threads ID in each block in the *x*, *y*, and *z* direction, respectively.

The global thread ID in the particular direction can be calculated as:

- *i = blockIdx.x * blockDim.x + threadIdx.x;*

- *j = blockIdx.y * blockDim.y + threadIdx.y;*

- *k = blockIdx.z * blockDim.z + threadIdx.z;*

Furthermore, the block in the CUDA programming model runs parallel in different SMs, and the threads in each block run directly into CUDA cores. The important thing to understand is that the threads in each block can communicate with each other. However, threads in a different block cannot communicate. The more details can be found in Guide [2013].

3.3.3 LBM implementation on GPU cluster

The step by step procedure for the implementation of LBM algorithm on GPU architecture is given below:

- **step 1:** Allocate memory for the variables both on the host and device.

- **step 2:** Copy related variables from the host to device.

- **step 3:** Divide the domain into a combination of blocks and threads.

- **step 4:** Call the kernel function to perform different LBM operation, i.e, collision, streaming, boundary condition and updation of macroscopic variables.

- **step 5:** After completion of the simulation, transfer desired output files from device to host.

3.4 SUMMARY

In summary, this chapter conclude with the detailed description of the governing mathematical formulations of the popular SRT-LBM model for the different versions of 3D lattice structures. It has been described that the SRT-LBM model includes steps of collision and streaming in its expression. The LES turbulence model in the frame of LBM is described in detail, explaining the filtered equation of SRT-LBM to model the turbulent flow, along with the Smagorinsky SGS model to resolve the small-scale turbulent structures. The chapter provides the complete derivation of the incorporation of LES turbulence model in the LBE. The chapter also includes the description of the different types of boundary conditions in LBM, such as on-site velocity boundary condition, bounce-back method, and immersed boundary method. Lastly, the chapter illustrates the implementation of the LBM algorithm on the GPU platform.