Master of Science Thesis

IMMISCIBLE TWO PHASE FLOWS USING LATTICE BOLTZMANN METHOD

By

Ganesh Chithambalam Anga Sivagurudassprakash

Thesis submitted to Swansea University in Partial fulfillment for the Degree of Master of Science in Computational Mechanics

Supervisor:Professor Yuntian Feng

Civil and Computational Engineering Centre
Swansea University, Swansea
United Kingdom

June 2011

DECLARATION AND STATEMENTS

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Candidate:_____

Date:_____

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Candidate:_____

Date:_____

Statement 2

I am hereby consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Candidate:_____

Date:_____

SUMMARY

This thesis focuses on the implementation of the lattice Boltzmann method for simulating incompressible immiscible two phase flow problems. This implementation was carried out in a sequential manner. First the single phase lattice Boltzmann model was implemented and the code was tested for stability and accuracy by simulating a number of bench mark problems like Poiseuille flow and lid driven cavity flow. After the stability of the single phase flow model was ascertained the single phase lattice Boltzmann implementation was then extended to a framework for simulating incompressible two phase flow problems. The implemented two phase flow model was then subjected to numerical simulation of a number of two phase flow problems to study its accuracy and stability. It was observed that the implemented code was successfully in simulating a wide variety of immiscible two phase flow problems.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to express my profound gratitude to Professor Yuntian Feng for being my thesis supervisor. He has been very helpful in guiding me and encouraging me to learn new things, without which, this thesis would not have been possible.

I am also thankful to European Commission for sponsoring this Erasmus Mundus Master Program and also to International Centre for Numerical Methods in Engineering (CIMNE), Barcelona for forming such wonderful curriculum.

I am thankful to Dr. Antonio J. Gil being for all the help and support provided during the Masters course.

I would like to express my gratitude to Bruce Jones and Shankar for being part of lengthy discussions and providing valuable guidance and suggestions during the period of my thesis.

I would also like to thank my course mates Venkatesh and Rodolfo and other friends for making my stay in Swansea memorable.

I owe my loving thanks to my family and friends. Without their encouragement and understanding it would have been impossible for me to finish this work.

LIST OF FIGURES

LIST OF SYMBOLS

$\mathbf{v}$	Velocity
$\mathbf{x}$	Position Vector
$\mathbf{p}$	Momentum
Av	Avogadro Number
n_i	Particle population in LGCA
$\langle\ \rangle$	Ensemble of particles
f_i	Particle distribution functions
Δt	Time step
Ω_i	Collision Operator
f^+	Post collision distribution function
ρ	Density
$\rho\mathbf{u}$	Momentum flux
c	Lattice speed
c_s	Speed of sound
M_{ij}	Collision matrix
h	Lattice width
Ma	Mach Number
ν	Viscosity
$\bar{h}$	Lattice spacing in lattice units
ρ_0	True fluid density
G	Body force
σ	Two phase collision operator
$\Omega_i^{\sigma}2$	Two phase surface tension operator
$\Omega_i^{\sigma}3$	Recolouring Operator
σ	Surface Tension

LIST OF ABBREVIATIONS

MAC	Marker and Cell Method
VOF	Volume of Fluid Method
BEM	Boundary Element Method
LBM	Lattice Boltzmann Method
NS	Navier-Stokes Equations
CFD	Computational Fluid Dynamics
BE	Boltzmann Equation
RHS	Right Hand Side
LGCA	Lattice gas cellular automaton

CHAPTER 1

INTRODUCTION

Multiphase flows play an important role in many natural processes and engineering applications. They occur in a variety of environmental phenomena like rain, fog, snow, avalanches, soil erosion, avalanches, landslides, etc. Very critical biological and medical flows like blood flow are multiphase flows. Virtually every processing technology deals with multiphase flows [29]. They find their application in nuclear power plants, combustion engines, propulsion systems, blood flows inside human body, chemical industry, oil and gas production and transportation, biological industry, food production, etc. In all these applications, it is important to predict the multiphase behaviour. Thus it is very essential to understand the principles of multiphase flow for both fundamental research and engineering applications

A common feature in all multiphase flow problems is the existence of distinguishable interfaces, or boundaries, that separate one phase from another and also the discontinuities of the associated properties at the interface. The topology of the interface constantly changes as the phases interact with each other exchanging energy, momentum and mass (where phase change is involved).Also, a wide variety of multiphase flow patterns were observed by different researchers. Shad *et al* [30] suggested that 84 different flow patterns exist in multiphase flow problems when they performed a survey of different flow patterns in multiphase flow. These flow patterns are highly dependent on different flow parameters like fluid properties, geometry of the flow channel, etc. The rate of exchange of mass, momentum and energy between different phases and also between the fluids and the external boundaries are dependent on these flow patterns. Thus the study and prediction of multiphase flow behaviour is challenging due to the diverse nature of these flow patterns.

The main focus in the study of multiphase flows is the need to model and predict the detailed behaviour of those flows and the phenomena that they manifest as the accuracy and consistency of the model directly influences the design process. Although theoretical and experimental approaches are available, they are not viable for a majority of multiphase flow problems due their complex nature. Thus computational approaches serve as valuables tools for predicting and understanding multiphase flow simulations.

1.1 Approaches used in two phase flow simulation

The traditional methods for simulating two phase flow problems are based on solving either the differential or integral form of partial differential equations. These approaches are called as "top down" approaches. The governing partial differential equations are discretised by finite difference, finite volumes or finite element methods and the solutions are obtained on the discretised spatial and temporal scales. These traditional methods can be classified into two categories: the front capturing method and the front tracking method. Each of these methods is briefly reviewed.

1.1.1 Front Capturing Method

In the front capturing method, the movement of the fluid is tracked first and the interface is captured afterwards. The two fluids are considered as a single continuum with discontinuous features at the interface. The same Eulerian mesh is used for solving the fluid flow equations of both the fluids. Based on how interface propagation is captured, there are three types of front capturing methods: the Marker-and-Cell (MAC) method [31], Volume-of-Fluid (VOF) method [32], and level set method [33].

The Marker-and-Cell (MAC) method makes use of Lagrangian markers to represent the location of a particular phase. The interface is then constructed by using the location details of the markers. In order to track the interface accurately, a large number of markers are required, thus making the MAC method computationally expensive.

The objective of introduction of Volume-of-Fluid (VOF) method was to reduce the heavy computational load of the MAC method [32]. In the VOF method, the volume fraction of fluid in each cell is used to track the movement of the liquid instead of tracking a large number of markers. Even though the computational costs reduced greatly, determining the exact location of the interface was still difficult.

In the level set method two different sets of equations are used to model the two-phase flow [33]. The first set of equations contains the single fluid Navier-Stokes equations. The second set is a transient scalar advection equation which tracks a level set. The level set functions are defined in such a way that its value is equal to zero at the interface, negative in one phase and positive in other phase. The interface location is determined by interpolating between the level set function values. The advantage of the level set method is that it is easier to track the interface compared to the MAC and VOF method. However, the advection of the level set function is not based on a strictly conservative equation and thus mass is not conserved properly in the level set method.

1.1.2 Front Tracking Methods

In the front-tracking method, the location of the interface is tracked directly. It is therefore possible to calculate the curvature of the interface more accurately [35]. There are three commonly used front-tracking methods: the boundary-fitted grid method [37], Tryggvassions's hybrid method [35], and Boundary Element Method (BEM) [36].

Two sets of Navier-Stokes equations are solved in the boundary-fitted grid method, each equation corresponding to one fluid. The mesh in the computational domain is constructed in such a manner that the interface between two phases is located along a grid line and the movement of the interface is determined by a force balance.

Two sets of grids are used in the hybrid method proposed by Tryggvason [35]. A stationary grid is used to determine the fluid flow and a lower dimension grid used to track the interface.

In the Boundary Element Method (BEM), a multitude of boundary nodes are employed to represent the two-phase interface [36]. The movement of these boundary nodes is based on the potential function equations.

The front-tracking and front-capturing methods are conceptually simple. However it is more difficult to implement these methods in two-phase flow simulations because a lot of difficulties arise from the interface deformation and interaction. Also tracking a lot of interfaces is computationally expensive.

1.1.3 Lattice Boltzmann Two Phase Models

Unlike the traditional CFD method, the lattice Boltzmann method is based on a "bottom-up" approach. The lattice Boltzmann method solves mesoscopic equations (such as the Boltzmann equation) for an ensemble-averaged distribution of moving, interacting fluid particles on a discrete lattice. The desired macroscopic partial differential equations are then recovered using multi-scale analysis. The lattice Boltzmann method for simulating two phase flow is discussed in great detail in the following chapters.

1.2 Objectives of the Research

The aim of this research is to implement a lattice Boltzmann method (LBM) framework for simulating immiscible two phase flows. This framework is then to be applied to a variety of two phase flow problems. The development of this framework can be split into the following components.

1) Implement the single phase lattice Boltzmann method and validate the performance of code by simulating benchmark problems.
2) Research and implement the immiscible two phase lattice Boltzmann framework by extending the single phase lattice Boltzmann code with the principles of two phase flow.
3) Validate the newly implemented two phase immiscible lattice Boltzmann method code by testing it against a number of two phase flow problems.

1.3 Document Layout

This thesis is divided into seven chapters. Following this introductory chapter, Chapter 2 discusses the theory of lattice Boltzmann method. The evolution of the lattice Boltzmann method from the kinetic theory of gases is discussed and the various principles of the numerical model are outlined.

Chapter 3 discusses the principles behind various initial and boundary conditions and the technique for incorporating body forces in LB simulations is described.

Chapter 4 discusses the numerical validation of the implemented lattice Boltzmann code by simulating a number of benchmark single phase flow problems. The problems implemented were 2D Poiseuille flow and 2D cavity flow.

Chapter 5 discusses the development of a colour gradient based lattice Boltzmann model to simulate incompressible immiscible two phase flows. The algorithm for interface tracking and the various parameters controlling the surface tension at the interface are outlined.

Chapter 6 discusses the numerical validation of the implemented immiscible two phase lattice Boltzmann code. The implemented code is subjected to a number of tests by simulating various cases of two phase flow problems.

Chapter 7 discusses and offers concluding remarks on the results of single phase and two phase flow analyses. The scope for additional research, improvement and further development are outlined.

CHAPTER 2

THEORY OF LATTICE BOLTZMANN METHOD

The phenomenon of fluid flow can be studied at different spatial and temporal scales. At the macroscopic level a fluid can be characterized in terms of continuous hydrodynamic quantities such as velocity and density fields whereas at the microscopic [4] level a fluid can be represented as a system of a large number of molecular particles interacting in a complicated way. In general, the details of interaction at the microscopic level do not affect the behaviour at the macroscopic level.

The fundamental principle of LBM is to construct a simplified kinetic model which exhibits behaviour identical to classical hydrodynamic equations at the macroscopic level [2] by incorporating mesoscopic processes. This approach can be justified because of the fact that collective behaviour of many microscopic particles result into macroscopic behaviour and the details of the microscopic interactions are not essential [3] for majority of fluid flow problems. Thus the process of following each particle in molecular dynamic simulations and the process of solving complicated kinetics equations like the Boltzmann equation can be avoided. Even though the LBM is based on a microscopic picture, its principal focus is the averaged macroscopic behaviour. The macroscopic Navier-Stokes (NS) equation in the near-incompressible limit can be obtained from the LBM. Also pressure can be obtained from the equation of state [2]. The kinetic nature inherently provides many advantages of molecular dynamics, including clear physical pictures, easy implementation of boundary conditions, and fully parallel algorithms.

The lattice Boltzmann method historically originated as an extension of lattice gas cellular automata (LGCA) to overcome the various shortcomings like

statistical noise, etc. The lattice gas cellular automaton and lattice Boltzmann method can be both considered as simplifications to the kinetic theory of gases originally introduced by Ludwig Boltzmann [17]. The basic idea of Boltzmann's work was that a gas is composed of interacting particles that can be described by classical mechanics, and, because there are so many particles, a statistical treatment is necessary and appropriate. The mechanics of interacting particles can be extremely simplified and encapsulated by utilizing the notions of streaming in space and billiard-like collision interactions. The LB models simplify even further and yet, like lattice gas models, still reproduce the behaviour of real fluids. The basic concept of kinetic theory of gases and statistical mechanics and a simplified form of the Boltzmann equation are described in the following section.

2.1 Kinetic Theory of Gases

Consider a dilute gas consisting of hard spherical particles moving with some arbitrary velocity $\mathbf{v}$. It is assumed that the particles interact among them only through elastic collisions. If we know the position vector $\mathbf{x}$ and momentum $\mathbf{p}$ of each of the particle at some instant of time t, the exact dynamical state of the system can be determined which, together with classical mechanics, would allow exact predictions of all future states. Let $f^N\left(\mathbf{x}^N, \mathbf{p}^\mathbf{N}, t\right)$ be one such function describing the system where N is the number of particles. This function also known as the distribution function can be thought as residing in a 'phase space', which is a space in which the coordinates are position and momentum vectors and time. The time evolution of phase space distribution function is given by the Liouville equation which introduces 6N functions of time. This description of the distribution function is unviable because N is generally of the order of the Avogadro number $Av \sim 10^{23}$, far too big for any foreseeable computer. Therefore it appears wise to approach the collective behaviour of ensemble of molecules from a statistical point of view. This can be done at various levels of complexity. But the simplest one: the $one-body$ kinetic level is considered here.

The distribution $f^1(\mathbf{x}, \mathbf{p}, t)$ gives the probability of finding a particular molecule with a given position and momentum. The positions and momenta of the

remaining N-1 molecules can be ignored. The single particle distribution function f^1 is sufficient for describing all gas properties which do not depend on relative positions of molecules. The probable number of molecules with position coordinates in the range $\mathbf{x} \pm d\mathbf{x}$ and momentum coordinates $\mathbf{p} \pm d\mathbf{p}$ can be determined by $f^1(\mathbf{x},\mathbf{p},t)\,dxd\mathbf{p}$ .It is assumed that an external force $\mathbf{F}$ that is small relative to the intermolecular forces is acting on the system. If there are no collisions, then at time $t + dt$ the new positions of the molecules starting at $\mathbf{x}$ are given by

$$\mathbf{x} + (\mathbf{p}/m)\,dt = \mathbf{x} + (d\mathbf{x}/dt)\,dt = \mathbf{x} + d\mathbf{x} \tag{2.1}$$

The new momenta are given by

$$\mathbf{p} = \mathbf{p} + \mathbf{F}dt = \mathbf{p} + (d\mathbf{p}/dt)\,dt = \mathbf{p} + d\mathbf{p} \tag{2.2}$$

Thus, when there are no collisions and when the positions and the momenta are known at a particular time t , $f^{(1)}$ at time $t + dt$ can be determined using

$$f^{(1)}(\mathbf{x} + d\mathbf{x}, \mathbf{p} + d\mathbf{p}, t + dt)d\mathbf{x}d\mathbf{p} = f^{(1)}(\mathbf{x},\mathbf{p},t)\,d\mathbf{x}d\mathbf{p} \tag{2.3}$$

This is the streaming process. However, in reality there are collisions occurring in the system. Due to these collisions some phase points starting at $(\mathbf{x},\mathbf{p})$ do not arrive at $(\mathbf{x} + d\mathbf{x}, \mathbf{p} + d\mathbf{p})$. Also there are some molecules not starting at $(\mathbf{x},\mathbf{p})$ arriving at $(\mathbf{x} + d\mathbf{x}, \mathbf{p} + d\mathbf{p})$. Let $\mathbf{\Gamma}^{(-)}d\mathbf{x}d\mathbf{p}dt$ and $\mathbf{\Gamma}^{(+)}d\mathbf{x}d\mathbf{p}dt$ be the number of molecules that do not arrive at the expected position $(\mathbf{x} + d\mathbf{x}, \mathbf{p} + d\mathbf{p})$ and the number of molecules that start somewhere else other than $(\mathbf{x},\mathbf{p})$ and arrive at $(\mathbf{x} + d\mathbf{x}, \mathbf{p} + d\mathbf{p})$ respectively due to collisions occurring during time interval dt . Thus after considering the effects of collision Equation (2.3) can be modified and written as,

$$\begin{aligned} &f^{(1)}(\mathbf{x} + d\mathbf{x}, \mathbf{p} + d\mathbf{p}, t + dt)d\mathbf{x}d\mathbf{p} = \\ &f^{(1)}(\mathbf{x},\mathbf{p},t)\,d\mathbf{x}d\mathbf{p} + \left[\mathbf{\Gamma}^{(+)} - \mathbf{\Gamma}^{(-)}\right]d\mathbf{x}d\mathbf{p}dt \end{aligned} \tag{2.4}$$

The first order terms of a Taylor series expansion of the LHS of Equation (2.4) is given by

$$f^{(1)}(\mathbf{x}+d\mathbf{x},\mathbf{p}+d\mathbf{p},t+dt)=$$
$$f^{(1)}\left(\mathbf{x},\mathbf{p},t\right)+d\mathbf{x}\cdot\nabla_{\mathbf{x}}f^{(1)}+d\mathbf{p}\cdot\nabla_{\mathbf{p}}f^{(1)}+\left(\frac{\partial f^{(1)}}{\partial t}\right)dt+\cdots, \tag{2.5}$$

Thus combining Equations (2.4) and (2.5), it can be written as

$$\begin{aligned}&\left[f^{(1)}\left(\mathbf{x},\mathbf{p},t\right)+d\mathbf{x}\cdot\nabla_{\mathbf{x}}f^{(1)}+d\mathbf{p}\cdot\nabla_{\mathbf{p}}f^{(1)}+\left(\frac{\partial f^{(1)}}{\partial t}\right)dt+\cdots\right]d\mathbf{x}d\mathbf{p}\\&=f^{(1)}\left(\mathbf{x},\mathbf{p},t\right)d\mathbf{x}d\mathbf{p}+\left[\mathbf{\Gamma}^{(+)}-\mathbf{\Gamma}^{(-)}\right]d\mathbf{x}d\mathbf{p}dt\end{aligned} \tag{2.6}$$

Simplifying Equation (2.6) it can be written as

$$\mathbf{v}\cdot\nabla_{\mathbf{x}}f^{(1)}+\mathbf{F}\cdot\nabla_{\mathbf{p}}f^{(1)}+\frac{\partial f^{(1)}}{\partial t}=\mathbf{\Gamma}^{(+)}-\mathbf{\Gamma}^{(-)} \tag{2.7}$$

The Equation (2.7) is the Boltzmann equation (BE). The RHS of the Boltzmann equation is the collision operator which in its complete form is a complex nonlinear integral differential equation. With LGCA and the LBM the Boltzmann equation is approximately solved from the particle perspective and focus to obtain an equation similar to Equation (2.4) which explicitly contains the 'collision' and 'streaming' operations.

2.2 Lattice Gas Cellular Automaton (LGCA)

A cellular automaton (CA) is a discrete model which comprises a regular grid of cells called as the lattice. Each of the sites in the lattice can take certain number of different states. The lattice grid can be in any finite number of dimensions. The neighbourhood of each lattice node is defined relative to the specified node. In LGCA the various states are particles $n(\mathbf{x},t)$ with certain velocities, $\mathbf{x}$ being the lattice site location. The evolution of simulation is carried out in discrete time steps. The state at a given site after each time step can be determined from the state of the site itself and the neighbouring sites, at the previous time step. In LGCA the state at each site is purely Boolean (i.e. *true* if a particle is present and *false* if it is not). Two processes are carried out after each time step, propagation and collision. Each particle will move to the neighbouring step in the propagation step determined by the current velocity of the particle. In the collision step, collision rules are used to

determine what happens if multiple particles reach the same site at the same time. The stream-collision evolution of the particle populations can be written as

$$n_i(\mathbf{x}+\mathbf{c_i}, t+1) = n_i(\mathbf{x}, t) + \Omega_i(n(\mathbf{x}, t)) \qquad (i = 0,1,2,.....M) \tag{2.8}$$

Where $\mathbf{x}$ is the position of the lattice node, $\mathbf{c_i}$ is the particle velocity; Ω_i is the collision operator and M is the number of velocity directions at each lattice node. Macroscopic quantities like density, momentum can be calculated by summing up the particles at each site and multiplying the particle velocities before summing up respectively.

The LGCA method has its own set of advantages and disadvantages. The main assets were that the states of the model are Boolean in nature which enabled exact computing without any round-off error due to floating point precision. Also the LGCA meant that the simulations could be run using parallel computing. The development of the Lattice Boltzmann method (LBM) was motivated by the characteristic shortcomings of LGCA such as the large noise-signal ratio, non-Galilean invariance and the unphysical dependence of pressure on the velocity field. Another limitation of LGCA is the transport coefficients (i.e. viscosity) that emerge from the microscopic collision operators are of very limited range [8].

2.3 Principles of Lattice Boltzmann Method

The fundamental feature of LBM is that the Boolean particle occupation variables $n_i(\mathbf{x}, t)$ in the LGCA method is replaced by real valued particle distribution functions, $f_i(\mathbf{x}, t) = \langle n_i(\mathbf{x}, t) \rangle$ [9], where $\langle\ \rangle$ denotes an ensemble average. This procedure helps to eliminate statistical noise in LBM. These particle distribution functions exist at each of the lattice grid node that make up the fluid domain and form the primary variables in LBM unlike density and velocity in macroscopic fluid dynamics. The particle distribution functions represent the probable amount of fluid particles at each node, moving with a discrete speed in a discrete direction at each time step.

2.3.1 LB Formulation

The central issue to the LB formulation [5] is to define the rule that governs the movement of fluid particles in the lattice at discrete time instances. The lattice Boltzmann formulation of evolution of particle distribution function is given by

$$f_i(\mathbf{x}+\mathbf{e_i}\Delta t, t+\Delta t) = f_i(\mathbf{x},t) + \Omega_i(f(\mathbf{x},t)) \qquad (i = 0,1,2,.....M) \tag{2.9}$$

where $\mathbf{x}$ is the position, h is the lattice spacing, Δt is the explicit time step, $\mathbf{e_i}$ is the velocity along direction i and $\Omega_i(f(\mathbf{x},t))$ is the collision operator controlling the relaxation rate of the particle distribution functions that meet at a node and M is the number of velocity directions at each lattice node. The discretisation of space in LBM is performed in a manner which is consistent with the kinetic equation such that the particle distribution function velocities are parallel to the locations of the neighbouring nodes. Therefore, the neighbours to node $\mathbf{x}$ can be defined as $\mathbf{x}+\mathbf{c_i}\Delta t$. The solution of LBM advances via a two stage process at each time step. Collision (also known as relaxation) redistributes the functions that arrive at each node and then streaming (also known as convection) propagates the redistributed functions to their nearest neighbouring nodes.

The process of collision and streaming are shown in Figure (2.1). The image to the left shows the post collision particle distribution functions.

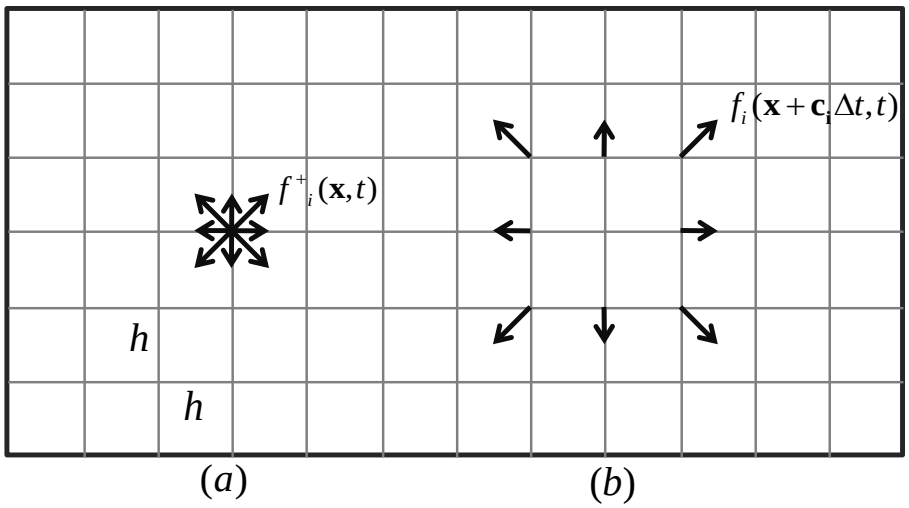


Figure 2.1: *A simple enclosed domain discretised by the lattice Boltzmann method showing the particle distribution functions after collisions (a) which are then streamed (b) to their respective neighbouring nodes.*

The functions then stream to their respective neighbouring nodes as shown in the image to the right.The lattice Boltzmann Equation (2.9) can be written in terms of the post-collision distribution function given by,

$$f_i^+(\mathbf{x},t) = f_i(\mathbf{x},t) + \Omega_i(f(\mathbf{x},t)) \tag{2.10}$$

Then Equation (2.9) can be simplified as,

$$f_i(\mathbf{x}+\mathbf{c_i}\Delta t, t+\Delta t) = f_i^+(\mathbf{x},t). \tag{2.11}$$

Therefore, it can be seen that no additional calculations are required in streaming process. The post collision distribution functions are just shifted to the neighbouring nodes. The macroscopic fluid variables like density ρ, and momentum flux, $\rho\mathbf{u}$ can be calculated at each lattice node as velocity moments of the particle distribution functions given by,

$$\rho = \sum_i f_i \tag{2.12}$$

$$\rho\mathbf{u} = \sum_i f_i \mathbf{c_i} \tag{2.13}$$

The isothermal equation of state is used to calculate the pressure directly from density,

$$p = c_s^2 \rho \tag{2.14}$$

where $c_s = c/\sqrt{3}$ is the lattice speed of sound.

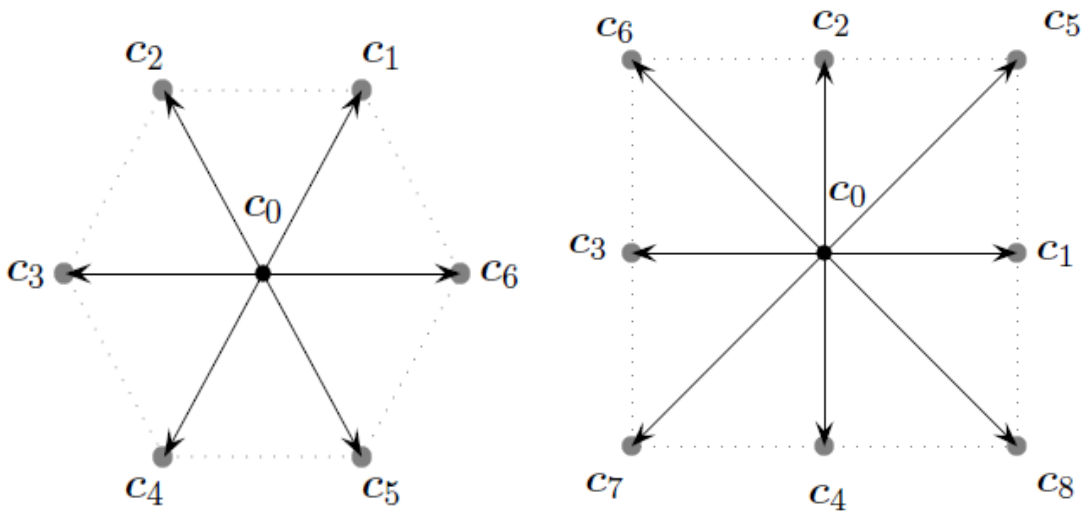


Figure 2.2: *Popular2D LB discretisation methods*

2.3.2 Domain Discretisation

The lattice gas models [6] required certain minimum lattice symmetry to ensure the isotropy of the velocity tensors and recover the Navier-Stokes equation from the kinetic model. The LBM also has the same symmetry requirement. Periodic arrays of polyhedra, both in two and three dimensions have been traditionally used as the lattice structure with increasing number of velocities and symmetry. But the benefits of increased symmetry can increase the associated computational cost. A wide family of lattice structures dubbed as $DnQm$ for m speed model in n directions is available. Popular examples for two dimensional simulations are shown the Figure (2.2). The $D2Q9$ discretisation scheme is used in the present work considering the accuracy and computational cost factors mentioned above.

2.3.2.1 D2Q9 Discretisation

Consider a 2D incompressible fluid flow domain with density ρ and kinematic viscosity ν, in a rectangular domain $\mathbb{R}$. The domain is divided into a regular grid, or lattice with spacing h in both x and y directions, as shown in Figure (2.3a). The fluid phase is represented as a collection of fluid 'particles' residing at lattice nodes[5] moving to neighbouring nodes along a fixed set of discrete directions with given discrete velocities at discrete time steps. The D2Q9 model, shown in Figure (2.3b) is a widely used 2D LB discretisation scheme. In this scheme the fluid particles at each node are allowed to move to its eight immediate neighbours with eight different velocities $\mathbf{c}_i, (i = 1,...,8)$.

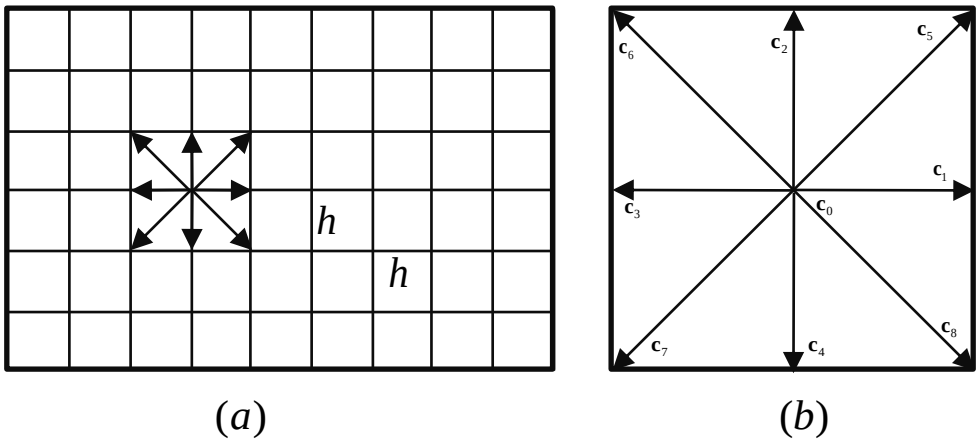


Figure 2.3: *a) Standard 2D LB lattice. b) D2Q9 model*

A certain proportion of particles at each node can be at rest, thus having zero velocity $\mathbf{c_0}$. Following the numbering system shown in Figure (2b), these nine discrete velocity vectors can be defined as

$$\mathbf{c_i} = \begin{cases} 0 & (i = 0) \\ C(\cos\frac{\pi(i-1)}{2}, \sin\frac{\pi(i-1)}{2}) & (i = 1,...,4) \\ C(\cos\frac{\pi(2i-9)}{4}, \sin\frac{\pi(2i-9)}{4}) & (i = 5,...,8) \end{cases} \tag{2.15}$$

where C is the lattice speed and given by

$$C = h/\Delta t \tag{2.16}$$

2.3.3 The Collision Process

In the LGCA, the collision process was carried out by a set of scattering rules that listed probabilities of every output state for each input state. The handling of these rules was costly from a computational point of view. Higuera and Jiménez [10] proposed a simplified Lattice Boltzmann Method with linearised collision operator in Equation (4) and (5) is given by,

$$\Omega_i(f(\mathbf{x},t)) = -M_{ij}(f(\mathbf{x},t) - f^{eq}(\mathbf{x},t)) \tag{2.17}$$

Where M_{ij} is the collision matrix, $f^{eq}(\mathbf{x},t)$ is the equilibrium distribution function. It is assumed that the particle distribution functions are never far from their equilibrium state, even for dynamic flows. The dimension of the square collision matrix is equal to the number of lattice functions. Also the components depend purely on the angle between the colliding functions. The collision operator corresponds to the action of hydrodynamic viscosity in lattice Boltzmann method [1]. From Equation (2.17) it can be seen that the collision process drives the non-equilibrium part of the particle distribution function at a node to equilibrium. Also the collision process is local. Therefore, only the distribution functions arriving at a particular node are considered in the collision operation. This local nature of the collision process is an important attribute for the parallel nature of the LBM.

2.3.3.1 Lattice Bhatnagar-Gross-Krook Collision Model

The lattice Bhatnagar-Gross-Krook [LBGK] model proposed by a number of authors [11, 12, 13] simplified the components of the collision matrix M_{ij} assuming that the collision operator relaxes the local particle distribution at a single rate and is given by,

$$M_{ij} = -\frac{1}{\tau}\delta_{ij} \tag{2.18}$$

The LBGK collision operator is written as,

$$\Omega_i(f(\mathbf{x},t)) = -\frac{\Delta t}{\tau}(f(\mathbf{x},t) - f^{eq}(\mathbf{x},t)) \tag{2.19}$$

The relaxation parameter τ controls the rate at which the particle distribution relaxes towards the equilibrium by operating directly on the non-equilibrium function. The kinematic viscosity of the fluid ν can be implicitly determined by the discretisation parameters h, Δt and τ as

$$\nu = \frac{1}{3}\left(\tau - \frac{1}{2}\right)\frac{h^2}{\Delta t} = \frac{1}{3}\left(\tau - \frac{1}{2}\right)Ch \tag{2.20}$$

2.3.4 The Equilibrium Function

The equilibrium distribution functions f^{eq} used in the collision process in section (1.3.1) is given by the following relations for a $D2Q9$ discretisation scheme:

$$\begin{aligned} f_0^{eq} &= w_0\rho(1 - \frac{3}{2C^2}\mathbf{v}.\mathbf{v}) && (i = 0) \\ f_i^{eq} &= w_i\rho(1 + \frac{3}{C^2}\mathbf{e_i}.\mathbf{v} + \frac{9}{2C^4}(\mathbf{e_i}.\mathbf{v})^2 - \frac{3}{2C^2}\mathbf{v}.\mathbf{v}) && (i = 1,...,8) \end{aligned} \tag{2.21}$$

In which w_i are weighting factors:

$$w_o = \frac{4}{9} \quad w_{1,2,3,4} = \frac{1}{9} \quad w_{5,6,7,8} = \frac{1}{36} \tag{2.22}$$

2.4 Accuracy and Stability

The lattice Boltzmann method recovers the incompressible Navier-Stokes equation to the second order in both space and time [5]. Due to the fact that the LB Equation (2.9) is obtained by the linearised expansion of the original kinetic theory based LB equation, the macroscopic results obtained converge to the solution of incompressible Navier-Stokes equation with order Ma^2, where Ma is termed the 'computational ' Mach number. It is given by

$$Ma = \frac{\upsilon_{\max}}{C} \tag{2.23}$$

where $\upsilon_{\max}$ is the maximum simulated velocity in the flow. Therefore, it is required that

$$Ma << 1 \tag{2.24}$$

This indicates that the lattice speed should be sufficiently larger than the maximum fluid velocity to ensure that the solution is reasonably accurate. Also the pressure state from Equation (2.14) reveals that the incompressibility of the fluid is satisfied only approximately by the lattice Boltzmann formulation. It indicates that the LB model can be considered as a 'penalty-based' method that allows a limited degree of compressibility to occur and the fluid speed of sound C_s acts as the penalty value. Thus, a larger value of C will lead to a better approximation of incompressibility condition.

2.4.1 Dependence of Accuracy on Model Parameters

There are three model parameters in Lattice Boltzmann simulation, viz. the relaxation time τ, the lattice spacing h and the time step Δt. These three parameters are to be chosen in such a way that desired solution accuracy can be achieved with reasonable computational cost. The spatial discretisation error is determined by the lattice spacing h. The temporal discretisation error and the total number of time steps of simulation are determined by the time step Δt. In general the error in a LB simulation can be expressed as

$$O(h^2) + O(Ma^2) \tag{2.25}$$

Hence the compressibility error has to be in balance with the resolution error. The relaxation time τ characterizes the time-scale behaviour of fluid particle collisions and determines the lattice fluid viscosity $\bar{\nu}$ in the lattice scale. The viscosity in LB units is given as

$$\bar{\nu} = \frac{1}{3}\left(\tau - \frac{1}{2}\right) \tag{2.26}$$

Thus the true fluid viscosity ν and the lattice viscosity $\bar{\nu}$ are related by,

$$\nu = \bar{\nu}\frac{h^2}{\Delta t} = \bar{\nu}C^2\Delta t \tag{2.27}$$

It can be seen from relations (2.20) and (2.26) that τ should be,

$$\tau > \frac{1}{2} \tag{2.28}$$

τ also influences the stability of the solution. Higher value of τ means that the fluid is more viscous and the LB simulation is more stable.

2.4.2 Methodology of choosing model Parameters

From Equation (2.20) Δt be determined as

$$\Delta t = \left(\tau - \frac{1}{2}\right)\frac{h^2}{3\nu} \tag{2.29}$$

The following methodology may be used to determine the optimum model parameters for a fluid with given viscosity ν.

1) The lattice spacing h is chosen purely from the computational point of view.
2) The value of τ is chosen from stability consideration Equation (2.28).
3) The value of Δt is determined from Relation (2.29) and the corresponding lattice speed is the calculated.
4) The simulation is performed. If a stable solution is achieved, the maximum fluid velocity is obtained and the Mach number Ma is checked. A reasonably accurately solution may have been achieved if Ma is sufficiently small.
5) If the numerical solution obtained is not stable the above procedure is repeated with a larger value of τ.

It can be seen from the Relation (2.29) that for a given viscosity ν and with τ constant, reducing h by half will result in the decrease of Δt. Thus the lattice speed C doubles leading to a much accurate solution. However the computational cost of this method will increase the computational cost. On the other hand, if h is reduced keeping C constant, the value of τ will increase following the Relation (2.20), thereby enhancing the stability of the solution.

2.5 Lattice Units

The LB formulation developed in the preceding sections is introduced in physical coordinates. All the relevant physical variables are in their standard units. It is advantageous to perform numerical simulation in lattice units as non-dimensional variables can be used in the simulation and are easy to handle. Therefore lattice units are used for simulations in the present work also. The particle density distribution functions f_i and the relaxation parameter τ are already non-dimensional and thus remain the same in both the systems. In lattice units, the spacing $\bar{h}$ and time step $\overline{\Delta t}$ are both unity. Thus the lattice speed $\bar{C}$ also becomes equal to unity. The indices of the grid nodes become their lattice coordinates $\bar{\mathbf{x}}$, and time instances $\bar{t}$ become integers. The lattice density is defined as $\bar{\rho} = \rho / \rho_0$, where ρ_0 is the true fluid density. The incompressibility condition is not exactly satisfied in LB simulations. Thus ρ is slightly different than ρ_0. The lattice Boltzmann Equations (2.21) can be converted to lattice units by replacing all the variables by the corresponding lattice variables. Thus the converted LBGK equation in lattice units is written as

$$f_i\left(\bar{\mathbf{x}} + \bar{\mathbf{e}}_i, \bar{t} + 1\right) = f_i\left(\bar{\mathbf{x}}, t\right) - \frac{1}{\tau}\left[f_i\left(\bar{\mathbf{x}}, \bar{t}\right) - f_i^{eq}(\bar{\mathbf{x}}, \bar{t}) \right] \qquad (i = 0,...,8) \qquad (2.30)$$

The equilibrium distribution functions in lattice system f_i^{eq} are given by

$$\begin{aligned} f_0^{eq} &= w_0 \bar{\rho}\left(1 - \frac{3}{2}\bar{\mathbf{v}}.\bar{\mathbf{v}}\right) && (i = 0) \\ f_i^{eq} &= w_i \bar{\rho}\left(1 + 3\bar{\mathbf{e}}_\mathbf{i}.\bar{\mathbf{v}} + \frac{9}{2}\left(\bar{\mathbf{e}}_\mathbf{i}.\bar{\mathbf{v}}\right)^2 - \frac{3}{2}\bar{\mathbf{v}}.\bar{\mathbf{v}}\right) && (i = 1,...,8) \end{aligned} \qquad (2.31)$$

CHAPTER 3

BOUNDARY CONDITIONS

Boundary conditions and initial conditions are essential for any computational fluid dynamic methods. In a LB simulation, the particle distribution functions are primary variables. Thus pressure, velocity and other macroscopic variables are determined by operations on the particle distribution functions. The principal aspect of applying boundary conditions in LB method is to determine a set of particle distribution functions such that desired macroscopic quantities are applied at the boundary nodes. Also it would be advantageous if the boundary implementation procedure maintains the simplicity and locality of the LB solution process.

3.1 Periodic Boundary Conditions

Periodic boundary conditions are the simplest type of boundary conditions and can be easily implemented. In this boundary condition, the output of a fluid domain is used as its input. Thus, an infinite array of analysed fluid domain is created in the direction of the boundary normal. The schematic diagram of the periodic boundary condition is shown in Figure (3.1).

3.1.1 Periodic Boundary Conditions at Left and Right Boundaries

At the left boundary the inward facing particle distributions f_1, f_5 and f_8 are unknown after the streaming operation. These values are obtained from the outward facing distributions at the right boundary. Thus,

$$\begin{aligned} f_1^{left} &= f_1^{right} \\ f_5^{left} &= f_5^{right} \\ f_8^{left} &= f_8^{right} \end{aligned} \tag{3.1}$$

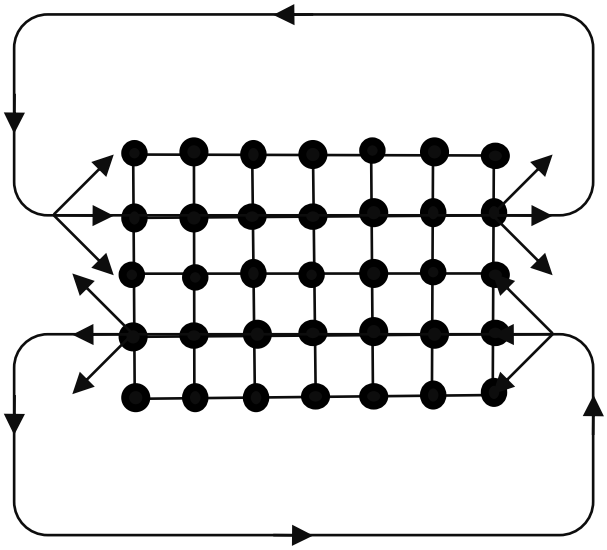


Figure 3.1: *Periodic Boundary Conditions*

At the right boundary the inward facing particle distribution functions f_3, f_6 and f_7 are unknown after streaming. These values are obtained from the outward facing populations at the left boundary. Thus,

$$\begin{aligned} f_3^{right} &= f_3^{left} \\ f_6^{right} &= f_6^{left} \\ f_7^{right} &= f_7^{left} \end{aligned} \tag{3.2}$$

3.1.2 Periodic Boundary Conditions at Corner Nodes

The treatment of periodic boundary condition at the corner nodes is performed in a slightly different way compared to other nodes at the boundary. At the top left corner the populations f_1, f_4 and f_8 are unknown after streaming. They are determined using the flowing rule.

$$\begin{aligned} f_1^{top-left} &= f_1^{top-right} \\ f_4^{top-left} &= f_4^{bot-left} \\ f_8^{top-left} &= f_8^{bot-right} \end{aligned} \tag{3.3}$$

At the bottom left corner the populations f_1, f_2 and f_5 are unknown after streaming. They are determined using the following rule.

$$
\begin{aligned}
f_1^{bot-left} &= f_1^{bot-right} \\
f_2^{bot-left} &= f_2^{top-left} \\
f_5^{bot-left} &= f_5^{top-right}
\end{aligned}
\tag{3.4}
$$

Similarly the unknown populations at the top-right corner can be determined using the following set of rules.

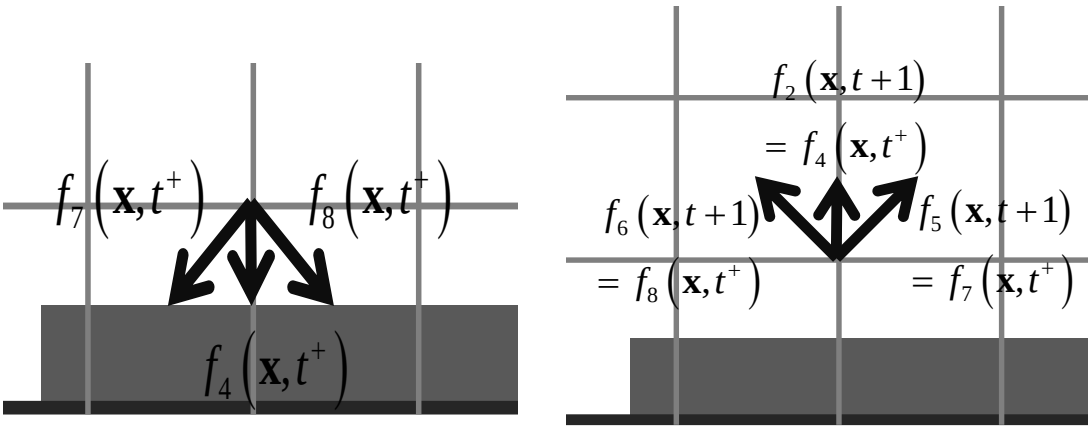
$$
\begin{aligned}
f_3^{top-right} &= f_3^{top-left} \\
f_4^{top-right} &= f_4^{bot-right} \\
f_7^{top-right} &= f_7^{bot-left}
\end{aligned}
\tag{3.5}
$$

The unknown populations at the bottom-right corner can be determined as follows.

$$
\begin{aligned}
f_2^{bot-right} &= f_2^{top-right} \\
f_3^{bot-right} &= f_3^{bot-left} \\
f_6^{bot-right} &= f_6^{top-left}
\end{aligned}
\tag{3.6}
$$

3.2 No-slip Boundary Conditions

The wall boundary condition or the no-slip boundary condition occurs at the interface of the fluid and the stationary solid wall. The no-slip boundary condition can be implemented by the bounce back method which reflects the particle distribution functions at boundary nodes in the direction of incidence. The main advantage of the bounce back method is that it is very easy to implement. Also the locality of the LB solution process is maintained. However the bounce-back boundary condition is only first order accurate in most of the situations. The no-slip condition can be enforced over one or two steps as described below.



(a) (b)

Figure 3.2: *Single* step bounce back condition at time *t* (left) and time *t+1* (right)

3.2.1 Single Step Bounce Back

The nodes lying within the wall region are called solid nodes and the nodes in the fluid region are fluid nodes. The single step bounce back method is applied at the fluid nodes just before the solid nodes. The post collision distribution functions ready to be streamed to the solid nodes are redirected in the opposite direction instead being streamed. The single step bounce back procedure is shown in Figure (3.2).

3.2.2 Two Step Bounce Back

The two step bounce back boundary condition is implemented in two time steps and at the solid nodes placed just adjacent to the boundary. As indicated in Figure (3.3) the particle distribution function at fluid nodes are relaxed at time t resulting in three particle distributions pointing towards the respective solid nodes. They are propagated to the solid nodes during the streaming process. The bounce-back condition is then enforced on all the solid nodes by modifying the relaxation process by of reversing the direction of each of the distributions at these nodes. Then the standard streaming process is performed so that the functions pointing in the direction of the fluid are streamed back to the fluid node. Thus at time $t+2$ the reflected particle distribution functions are at the fluid nodes where they started ready to be relaxed. The advantage of the two step approach is that it is not required

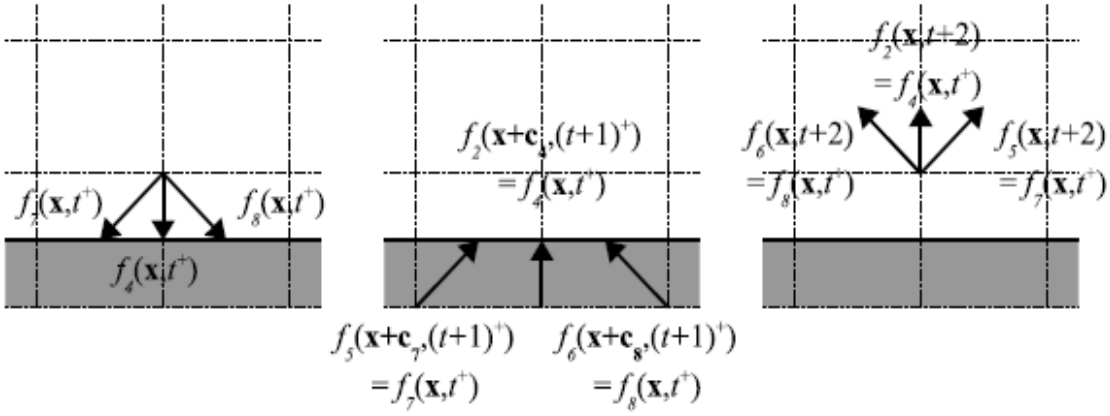


Figure 3.3: Two step bounce-back boundary condition at time *t* (left), time *t+1* (middle) and time *t+2* (right)

to define the fluid boundary nodes. Also if the physical boundary is assumed to lie halfway between the fluid node and the solid node the two step bounce back condition is found to be second order accurate.

3.3 Pressure and Velocity Boundaries

The application of pressure and velocity boundary conditions involves the inverse problem of determining the particle distribution functions from the prescribed macroscopic variables. The pressure and velocity boundary conditions discussed in the below sections was proposed by Zou and He [15]. Like the bounce-back boundary conditions, the operations in this boundary technique are also local preserving the parallel nature of the solution process.

3.3.1 Velocity Wall Boundary Condition

Consider a 2D flow boundary with prescribed velocity u_x and u_y. At the bottom node the boundary is aligned with the x-direction. The particle distribution functions f_4, f_7 and f_8 are pointing into the wall. The functions $f_0, f_1, f_3, f_4, f_7, f_8$ are known post the streaming operation. The distribution functions f_2, f_5, f_6 and density ρ are to be determined.

From the mass conservation equation (2.12), it can be written as,

$$f_2 + f_5 + f_6 = \rho - (f_0 + f_1 + f_3 + f_4 + f_7 + f_8) \tag{3.7}$$

From the momentum conservation equation (2.13), it can be written as,

$$f_5 - f_6 = \rho u_x - (f_1 - f_3 - f_7 + f_8) \tag{3.8}$$

$$f_2 + f_5 + f_6 = \rho u_y + (f_4 + f_7 + f_8) \tag{3.9}$$

Thus from equation (3.7) and (3.9) the density can be derived as

$$\rho = \frac{1}{1-u_y}\left[f_0 + f_1 + f_3 + 2(f_4 + f_7 + f_8)\right] \tag{3.10}$$

Assuming that the bounce-back rule is still valid for the non-equilibrium part of the particle distribution functions normal to the boundary the unknown populations can be determined as

$$\begin{aligned} f_2 &= f_4 + \frac{2}{3}\rho u_y, \\ f_5 &= f_7 - \frac{1}{2}\left(f_1 - f_3\right) + \frac{1}{2}\rho u_x + \frac{1}{6}\rho u_y, \\ f_6 &= f_8 + \frac{1}{2}\left(f_1 - f_3\right) - \frac{1}{2}\rho u_x + \frac{1}{6}\rho u_y \end{aligned} \tag{3.11}$$

3.3.2 Specification of Pressure on a Flow Boundary

In the lattice Boltzmann method the pressure is related to the density by the isothermal Equation (2.14) of state. Thus a specification of pressure difference means a specification of density difference. Consider a 2D domain as shown in Figure (3.4) with a pressure $p_{in} = c^2 \rho_{in}$ and velocity u_y applied to the left boundary. From the mass conservation Equation (2.12) it can be written as,

$$f_1 + f_5 + f_8 = \rho_{in} - \left(f_0 + f_2 + f_3 + f_4 + f_6 + f_7 \right) \tag{3.12}$$

From the momentum conservation equation it can be obtained as

$$\begin{aligned} f_1 + f_5 + f_8 &= \rho_{in} u_x + \left(f_3 + f_6 + f_7 \right) \\ f_5 - f_8 &= f_2 - f_4 + f_6 - f_7 \end{aligned} \tag{3.13}$$

From Equations (3.12) and (3.13), u_x can be derived as,

$$u_x = 1 - \frac{\left[f_0 + f_2 + f_4 + 2\left(f_3 + f_6 + f_7 \right) \right]}{\rho_{in}} \tag{3.14}$$

Assuming that the bounce-back rule is valid for the non-equilibrium particle distributions normal to the inlet boundary $\left(f_1 - f_1^{eq} = f_3 - f_3^{eq} \right)$, the unknown distributions can be determined by,

$$\begin{aligned} f_1 &= f_3 + \frac{2}{3}\rho_{in} u_x \\ f_5 &= f_7 - \frac{1}{2}\left(f_2 - f_4 \right) + \frac{1}{6}\rho_{in} u_x, \\ f_8 &= f_6 + \frac{1}{2}\left(f_2 - f_4 \right) + \frac{1}{6}\rho_{in} u_x \end{aligned} \tag{3.15}$$

The corner node at the inlet requires a special treatment. Consider the bottom node as example. The distribution functions f_3, f_4, f_7 are known post streaming. The density ρ is specified and the velocity u_x and u_y are equal to zero. The distribution functions f_1, f_2, f_5, f_6, f_8 are to be determined. Assuming that the bounce-back rule is valid for the non-equilibrium part of the particle distribution normal to the inlet and the boundary, we have

$$\begin{aligned} f_1 &= f_3 + \left(f_1^{(eq)} - f_3^{(eq)} \right) = f_3, \\ f_2 &= f_4 + \left(f_1^{(eq)} - f_3^{(eq)} \right) = f_4 \end{aligned} \tag{3.16}$$

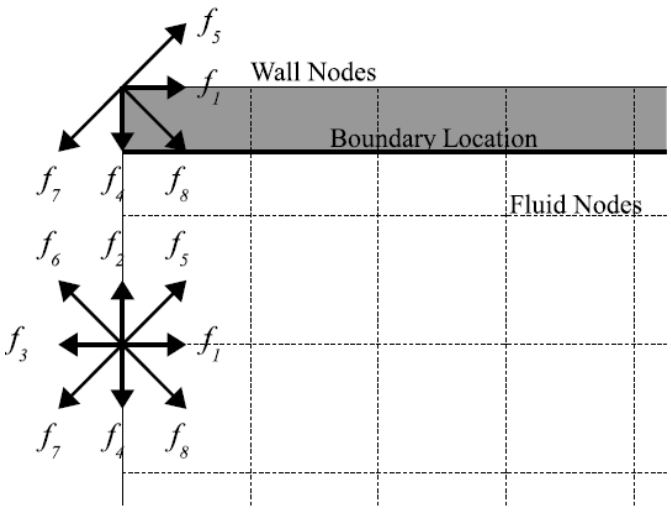


Figure 3.4: Specification of Pressure at flow boundary

From Equation (3.15) and the mass conservation Equation (2.12), the unknown distribution functions can be determined as,

$$\begin{aligned} f_5 &= f_7, \\ f_6 &= f_8 = \frac{1}{2}\left[\rho_{in} - \left(f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_7\right)\right] \end{aligned} \tag{3.17}$$

Similar procedure can be applied to the top inlet node, the outlet corner nodes and in situations where the vertical velocity u_y is non-zero.

3.3.3 Specification of Velocity on a Flow Boundary

Velocity boundary conditions are often defined in a fluid flow. Consider a 2D domain with both u_x and u_y specified at the inlet as shown in Figure (3.5). After streaming the particle distribution functions f_2, f_4, f_3, f_6, f_7 are known at the inlet. The density ρ and the functions f_1, f_5, f_8 are to be determined. Following a procedure similar to the pressure boundary conditions, the solution can be obtained as follows:

$$\begin{aligned} \rho_{in} &= \frac{1}{1-u_x}\left[f_0 + f_2 + f_4 + 2\left(f_3 + f_6 + f_7\right)\right] \\ f_1 &= f_3 + \frac{2}{3}\rho_{in}u_x \\ f_5 &= f_7 - \frac{1}{2}\left(f_2 - f_4\right) + \frac{1}{6}\rho_{in}u_x + \frac{1}{2}\rho_{in}u_y \\ f_8 &= f_6 + \frac{1}{2}\left(f_2 - f_4\right) + \frac{1}{6}\rho_{in}u_x - \frac{1}{2}\rho_{in}u_y \end{aligned} \tag{3.18}$$

Similarly the density and the distribution functions at the outlet boundary can be determined as,

$$\begin{aligned} \rho_{out} &= \frac{1}{1+u_x}\left[f_0 + f_2 + f_4 + 2\left(f_1 + f_5 + f_8\right)\right] \\ f_3 &= f_1 - \frac{2}{3}\rho_{out}u_x \\ f_6 &= f_8 - \frac{1}{2}\left(f_2 - f_4\right) - \frac{1}{6}\rho_{out}u_x + \frac{1}{2}\rho_{out}u_y \\ f_7 &= f_5 + \frac{1}{2}\left(f_2 - f_4\right) - \frac{1}{6}\rho_{out}u_x - \frac{1}{2}\rho_{out}u_y \end{aligned} \tag{3.19}$$

3.4 Inclusion of External Body Force

There are a wide range of fluid flow problems in which gravity and buoyancy effects are significant [38]. The incorporation of the effects of gravity and other external body forces in a lattice Boltzmann model was introduced by Buick *et al* [38]. They suggested that the body force can be incorporated by adding an additional term in the collision operator which modifies the distribution function given by.

$$\Omega_i(\mathbf{x},t) = -\frac{1}{\tau}\left[f_i(\mathbf{x},t) - f_i^{e}(\mathbf{x},t)\right] + \frac{D}{bc^2} F_i c_i \tag{3.20}$$

where Ω_i is the collision operator defined by Equation (2.19), D is the number of dimensions and b is the number of lattice directions and F is the body force acting on the fluid. The addition of a term in the collision operator results in change in momentum and this change is incorporated by modifying the momentum as follows.

$$\rho v_i = \rho u_i + \frac{1}{2} F_i \tag{3.21}$$

CHAPTER 4

SINGLE PHASE LBM NUMERICAL RESULTS

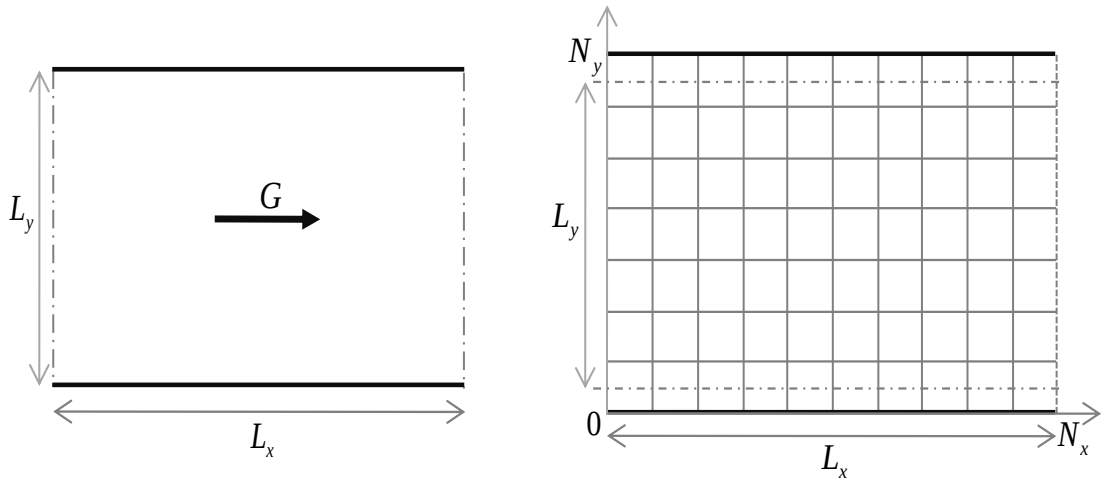
The lattice Boltzmann method and the various boundary conditions discussed in the preceding chapters were implemented and simulation of a number of flow validation problems was performed. The aim of these simulations was to determine the accuracy and the robustness of the code and also to investigate issues like stability, compressibility errors, etc. The D2Q9 LBGK model was used for the two dimensional simulations. The two step bounce back method was used for implementing the wall boundaries. This method was chosen over the single step bounce back method because the two step method is second order accurate if the wall is assumed to be midway between the wall and the fluid node. Also the bounce back method is extremely convenient in modelling irregular boundaries. The Zou and He [15] technique is used for defining the pressure and velocities at the boundaries. The body forces were included using a post-collision operation which alters the momenta of the particle distribution functions as discussed in Section (3.4).

4.1 2D Poiseuille Flow

Poiseuille flow is an incompressible flow between two stationary parallel plates. It is generally used as a benchmark for numerical analysis as the analytical solution can be derived from Navier-Stokes equation. For a flow driven by constant body force, the velocity profile across the width of the channel is given by

$$u_x(y) = \frac{-G}{2\nu}\left[\left(\frac{a}{2}\right)^2 - |y|^2\right] \tag{4.1}$$

where G is the body force and a is the channel height.



(a) (b)

Figure 4.1: *Poiseuille flow driven by body force G. (a) Domain setup (b) LB Grid*

4.1.1 Problem Setup

The implemented LBM code was applied to a plane Poiseuille flow driven by a body force. The schematic diagram of a rectangular domain of size $L_x \times L_y = 100 \times 24$ LU's is shown in the Figure (4.1a). The applied body force along the x-direction is given by $G = 0.00011$. Periodic boundaries were employed at the inlet and outlet boundaries in order to maintain a plane flow.

No-slip boundary condition was applied to the walls at the top and bottom using the two-step bounce back method. Since the two step bounce back method assumes the actual boundary to be present half way between the fluid and solid node, the effective height of the domain is $L_y - 1 = 23$ LU's. The initial condition of $u_x = 0, u_y = 0$ and an average density of $\rho_0 = 1$ was applied in the interior of the domain.

The criteria used to determine the steady state is given by

$$\sum_i \frac{\left|u(\mathbf{x_i}, t+1) - u(\mathbf{x_i}, t)\right|}{\left|u(\mathbf{x_i}, t+1)\right|} \leq 10^{-6} \tag{4.2}$$

where the summation is over all the lattice nodes in the domain. The steady state solution was obtained only after a few thousand iterations depending on the value of viscosity and the boundary conditions. The simulations were performed using

different values of τ (0.75, 0.9, 1.0, 1.25 and 1.5) to study the effect of the relaxation parameter on the velocity profile obtained.

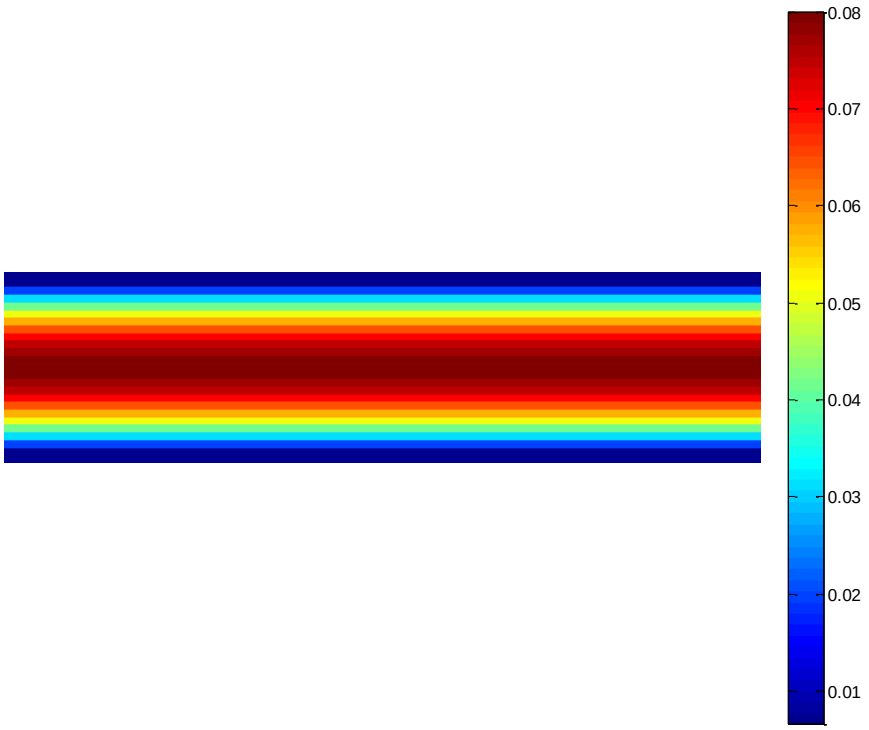
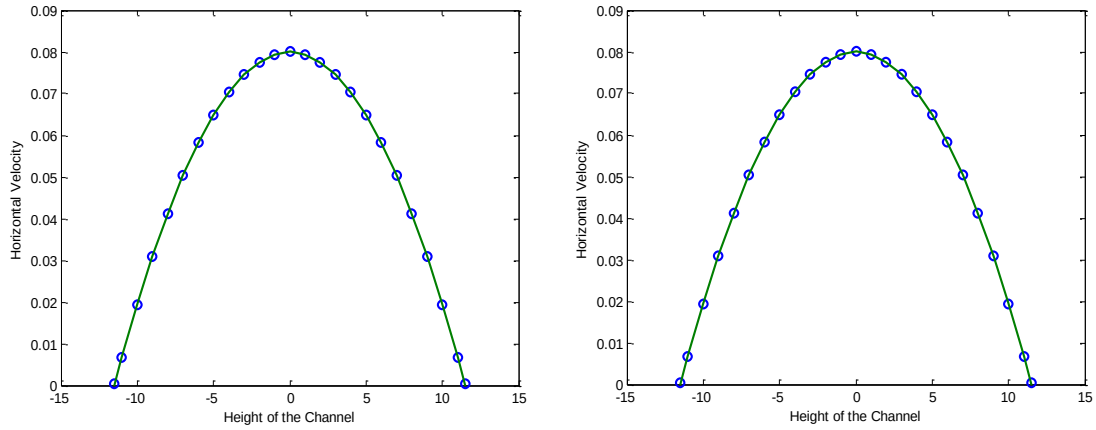


Figure 4.2: *Contour plot of the velocity profile for* $\tau = 1.0$

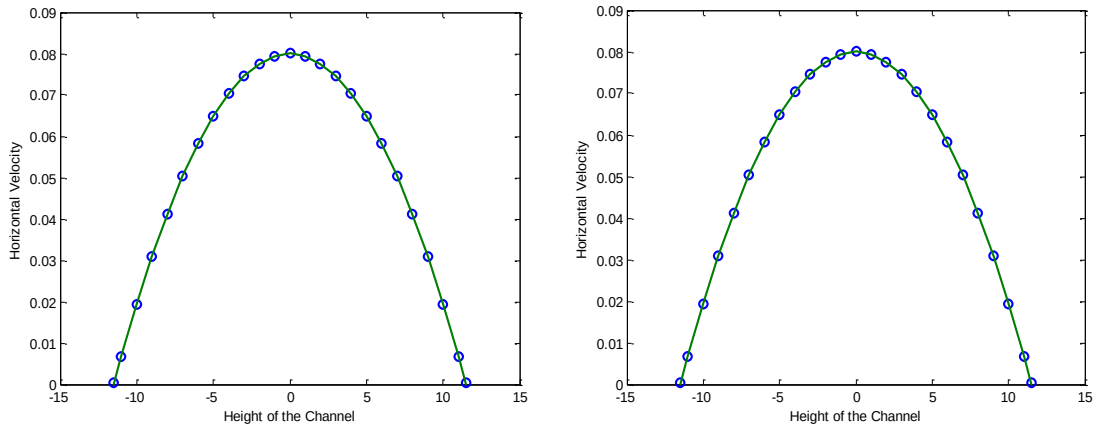
4.1.2 Results and Discussion

The velocity profile was measured at several cross sections. The magnitude of the vertical component of velocity u_y was always found to be smaller than 10^{-6} irrespective of τ .The steady state velocity contour plot obtained from the simulation for $\tau = 1.0$ is shown in Figure 4.2. Also a plot of comparison of the horizontal velocity profile with analytical velocity profile at various sections along the domain for $\tau = 1.0$ is show in Figure (4.3) and Figure (4.4). It can be seen that the velocity profile obtained from the LB simulation matches well with the analytical solution at all the sections in the channel. The velocity profile obtained for other values of τ is not shown since they were similar to plots shown in Figure (4.3) and Figure (4.4). In order to determine the effect of relaxation parameter on the LB simulation τ was varied from 0.55 to 3. It can be seen from Figure (4.5) that the maximum velocity relative error is close to zero only for values of τ ranging from 0.7 to 1. Hence for further LB simulations in this work, the value of τ is taken between 0.8 and 1.0.



(a) (b)

Figure 4.3: *Horizontal velocity comparison. (a) At x=0. (b) At x=25[− indicates the analytical velocity and ⊙ indicates the actual velocity]*



(a) (b)

Figure 4.4: *Horizontal velocity comparison. (a) At x=50. (b) At x=100[− indicates the analytical velocity and ⊙ indicates the actual velocity]*

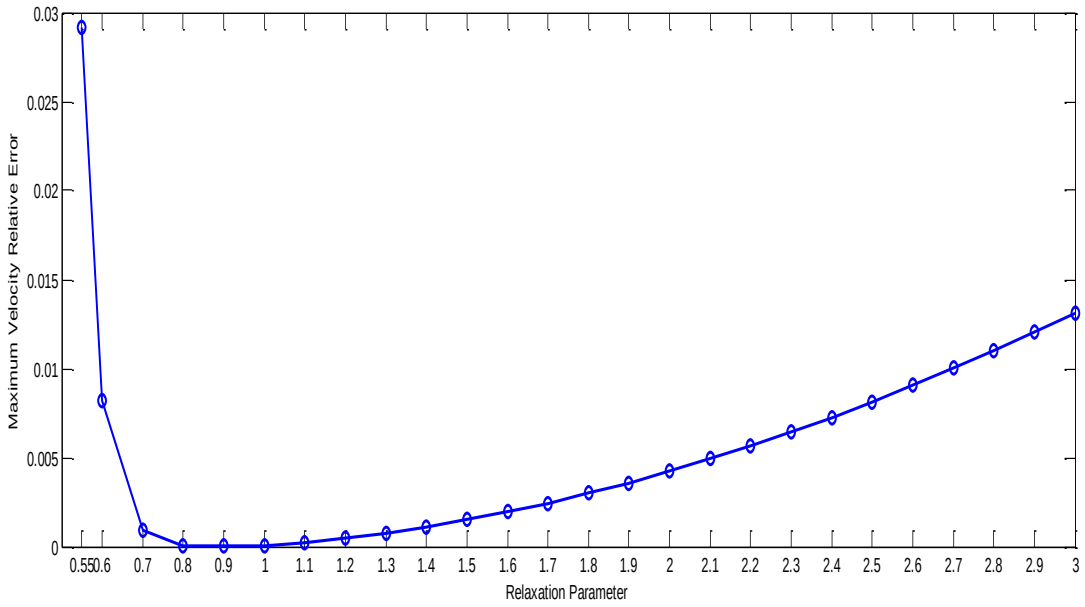


Figure 4.5: *Plot of the maximum velocity relative error v/s relaxation parameter* τ

4.2 2D Lid driven cavity flow

Lid driven cavity flow is a standard benchmark problem in the field of CFD. Most of the conventional numerical solutions for this problem are based on vorticity and stream function formulation and discretisation of the Navier-Stokes equation using finite difference, multi-grid or the finite element methods. A thorough analysis of the problem was done by Ghia et al [28]. The results obtained from the LB simulation of the problem will be compared with work of Ghia et al [28].

4.2.1 Problem setup

A square cavity of dimensions $L_x \times L_y = 129 \times 129$ LU's is shown in Figure (4.1). The cavity is filled with a fluid of uniform density 2.7LU's. The horizontal and vertical velocity at bottom horizontal wall and the two vertical walls are set to zero. This is done by the two step bounce back method described in Section (3.2.2). A uniform horizontal uniform velocity of $u_x = 1$ and a vertical velocity of $u_y = 0$ is applied to the horizontal wall at the top as shown in the Figure (4.1) using the velocity wall boundary condition described in Section (3.3.1).

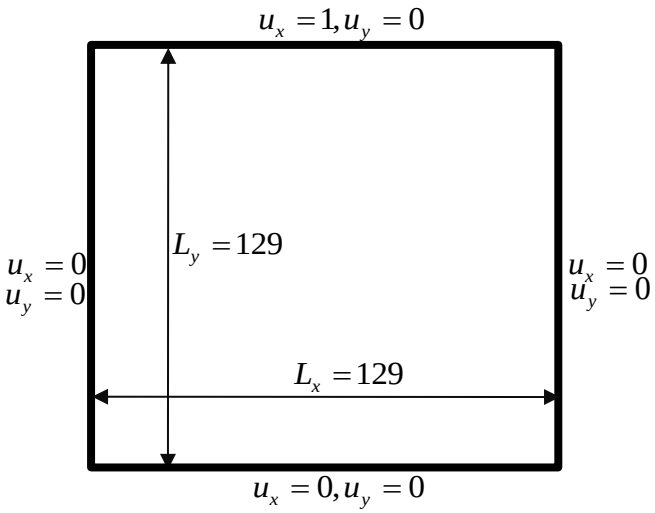


Figure 4.6: *Lid driven cavity flow problem setup*

4.2.2 Results and discussion

The simulations are performed for a Reynolds number of $\mathrm{Re} = 100$ and $\mathrm{Re} = 400$. The velocity contours obtained are shown in Figure (4.7) and Figure (4.8). It can be seen that as the Reynolds number is increased, the vortex formed moves towards the geometric center of the cavity

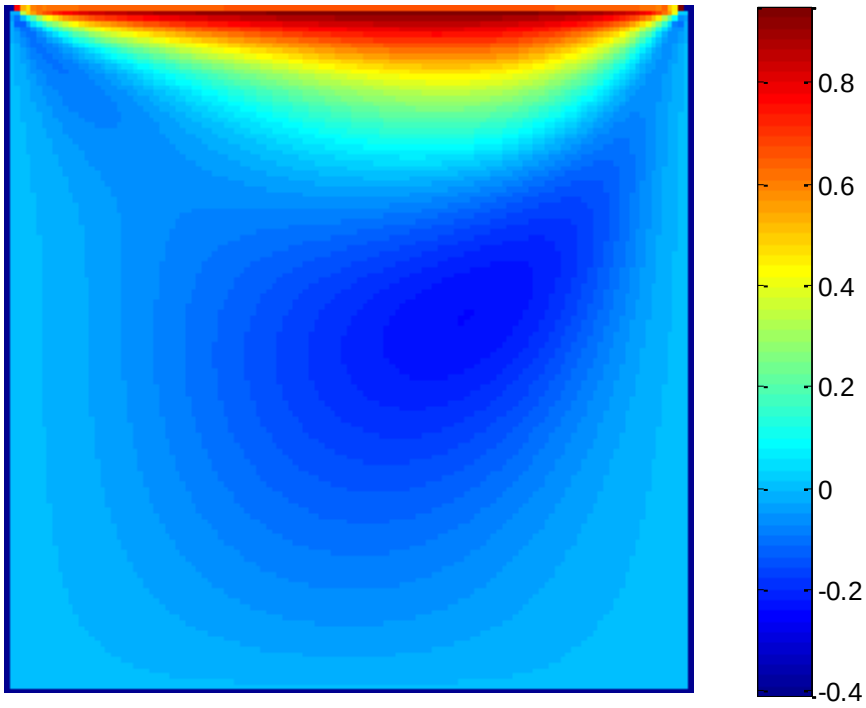


Figure 4.7: *Velocity contour for lid driven cavity flow at* $\mathrm{Re} = 100$

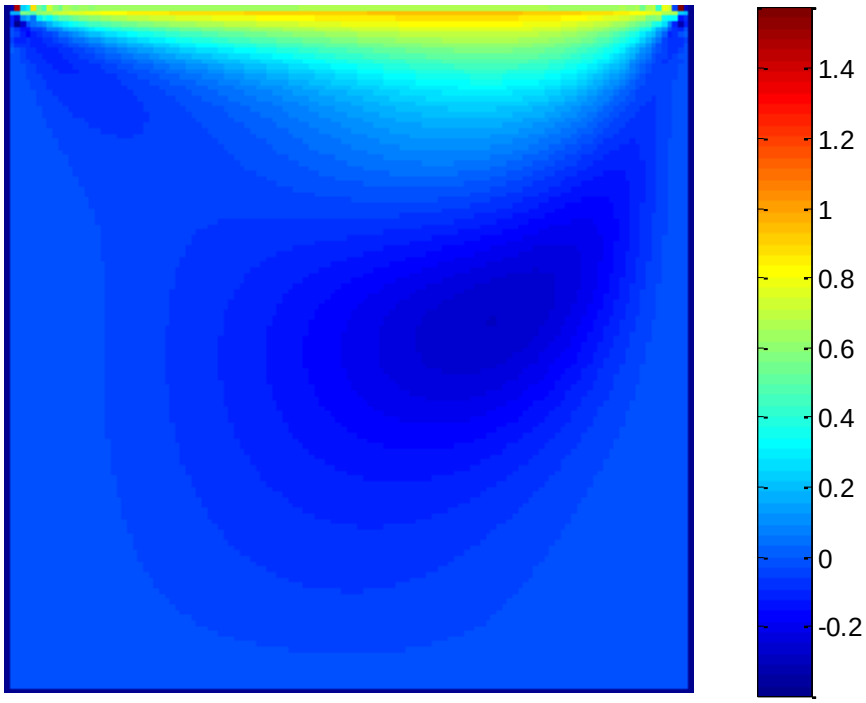
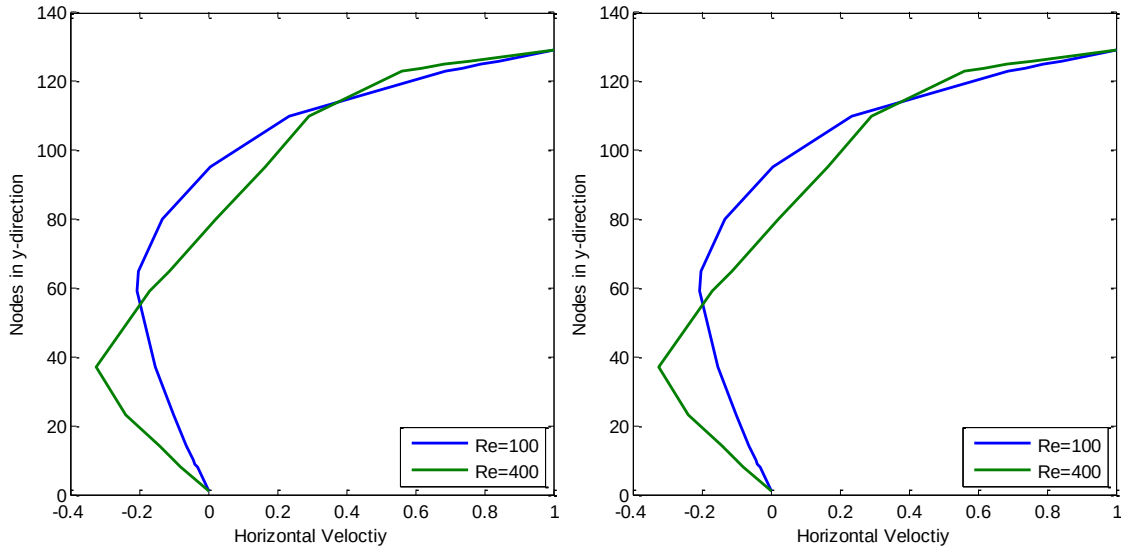


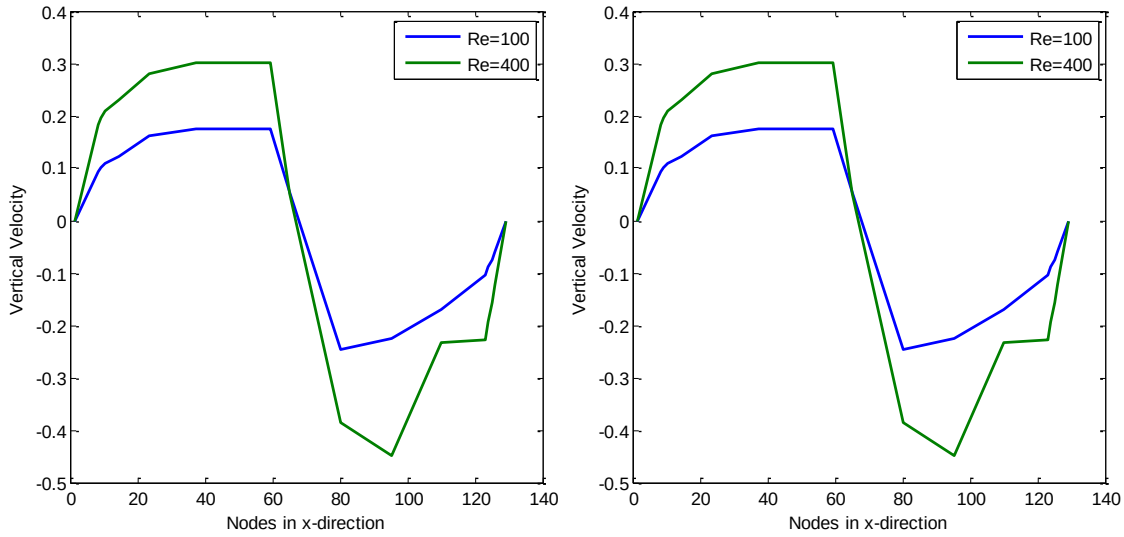
Figure 4.8: *Velocity contour for lid driven cavity flow at* $\mathrm{Re} = 400$

The velocity u_x in horizontal direction was observed at various nodes on the vertical line through the geometric center. The Figure (4.9) shows comparison plot of these



(a) Present Work (b) Ghia et al

Figure 4.9: *Plot of horizontal velocity along vertical line through geometric centre. (a) Present work. (b) Ghia et al*



(a) Present work (b) Ghia et al

Figure 4.10: *Plot of vertical velocity along horizontal line through geometric centre. (a) Present work. (b) Ghia et al*

velocities obtained in the present work and those found by Ghia et al [28]. It can be seen that the results obtained in the present work closely match the values obtained by Ghia et al [28]. Similarly the velocities in the vertical direction were observed at a horizontal line through the geometric centre and plotted as shown in Figure (4.10). The obtained results in the present work agree with results obtained by Ghia et al [28].

CHAPTER 5

LATTICE BOLTZMANN MODEL FOR SIMULATING IMMISCIBLE TWO-PHASE FLOWS

In problems involving immiscible multiphase flows, the equations of motion that hold in each fluid are solved with appropriate conditions defined at the interface between the fluids. The interface is a free surface that evolves in time and the conditions that hold at the interface involve the physical properties of the fluids such as surface tension. An alternative description of immiscible two-phase flows is based on diffuse-interface models in which quantities such as surface tension, for example, are distributed throughout an interfacial region. In such a description, surface tension is represented as a distributed stress within this region.

The numerical simulation of multiphase flow problems can be performed using two different approaches; the Lagrangian approach and the Eulerian approach. In both these approaches it is required to track the interface(s) between the two phases as they evolve in time. The Lagrangian approach can accurately track an interface as the fluid flow is analysed by observing the trajectories of specific fluid particles. The dynamics of the interface can be captured from the evolution of the boundary. However, if the interface topology is completely changed the Lagrangian method can suffer from ill-conditioning and singularities. Also, three dimensional computations can prove costly due to the necessity to mesh complex geometries. In the Eulerian approach the analysis is performed by observing the field variables like velocity, pressure, density, etc. Instead of tracking the interface explicitly, the Eulerian method reconstructs it as an isocontour of a field variable. Thus the Eulerian overcomes the difficulties in the Lagrangian approach since large deformations in an interface can be captured without a re-discretisation of the domain. However, the

problem with this method is that interfacial diffusion effects are generally smeared over a region surrounding the interface due to the lack of explicit treatment.

The micro/mesoscopic consideration of a two-phase flow indicates that the segregation of two fluids is due to inter-particle forces at the kinetic level. The lattice Boltzmann model, as described in Chapter 2, constructs a simplified kinetic model which exhibits behaviour identical to classical hydrodynamic equations at the macroscopic level by incorporating the mesoscopic processes. Also the LBM can be implemented efficiently on parallel computers. Thus the LBM is in a strong position compared to other continuum based approaches since the particle interactions can be incorporated into the evolution of the distribution function. As a result, a multiphase lattice Boltzmann model facilitates the interfaces to emerge spontaneously from the underlying dynamics rather than tracking them.

Many lattice gas and lattice Boltzmann models have been developed to predict the flow of two interacting fluids, each having its own set of advantages and limitations.

The seven-velocity Shan-Chen model proposed by Shan and Chen [18] is widely used for multiphase flows due to its simplicity and elegance. Since the flows with more than one phase have a non-ideal equation of state, Shan and Chen [18] looked to preserve this feature in a lattice Boltzmann framework by incorporating non-local interactions among particles by defining a concept of interaction potential. In the Shan and Chen [18] model the surface tension is given by strength of the microscopic interaction and the shape of the density profile at the liquid-gas interface. Although it is possible to adjust these two independently, keeping the density profile constant and changing the surface tension requires the adjustment of two parameters. Furthermore the absolute value of surface tension is not known prior to evaluation of the density profile on the interface. A simulation of flows of immiscible fluids with different viscosities was done by Chin *et al* [19] by using the Shan-Chen model on a D2Q9 lattice with LBGK collision operator. The evaluation of Laplace's law for surface tension showed noticeable errors and two-phase Poiseuille flow gave noticeable discrepancies near the interface region.

Alternative LB model for simulating immiscible fluid flow was proposed by Swift *et al.* [20] by introducing the phase effects directly in collision using a free-energy approach. Their model is constructed such that the pressure tensor is consistent with the tensor derived from the free-energy function of non-uniform

fluids. This model leaves the interface width relatively wide as pointed out by Latva-Kokko [22]. Theoretical criticisms of the free-energy approach are made by Luo [21].

Another approach for simulating the two phase flow for immiscible fluids is a method based on colour gradients. The colour gradient method was first proposed by Gunstensen *et al* [23]. This method combined the single phase LBM of McNamara and Zanetti [24] with the multiphase lattice gas algorithm of Keller and Rothman [25]. In this method a perturbation is added to the linearised collision operator which makes the pressure tensor locally anisotropic near the interface. The result of this addition is the surface tension at the interfaces while retaining the adherence to the Navier-Stokes equations in the homogeneous regions. The perturbation addition also leads to mass being depleted along lattice links parallel to an interface and mass being added to lattice link perpendicular to an interface. This model is utilises a fully linearised collision operator which is computationally inefficient in 3D and is restricted to flows in which the fluids have same densities and viscosities. Also the governing equations are not exactly solved for two-phase flow. The key advantage of LB models which followed from the multiphase lattice gas algorithm of Keller and Rothman [25] is that the density ratio, the viscosity ratio and surface tension can be chosen independently. A number of other models extending the Rothman-Keller model have been proposed. The models proposed by Gunstensen *et al* [23] and Tölke *et al* [26] allowed for different densities and viscosities by incorporating the freedom of rest particle equilibrium distribution function and space dependent relaxation process.

This work uses the model proposed by Reis and Phillips [27] for simulating immiscible two phase flows on a D2Q9 lattice. The model proposed by Reis and Phillips [27] was similar to the model of Gunstensen *et al* [23] but with several important modifications.

5.1 Immiscible lattice Boltzmann model

Consider a 2D incompressible two phase flow domain. Let ‘ R ’ and ‘ B ’ denote the two phases in the domain and let their densities be ρ^R and ρ^B respectively. A D2Q9 lattice Boltzmann model is constructed on the domain as described in Section

2.3.2.1. Let $f_i^{\sigma}(\mathbf{x},t)$ be the density distribution function at position $\mathbf{x}$ and time instant t. The superscript $\sigma = R$ or B indicates the species of the fluid. Thus the total at node $\mathbf{x}$ at any instant of time t is given by

$$f_i(\mathbf{x},t) = f_i^R(\mathbf{x},t) + f_i^B(\mathbf{x},t) \tag{5.1}$$

As in the case of single phase flow, the LBM solution progresses via a two stage process of collision and streaming at each time step. However in the case of two-phase flow the operations of collision and streaming are performed separately on the two phases. Thus the evolution of each phase is given by

$$f_i^{\sigma}(\mathbf{x}+\mathbf{c_i},t+1) = f_i^{\sigma}(\mathbf{x},t) + \Omega_i^{\sigma}(\mathbf{x},t) \tag{5.2}$$

The collision operator $\Omega_i^{\sigma}(\mathbf{x},t)$ consists of three parts given by

$$\Omega_i^{\sigma}(\mathbf{x},t) = \Omega 3_i^{\sigma}\left\{\Omega 1_i^{\sigma} + \Omega 2_i^{\sigma}\right\} \tag{5.3}$$

The first operation $\Omega 1_i^{\sigma}$ is similar to the collision operator incorporating the relaxation to the local equilibrium state described in Sections 2.4 and 2.7. Using the LBGK operator $\Omega 1_i^{\sigma}$ is given by,

$$\Omega 1_i^{\sigma} = -\frac{1}{\tau^{\sigma}}\left(f_i^{\sigma} - f_i^{\sigma(e)}\right) \tag{5.4}$$

where τ^{σ} is the relaxation time, $f_i^{\sigma(e)}$ is the equilibrium distribution function corresponding to the phase σ. Taking $\omega^{\sigma} = 1/\tau^{\sigma}$ as the relaxation parameter, Equation 5.4 can be written as

$$\Omega 1_i^{\sigma} = -\omega^{\sigma}\left(f_i^{\sigma} - f_i^{\sigma(e)}\right) \tag{5.5}$$

The equilibrium functions are chosen as follows:

$$f_0^{\sigma(e)} = \rho^{\sigma}\left(\alpha^{\sigma} - \frac{2}{3}\mathbf{u}^2\right) \qquad (i=0) \tag{5.6}$$

$$f_i^{\sigma(e)} = \rho^{\sigma}\left(\frac{1-\alpha^{\sigma}}{5} + W_i\left[3\mathbf{c}_i\cdot\mathbf{u} + \frac{9}{2}(\mathbf{c}_i\cdot\mathbf{u})^2 - \frac{3}{2}\mathbf{u}^2\right]\right), \quad (i=1,2,3,4) \tag{5.7}$$

$$f_i^{\sigma(e)} = \rho^{\sigma}\left(\frac{1-\alpha^{\sigma}}{20} + W_i\left[3\mathbf{c}_i\cdot\mathbf{u} + \frac{9}{2}(\mathbf{c}_i\cdot\mathbf{u})^2 - \frac{3}{2}\mathbf{u}^2\right]\right), \quad (i=5,6,7,8) \tag{5.8}$$

where α^{σ} is a free parameter which determines the speed of sound, $\left(c_s^{\sigma}\right)^2$. Thus α^{σ} controls the hydrodynamic pressure at the interfaces. It can be viewed as representing the ensemble average number of degenerate rest particles, which are required to maintain a stable interface and achieve a density variation between the fluids. Thus, the choice of the parameter α^{σ} is very important for multiphase flows with large density difference. In order to ensure that $0 < f_i^{\sigma} < 1$, it is required that the value of α^{σ} is chosen between 0 and 1.

The weights W_i used in Equations 5.6, 5.7, 5.8 are given by

$$W_i = \begin{cases} \frac{4}{9}, & (i = 0) \\ \frac{1}{9}, & (i = 1,2,3,4) \\ \frac{1}{36}, & (i = 5,6,7,8) \end{cases} \tag{5.9}$$

Mass is defined as the first moment of the density distribution function respectively and is given by

$$\rho^{\sigma} = \sum_i f_i^{\sigma} = \sum_i f_i^{\sigma(e)} \tag{5.10}$$

where ρ^{σ} is the density of the fluid σ, $\rho = \rho^R + \rho^B$ is the total density. The momentum is defined as the second moment of the density distribution function respectively and is given by

$$\rho \mathbf{u} = \sum_i \sum_{\sigma} f_i^{\sigma} \mathbf{c_i} = \sum_i \sum_{\sigma} f_i^{\sigma(e)} \mathbf{c_i} \tag{5.11}$$

where $\mathbf{u}$ is the local fluid velocity. The pressure is given by

$$p_0^{\sigma} = \frac{3\rho^{\sigma}\left(1-\alpha^{\sigma}\right)}{5} = \rho^{\sigma}\left(c_s^{\sigma}\right)^2 \tag{5.12}$$

Also the density ratio γ is defined as

$$\gamma = \frac{\rho^R}{\rho^B} = \frac{1-\alpha^B}{1-\alpha^R} \tag{5.13}$$

5.1.1 Two-phase collision operator

The two-phase collision operator, $\Omega 2_i^{\sigma}$, is responsible for generation of surface tension at the interface while satisfying the mass and momentum conservation given by Equation 5.10 and Equation 5.11 respectively. In the two-phase collision operation an anisotropic perturbation is added to the particle distribution functions near the interface. The position of interface is located by examining the magnitude of the local colour gradient. If the spatial colour difference, $\bar{\rho}$ is defined as

$$\bar{\rho}(\mathbf{x}) = \rho^R(\mathbf{x}) - \rho^B(\mathbf{x}), \tag{5.14}$$

the colour gradient can be calculated in terms of the colour difference as follows:

$$\mathbf{H}(\mathbf{x}) = \nabla\bar{\rho}(\mathbf{x}) \tag{5.15}$$

A fourth-order approximation of the colour gradient is given by

$$\mathbf{F}(\mathbf{x}) = \sum_{i=1}^{8} \mathbf{c}_i \left[\rho^R(\mathbf{x}+\mathbf{c}_i) - \rho^B(\mathbf{x}+\mathbf{c}_i) \right] \tag{5.16}$$

It should be noted that the value of $\mathbf{F} = 0$ in pure phases. The two-phase collision operator $\Omega 2_i^{\sigma}$ is obtained from the colour gradient using the relation

$$\Omega 2_i^{\sigma} = \frac{A^{\sigma}}{2}|\mathbf{F}|\left[W_i \frac{(\mathbf{c}_i \cdot \mathbf{F})^2}{|\mathbf{F}|^2} - B_i \right] \qquad (i = 0,...,8) \tag{5.17}$$

The parameter A^{σ} is a free parameter controlling the surface tension. Since the magnitude of $\mathbf{F} = 0$ in pure phases, $\Omega 2_i^{\sigma}$ only contributes to mixed interfacial regions. The weights W_i are the same as those defined in the relation 5.9. The parameter B_i used on the calculation of $\Omega 2_i^{\sigma}$ is given by

$$B_i = \begin{cases} -\dfrac{4}{27} & (i = 0) \\ \dfrac{2}{27} & (i = 1,2,3,4) \\ \dfrac{5}{108} & (i = 5,6,7,8) \end{cases} \tag{5.18}$$

It should be noted that the two phase collision operator should ensure that the mass and momentum are conserved. The relations for momentum and mass conservation are given by

$$\sum_{i}^{8} \Omega 2_i^{\sigma} = 0 \tag{5.19}$$

$$\sum_{i}^{8} \Omega 2_i^{\sigma} \mathbf{c}_i = 0 \tag{5.20}$$

The pictorial representation of the colour gradient is shown in Figure 5.1. The colour gradient $\mathbf{F}(\mathbf{x})$ is normal to the interface. Thus the two-phase collision operator $\Omega 2_i^{\sigma}$ serves to add mass to populations in the direction of the colour gradient and depletes mass parallel to the interface.

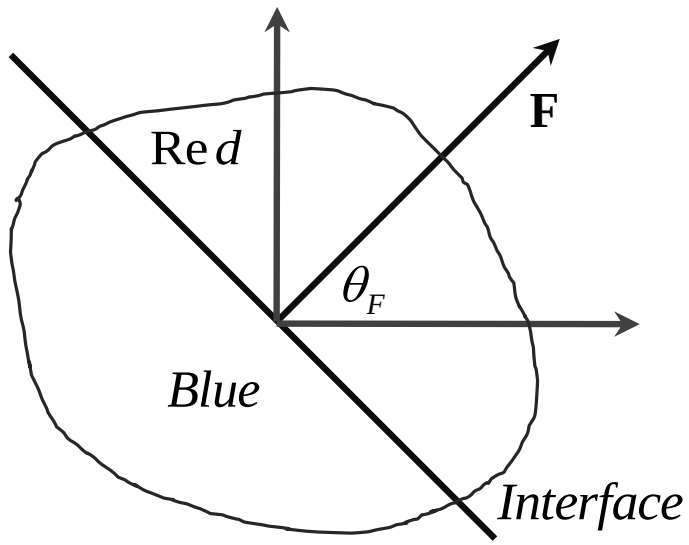


Figure 5.1: *Colour gradient* **F** *is normal to the interface*

5.1.2 Re-colouring

The re-colouring collision step redistributes the mass post two-phase collision step to achieve separation of the two fluids. Let $f_i^{\sigma'}$ and $f_i^{\sigma''}$ be the distributions post two-phase collision and re-colouring step respectively. Following the method proposed by Gunstensen *et al* [23], the re-colouring is carried out as follows:

1) The link vectors $\mathbf{c}_i$ are listed in descending order starting with the one nearest to the colour gradient vector $\mathbf{F}$.

2) The available numbers of red particles are sent in the directions close to **F** (i.e. perpendicular to the interface) and he blue particles are sent in the opposite direction.

It should be noted that the following constraints should be adhered to while performing the re-colouring operation.

$$f_i^{R'} + f_i^{B'} = f_i^{'} \tag{5.21}$$

$$\sum_i f_i^{R'} = \rho^R \tag{5.22}$$

Some potential draw backs in this re-colouring technique were pointed out by Latva-Kokko *et al* [22]. One such drawback is lattice-pinning. This situation occurs when one fluid is very close to the interface and the flow is too weak to move the red particles. Thus the interface gets pinned to that particular lattice. Alternative re-colouring techniques were proposed by Latva-Kokko *et al* [22] and Tölke *et al* [26] in which the separation of phases is not as strong as in the algorithm mentioned above, but the stability of the method is improved. It should be noted that the choice of the re-colouring algorithm does not affect the surface tension values. Due to ease of implementation the algorithm explained in Section 5.1.2 is used in this work.

5.1.3 Interface Relaxation parameter

The averaged relaxation parameter is an important factor in determining the thickness of an interface. When the relaxation parameters ω^σ are different, the viscosities of the fluid are also different, leading to an increase in the interface width. In order to ensure a stable interface and smooth change in viscosity, an order parameter $\psi = (\rho^R - \rho^B)/\left(\rho^R + \rho^B\right)$ and relaxation function is defined as follows

$$\omega = \begin{cases} \omega^R, & \psi > \delta, \\ f^R(\psi), & \delta \geq \psi > 0, \\ f^B(\psi), & 0 \geq \psi \geq -\delta, \\ \omega^B & \psi < -\delta \end{cases} \tag{5.23}$$

where $f^R(\psi)$ and $f^B(\psi)$ are given by

$$f^R(\psi) = \beta + \gamma\psi + \varepsilon\psi^2 \tag{5.24}$$

$$f^B(\psi) = \beta + \eta\psi + \xi\psi^2 \tag{5.25}$$

The constants $\beta, \gamma, \varepsilon, \eta$ and ξ are chosen in such a way that the relaxation parameter ω and its derivative are continuous functions. These constants can be determined from the following expressions.

$$\beta = \langle\omega\rangle \tag{5.26}$$

$$\gamma = \frac{2\left(\omega^R - \beta\right)}{\delta} \tag{5.27}$$

$$\varepsilon = -\frac{\gamma}{2\delta} \tag{5.28}$$

$$\eta = \frac{2\left(\beta - \omega^B\right)}{\delta} \tag{5.29}$$

$$\xi = \frac{\eta}{2\delta} \tag{5.30}$$

where $\langle\omega\rangle$ is the averaged relaxation parameter across the interface given by

$$\langle\omega\rangle = \frac{2\omega^R\omega^B}{\omega^R + \omega^B} \tag{5.31}$$

It should be noted that $\delta \leq 1$ is a free parameter which determines the interface thickness. The interface thickness varies from few lattice units to multiple lattice units when the value of δ varies from 0.1 to 1. Also the value of δ does not affect the thickness and the dynamics of the interface if the viscosities of the two fluids are equal.

5.1.4 Surface Tension

The magnitude of surface tension acting in planar interface region shown in Figure 5.2a as derived by Reis and Phillips [27] is given by

$$\sigma = \frac{A^R + A^B}{\omega} \tag{5.32}$$

Also the magnitude of surface tension acting on a thin plane interface parallel to y-axis as show in Figure 5.2b is given by

$$\sigma = \frac{5\rho\left(A^R + A^B\right)}{6\omega} \tag{5.33}$$

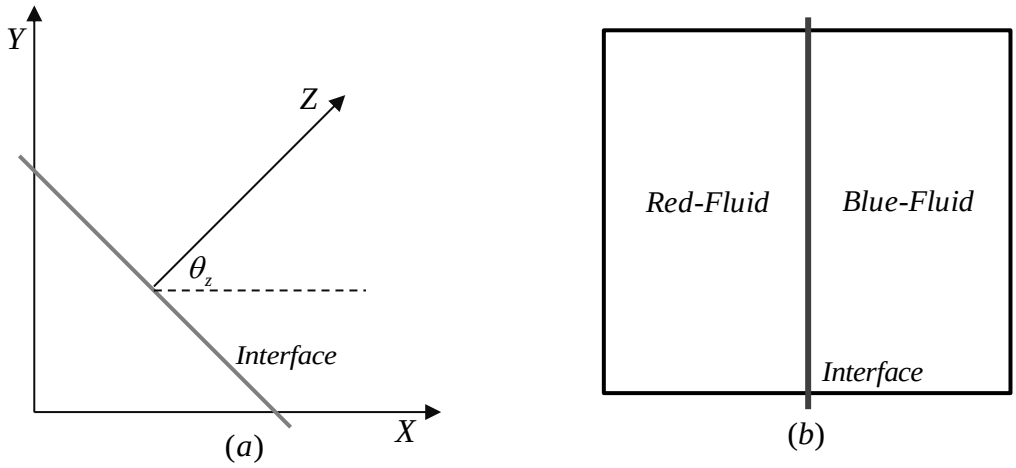


Figure 5.2: *(a) General plane interface. (b) Thin planer interface parallel to y-axis*

5.2 Simulation steps for Two-Phase flow

The procedure followed in this work for simulating the immiscible two-phase flow described in the above sections is outlined below.

1) After initializing the domain with initial conditions the relaxation parameters is determined at all the nodes using the Relation 5.24
2) The colour gradient is determined and the interface is located using the Relation 5.16
3) The equilibrium functions at all the lattice nodes are evaluated in all the directions using the Relations 5.6, 5.7 and 5.8.
4) The collision operation $\Omega 1_i^\sigma$ is performed using the Relation 5.5 and the relaxation parameters determined in step 1.
5) If the fluid flow is driven by body forces, the effect of body is included in the collision
6) The two-phase collision operation $\Omega 2_i^\sigma$ is performed based on the Relation 5.17 at lattice nodes where the magnitude of the colour gradient is positive.
7) The re-colouring operation is performed using the procedure described in Section 5.1.2
8) Streaming operation is performed for each of the phases using the Relation 2.11

9) The macroscopic variables are determined for post processing and for determination of the microscopic conditions for the next iteration.
10) The procedure mentioned in the above steps is performed till the steady state is reached.

CHAPTER 6

IMMISCIBLE TWO PHASE FLOW LBM NUMERICAL RESULTS

The immiscible two-phase lattice Boltzmann method discussed in Chapter 5 was implemented and simulation of a number of flow validation problems was performed. The aim of these simulations was to determine the accuracy and the robustness of the code and also to investigate issues like stability, compressibility errors, etc. The boundary conditions developed for single phase flow are also applicable for the two-phase flow problems. The two step bounce back method was used for implementing the wall boundaries. This method was chosen over the single step bounce back method because the two step method is second order accurate if the wall is assumed to be midway between the wall and the fluid node. Also the bounce back method is extremely convenient in modelling irregular boundaries. The Zou and He [15] technique is used for defining the pressure and velocities at the boundaries. The body forces were included using a post-collision operation which alters the momenta of the particle distribution functions as discussed in section (3.4).

6.1 Two Phase Poiseuille Flow

Consider a flow of two incompressible immiscible fluids, the high viscosity fluid sandwiched inside the low viscosity fluid, moving in the x-direction under the influence of a horizontal pressure gradient G as shown in Figure (6.1). Let the high viscosity and low viscosity fluid be the red and blue fluid respectively. If the flow is sufficiently small so that no instabilities occur within the interface then the analytical solutions for the steady flow are found to be [27]

$$u^B = \frac{G}{8}\left(\frac{3h^2}{\mu^R} + \frac{h^2 - 4y^2}{\mu^B}\right) \qquad -h \le y \le -\frac{h}{2} \tag{6.1}$$

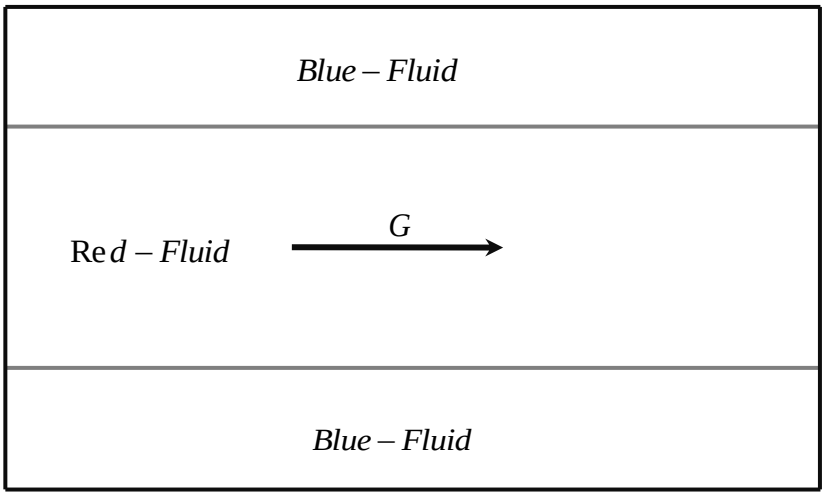


Figure 6.1: *Two-phase Poiseuille flow driven by body force G with red fluid sandwiched between blue fluid*

$$u^R = \frac{G}{2\mu^R}\left(h^2 - y^2\right) \qquad -\frac{h}{2} \le y \le \frac{h}{2} \tag{6.2}$$

$$u^B = \frac{G}{8}\left(\frac{3h^2}{\mu^R} + \frac{h^2 - 4y^2}{\mu^B}\right) \qquad \frac{h}{2} \le y \le h \tag{6.3}$$

where G is the body force and h is the channel height and μ^R and μ^B are the dynamic viscosities of red and blue fluid.

6.1.1 Problem Setup

The implemented LBM code was applied to a plane two-phase Poiseuille flow driven by a body force. The schematic diagram of a rectangular domain of size $L_x \times L_y = 128 \times 63$ LU's is shown in the Figure (6.1). The applied body force along the x-direction is given by $G = 0.00003$. Periodic boundaries were employed at the inlet and outlet boundaries in order to maintain a plane flow.

No-slip boundary condition was applied to the walls at the top and bottom using the two-step bounce back method. Since the two step bounce back method assumes the actual boundary to be present half way between the fluid and solid node, the effective height of the domain is $L_y - 1 = 62$ LU's. The initial condition of

$u_x = 0, u_y = 0$ and an average density of $\rho_0 = 1$ was applied in the interior of the domain. The following properties of the two fluids were used in the simulation.

Table 6.1: *Properties used for simulation*

Property	Blue Fluid	Red Fluid
A	0	0
α	0.1	0.1
μ	0.2525	0.795229
ω	0.795229	0.360685

The criteria used to determine the steady state is given by

$$\sum_i \frac{\left|u(\mathbf{x_i}, t+1) - u(\mathbf{x_i}, t)\right|}{\left|u(\mathbf{x_i}, t+1)\right|} \leq 10^{-6} \tag{6.4}$$

where the summation is over all the lattice nodes in the domain. The steady state solution was obtained only after a few thousand iterations depending on the value of viscosity and the boundary conditions. The simulations were performed using different values of δ (0.01, 0.1, 0.3, 0.5, 0.7 and 0.9) to study the effect of the interface thickness on the velocity profile obtained.

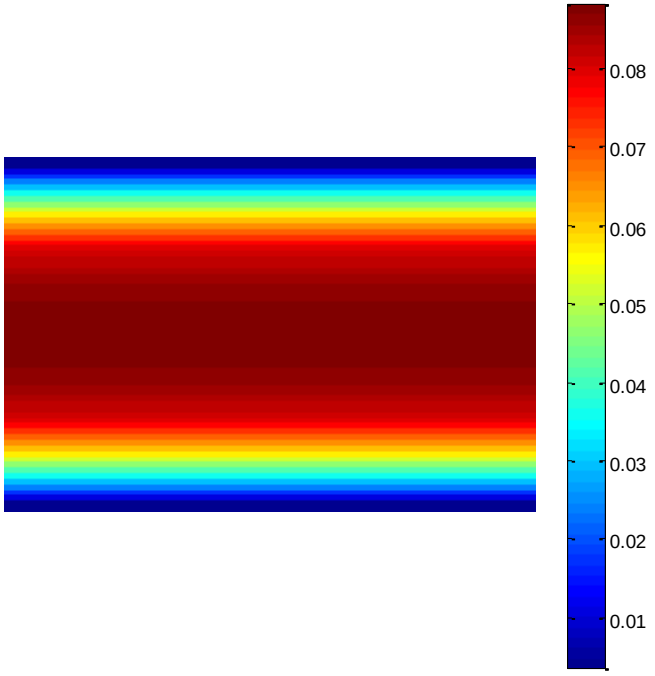


Figure 6.2: *Contour plot of the velocity profile for* $\delta = 0.1$

6.1.2 Results and Discussion

The velocity profile was measured at several cross sections. The magnitude of the vertical component of velocity u_y was always found to be smaller than 10^{-6} irrespective of τ .The steady state velocity contour plot obtained from the simulation for $\delta = 1.0$ is shown in Figure 4.2. Also a plot of comparison of the horizontal velocity profile with analytical velocity profile at various sections along the domain for $\delta = 1.0$ is show in

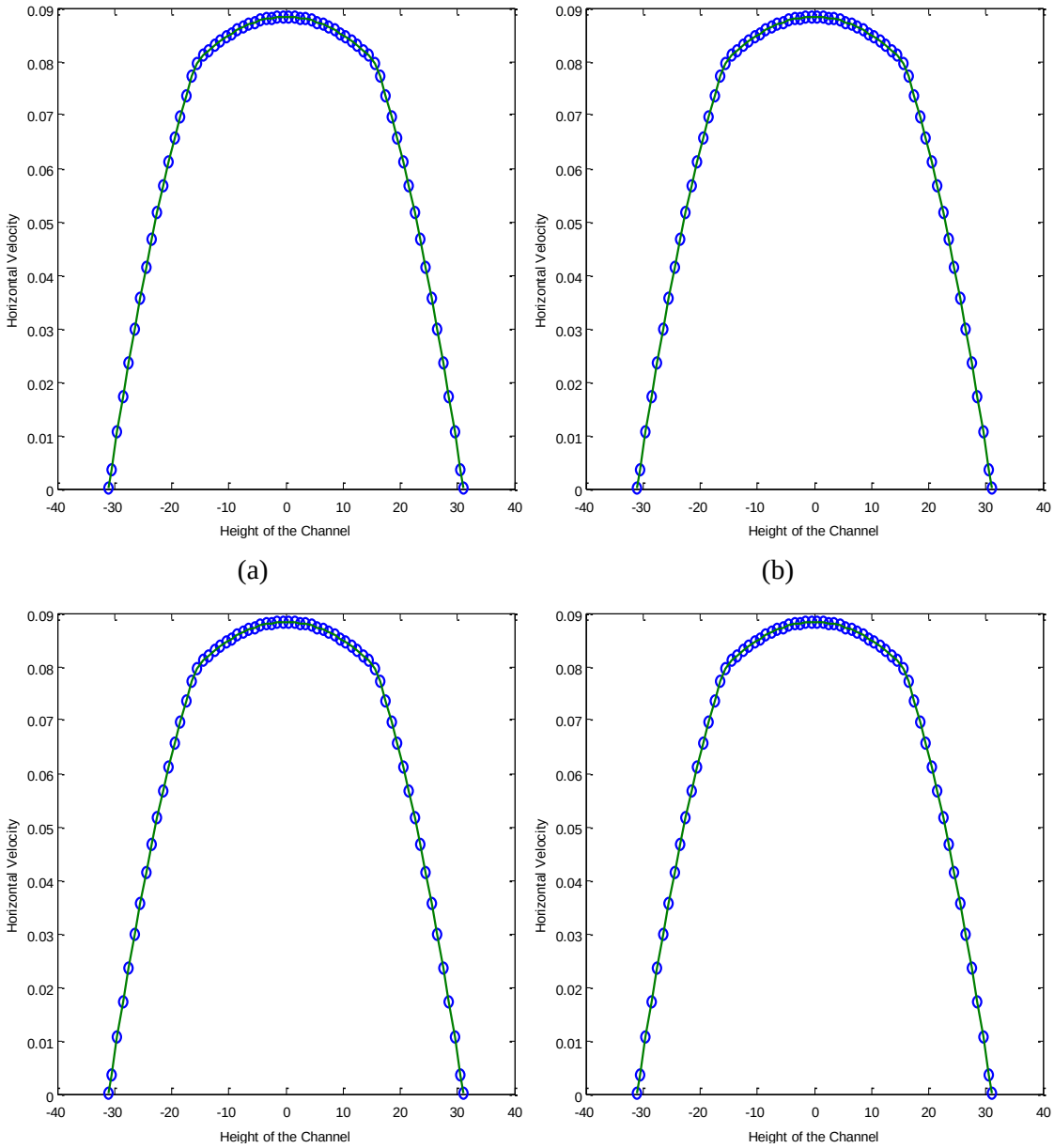


Figure 6.3: *Horizontal velocity comparison. (a) At x=1 (b) At x=32 (c) At x=96 (b) At x=128. [– indicates the analytical velocity and ⊙ indicates the actual velocity]*

Figure (6.3). It can be seen that the velocity profile obtained from the LB simulation matches well with the analytical solution at all the sections in the channel. The steady state velocity profile obtained for other values of δ is not shown since they were similar to plots shown in Figure (6.3)

In order to determine the effect of interface thickness on the LB simulation δ was varied from 0.01 to 0.9. It can be seen from Figure (6.4) that the maximum velocity relative error increases as the value of δ is increased towards 0.9. Hence for further two-phase LB simulations in this work, the value of δ is taken as 0.1.

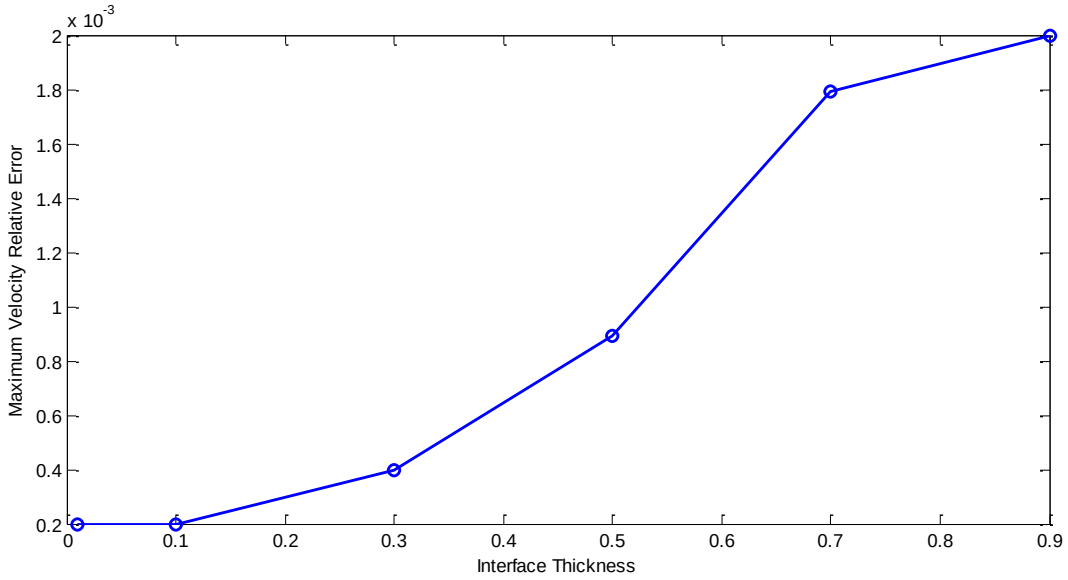


Figure 6.4: *Plot of the maximum velocity relative error v/s interface thickness* δ

6.2 Deformation of square droplet of one fluid in another fluid

When a square droplet of high density fluid is immersed in a low density fluid, the forces of surface tension cause it to deform and attain a circular shape. This behaviour is tested in the developed two-phase LB code.

6.2.1 Problem Setup

A red square droplet of size 32×32 LU's is immersed in larger blue droplet of size 64×64 LU's. The density of the red fluid and blue fluid is taken as $\rho^R = 2.0$ and $\rho^B = 1.0$ respectively thus creating a density ratio of 2.

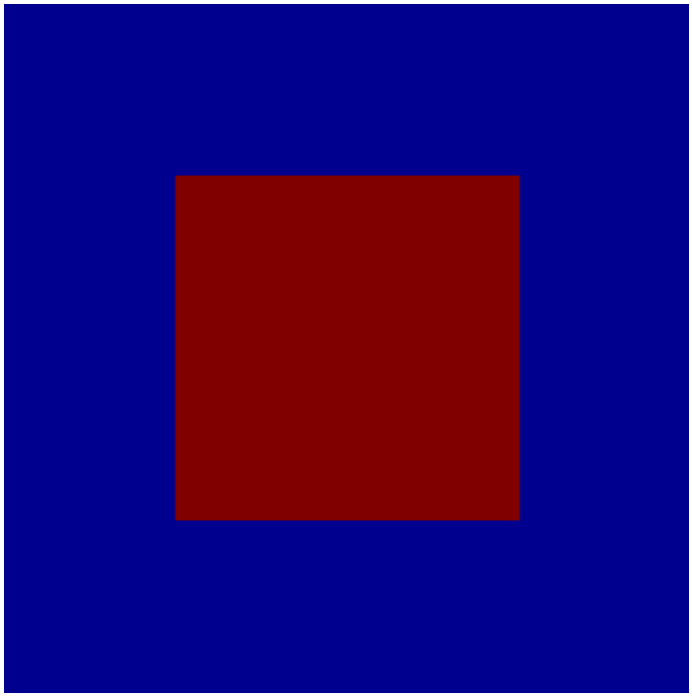


Figure 6.5: *Initial configuration- Deformation of square droplet of high density fluid (Red) inside low density fluid (Blue)*

The criterion used to determine the steady state is given by

$$\sum_i \frac{\left|\rho(\mathbf{x_i}, t+1) - \rho(\mathbf{x_i}, t)\right|}{\left|\rho(\mathbf{x_i}, t+1)\right|} \leq 10^{-6} \tag{6.5}$$

where the summation is over all the lattice nodes and ρ is the total density given by

$$\rho = \rho^R + \rho^B \tag{6.6}$$

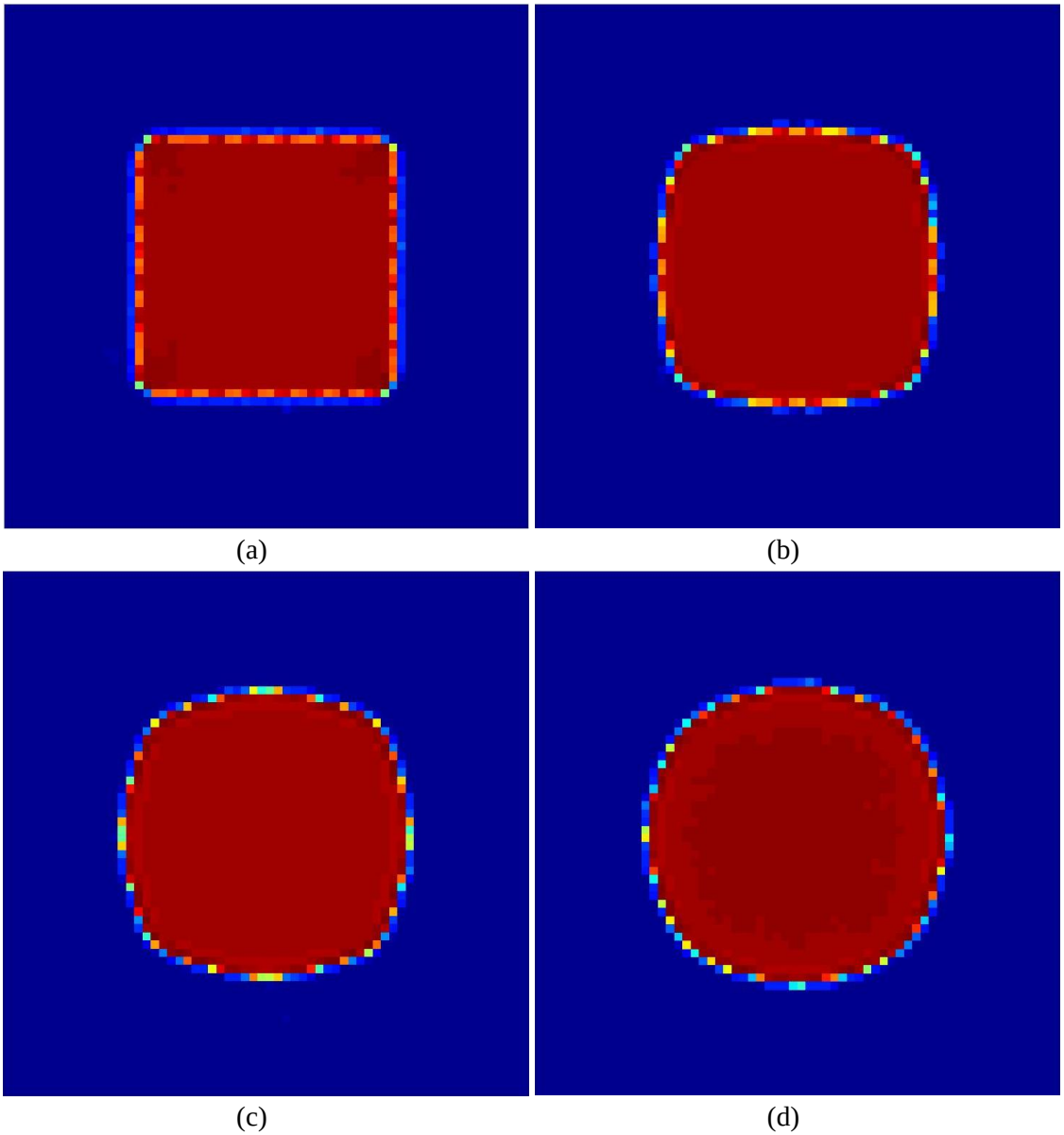
The parameters chosen for the simulation are tabulated below.

Table 6.1: *Properties used for simulation*

Property	**Blue Fluid**	**Red Fluid**
A	0.01	0.01
α	0	0.1
ν	1	1
ω	0.2857	0.2857

6.2.2 Results and Discussion

The initial configuration is shown in the Figure (6.5). The density profile of the red droplet at various time steps is shown in Figure (6.6). After 20 time steps the corners of



(a) (b)

(c) (d)

Figure 6.6: *Deformation of square droplet of high density fluid (a) At* $t = 20$, (b) At $t = 140$, (b) At $t = 300$,(b) At $t = 980$

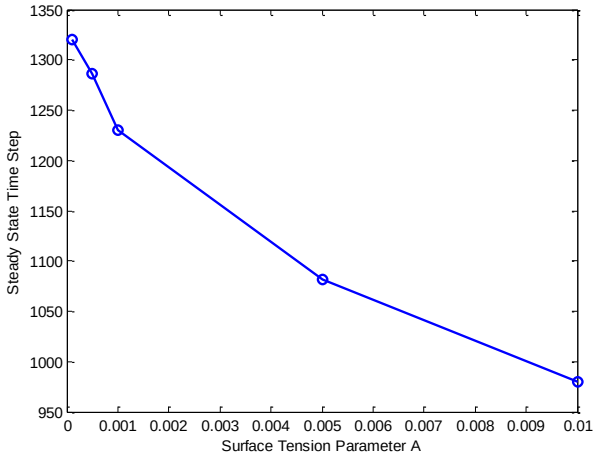


Figure 6.7: *Variation of simulation time with Surface tension parameter*

the square start collapsing due to the effects of surface tension. After 140 time steps the corners collapse completely which in turn starts pushing the centre of the edges outwards as shown in Figure (6.6b and 6.6c). This process continues till a steady state of smooth round shape is obtained as shown in Figure (6.6d).

In order to determine the effect of surface tension on the simulation, the surface tension parameter was varied in the range of 0.0001 to 0.01. The time step at which steady state was attained is plotted relative to the surface tension parameter in Figure (6.7). It was found that the number of time steps required in attaining steady state decreased with increase in surface tension parameter.

Also the radius of the circular droplet formed was found to be 18 LU's which is very close to the theoretical radius, 18.05, obtained from volume conservation principles.

6.3 Coalescence of two circular droplets of very high density

When two high density circular fluid droplets are immersed in a low density fluid, the forces of surface tension cause them to coalescence and fuse together forming a larger circular droplet.

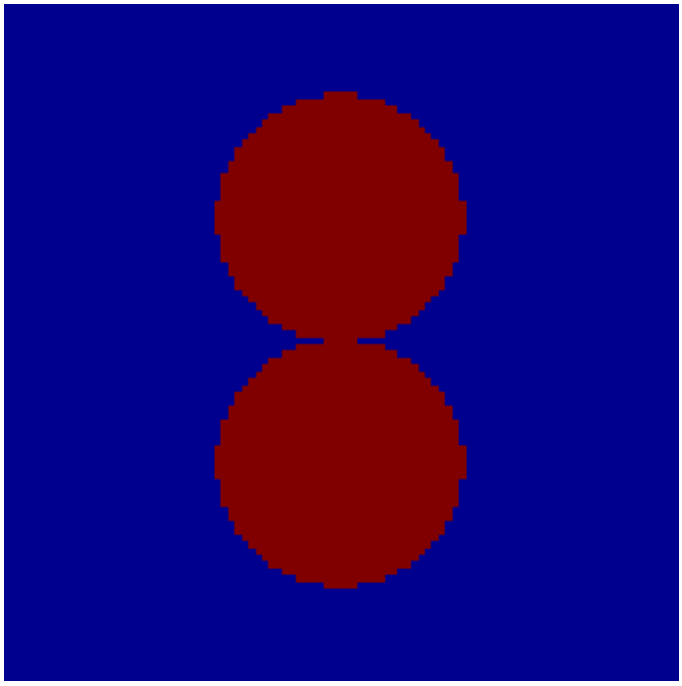


Figure 6.8: *Initial configuration- Coalescence of two circular droplets of very high density fluid (Red) inside low density fluid (Blue)*

In order to test the developed code for very high density difference multiphase flow, the above mentioned behaviour of the coalescence of two circular droplets is implemented.

6.3.1 Problem Setup

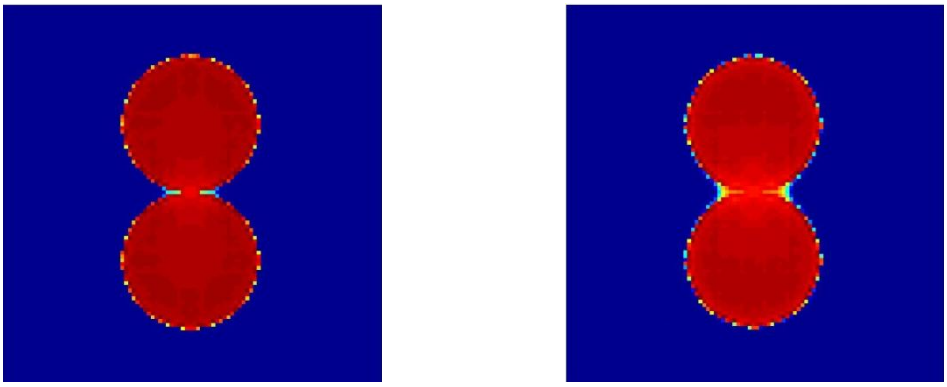
Two circular red droplets of radius 18 LU's are placed to each other and are immersed in blue droplet of size 100×100 LU's. The positions of the circular droplets are symmetric with respect to the geometric center along the y-direction. The density of the red fluid and blue fluid is taken as $\rho^R = 2.261$ and $\rho^B = 0.122$ respectively thus creating a density ratio of 18.5. The same criterion used in Section 6.2.1 is used to determine the steady state and is given by Equation (6.5).The parameters chosen for the simulation are tabulated below.

Table 6.2: *Properties used for simulation*

Property	Blue Fluid	Red Fluid
A	0.008	0.008
α	0.075	0.95
ν	1	1
ω	0.2857	0.2857

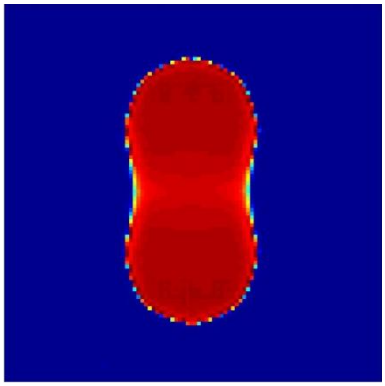
6.3.2 Results and Discussion

The initial configuration is shown in the Figure (6.8). The density profile of the red fluid at various time steps is shown in the Figure (6.9). The inter-molecular forces cause the bubbles to coalesce progressively over a period of time. The steady state solution is obtained after 4990 time steps.

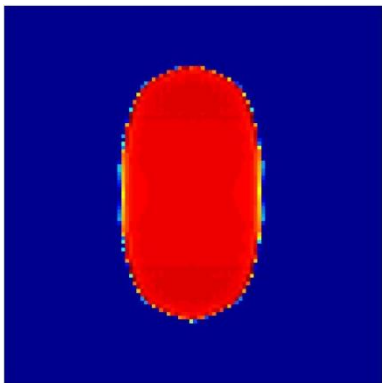


(a)

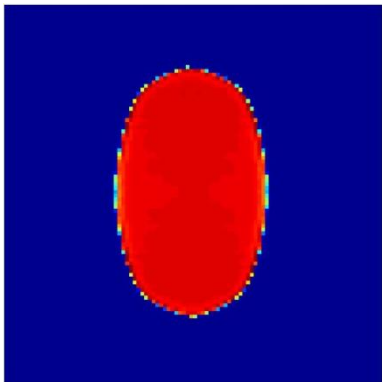
(b)



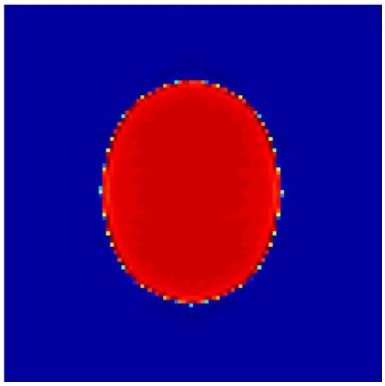
(c)



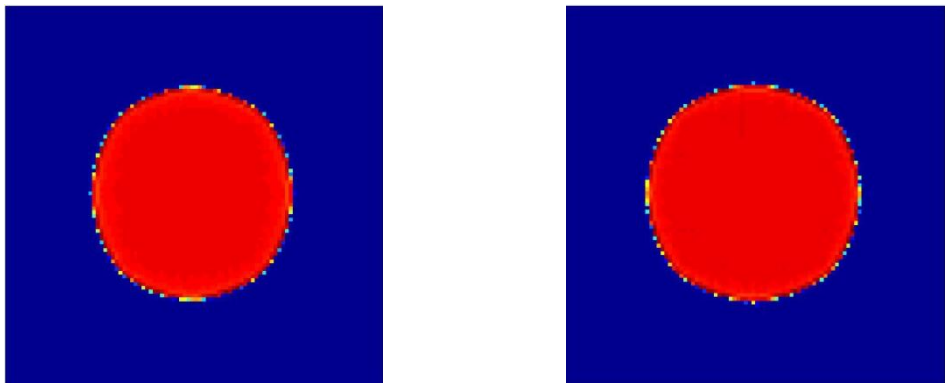
(d)



(e)



(f)



(g) (h)

Figure 6.9: *Coalescence of two circular droplets of very high density fluid (a) At* $t = 150$, *(b) At* $t = 450$, *(c) At* $t = 800$,*(d) At* $t = 1000$,*(e) At* $t = 2000$,*(f) At* $t = 3565$ *and (g) At* $t = 4990$

CHAPTER 7

CONCLUSION AND FURTHER RESEARCH

This thesis has presented the implementation of the lattice Boltzmann method for simulating incompressible two phase flow problems. This implementation was carried out through a sequence of steps. First the lattice Boltzmann method was implemented for single phase flow problems. The implemented code was tested for stability and accuracy by simulating a number of flow validating benchmark problems. The single phase lattice Boltzmann implementation was then extended to a framework for simulating incompressible two phase flow problems. The implemented two phase flow model was then subjected to numerical simulation of a number of two phase flow problems to study its accuracy and stability.

7.1 Discussion of Results

The implementation of the single phase lattice Boltzmann model was discussed in Chapter 2. The theoretical convergence behaviour of the method and the stability and accuracy constraints of the method were outlined.

The implementation of boundary conditions in the lattice Boltzmann method was outlined in Chapter 3. The various boundary conditions like the periodic boundary conditions, the no-slip boundary condition using single step and two step bounce back method, the velocity wall boundary conditions, the application of pressure and velocity at the boundary and inclusion of external forcing effects were explained in detail. Also it was outlined that two-step bounce back method for applying no-slip boundary condition is second order accurate if the solid wall is considered to be at the exact centre between the solid node and the fluid node.

The simulation of various benchmark problems using the implemented single phase LBM code was outlined in Chapter 4. The aim of the benchmark tests was to determine the accuracy and stability of the implemented method. The problem of 2D Poiseuille flow was implemented and simulations were performed for different values of relaxation parameters. It was found that accuracy of the solution decreased when the relaxation parameter was increased above the value of 1.0. When the relaxation parameter was kept within the range of 0.7 and 1.0 the velocity profiles obtained were exactly matching the velocity profile obtained from the analytical solution. The problem of 2D lid driven cavity flow was simulated using the developed single phase lattice Boltzmann code. Simulations were performed for Reynolds number values of 100 and 400 and the obtained horizontal and vertical velocities at the geometric centre matched exactly with the results obtained by Ghia *et al* [28].

The implementation of the two phase LBM was outlined in Chapter 5. This was done by implementing the colour gradient based two phase LBM algorithm proposed by Reis *et al* [27]. The algorithm involves the tracking the interface by evaluating the colour gradient at each of the nodes and adding a surface tension contribution to the nodes where a positive colour gradient exists. The post collision distributions were re-coloured to conserve mass before the streaming operation was performed. The dependence of the accuracy of the solution in the interface thickness was highlighted.

The numerical implementation of the immiscible two phase LBM code developed was described in Chapter 6. First the simulations of multiphase Poiseuille flow was performed with a layer of high viscosity fluid sandwiched between two layers of low viscosity fluid with various values of interface thickness. The accuracy of the solution seemed to reduce with an increase in interface thickness value. When the interface thickness was kept close to zero the velocity profile obtained matched accurately with the analytical solution. The second two phase simulation that was performed was the collapsing of a high density square bubble inside a low density fluid under the action of surface tension. The forces of surface tension caused the bubble to collapse to a circular shape. The radius of the circular bubble was found to be equal to the theoretical radius obtained from volume conservation principles. Next the simulation of coalescence of two very high density circular droplets inside a low density fluid was simulated. The two droplets gradually coalesced and formed a larger circular bubble with a radius very close to the theoretical value obtained from volume conservation.

7.2 Conclusions

1) The Lattice-Boltzmann method is based upon very simple conservation principles. In comparison to the top-down methods, the LBM is a more comprehensive approach consisting solely of local collision and propagation rules. The local nature of these rules also makes the LBM capable to parallelization.
2) The implemented LBM method code can be successfully used to simulate incompressible single phase flow and solutions very close to analytical solutions can be obtained
3) The implemented immiscible two phase LBM code can be successfully used to simulate a variety of two phase problems. It is also capable of simulating flows with very high density ratio.
4) In the case of immiscible two phase flow the interface can be successfully tracked and the thickness of the interface can be controlled by specifying a free parameter.

7.3 Recommendations for further research

1) The immiscible two phase LBM method can be extended to multiphase flow problems.
2) The size of the two-phase flow domain that can be solved in a serial processing architecture is very limited especially in the case of 3D flow problems. The implementation of the model in a parallel processing architecture would drastically increase the size of the domain that could be modelled.
3) Recently multiple-relaxation-time (MRT) collision operator has been used for simulating LBM flows which offer greater number of model parameters [39]. These parameters can be used to tune the hydrodynamic behaviour independent of the spatial and temporal discretisation. Thus the usage of MRT collision operators for multiphase flow simulations could be advantageous.

APPENDIX A

THE RELATIONSHIP BETWEEN LATTICE AND PHYSICAL UNITS

The variables in the lattice Boltzmann method are often reduced to lattice units as described in section (2.5). The relationship between two system for common variables used in LB simulation are outlined below from the work of [5].

Table A.1: *Relation between lattice and physical units*

Variables	**Physical Units**	**Lattice Units**	**Relationship**
Density	$\rho \approx \rho_0$	$\bar{\rho} \approx 1$	$\rho = \rho_0 \bar{\rho}$
Density Function	$f_i = \frac{\rho}{w_i}$	$\bar{f}_i = \frac{1}{w_i}$	$f_i = \rho_0 \bar{f}_i$
Relaxation Time	τ	$\bar{\tau}$	$\tau = \bar{\tau}\Delta t$
Lattice Spacing	Δx	$\Delta \bar{x} = 1$	-
Time Step	Δt	$\Delta \bar{t} = 1$	-
Lattice Speed	$c = \frac{\Delta x}{\Delta t}$	$\bar{c} = 1$	-
Viscosity	$\nu = \frac{1}{3}\left(\bar{\tau} - 1\frac{1}{2}\right)\frac{h^2}{\Delta t}$	$\bar{\nu} = \frac{1}{3}\left(\bar{\tau} - 1\frac{1}{2}\right)$	$\nu = \bar{\nu}\frac{h^2}{\Delta t}$
Displacement	$\mathbf{x}$	$\bar{\mathbf{x}}$	$\mathbf{x} = \Delta x \bar{\mathbf{x}}$
Velocity	$\mathbf{v} = \frac{d\mathbf{x}}{dt}$	$\bar{\mathbf{v}} = \frac{d\bar{\mathbf{x}}}{dt}$	$\mathbf{v} = c\bar{\mathbf{v}}$
Acceleration	$\mathbf{a} = \frac{d^2\mathbf{x}}{dt^2}$	$\bar{\mathbf{a}} = \frac{d^2\bar{\mathbf{x}}}{dt^2}$	$\mathbf{a} = \frac{c}{\Delta t}\bar{\mathbf{a}}$
Nodal Mass	$m = \rho(\Delta x)^D$	$\bar{m} \approx 1$	$m = \rho_0 (\Delta x)^D \bar{m}$

REFERENCES

[1] S. Chen and G. D. Doolen. Lattice Boltzmann method for fluid flows. Annual Review of Fluid Mechanics, Vol. 30: 329 -364

[2] S. Chen, H. Chen, D. Martinez, and W. Matthaeus. Lattice Boltzmann model for simulation of magneto hydrodynamics. Physical Review Letters, 67, 3776–3779 (1991)

[3] Shuling Hou, Qisu Zou, Shiyi Chen, Gary Doolen, and Allen C. Cogley. 1995. Simulation of cavity flow by the lattice Boltzmann method. J. Comput. Phys. 118, 2 (May 1995), 329-347.

[4] Bastien Chopard, Alexandre Masselot, Cellular automata and lattice Boltzmann methods: a new approach to computational fluid dynamics and particle transport, Future Generation Computer Systems, Volume 16, Issues 2-3, December 1999, Pages 249-257

[5] Feng, Y. T., Han, K. and Owen, D. R. J. (2007), Coupled lattice Boltzmann method and discrete element modelling of particle transport in turbulent fluid flows: Computational issues. International Journal for Numerical Methods in Engineering, 72: 1111–1134. doi: 10.1002/nme.2114

[6] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the Navier-Stokes equation. Physical Review Letters, 56(14):1505-1508, 1986

[7] Y. H. Qian, D. D'Humières and P. Lallemand. Lattice BGK Models for Navier-Stokes Equation. Europhys. Lett. 17 479 doi: 10.1209/0295-5075/17/6/001

[8] J. D. Sterling and S. Chen. Stability analysis of lattice boltzmann methods. Journal of Computational Physics, 123(1):196_206, 1996.

[9] G. R. McNamara and G. Zanetti. Use of the boltzmann equation to simulate lattice-gas automata. Physical Review Letters, 61(20):2332_2335, 1988.

[10] F. J. Higuera and J. Jimenez. Boltzmann approach to lattice gas simulations. Europhysics Letters, 9(7):663_668, 1989.

[11] H. Chen, S. Chen, and W. H. Matthaeus. Recovery of the Navier-stokes equations using a lattice-gas Boltzmann method. Physical Review A, 45(8):R5339-R5342, 1992.

[12] S. Chen, H. Chen, D. Martinez, and W. Matthaeus. Lattice Boltzmann model

for simulation of magnetohydrodynamics. Physical Review Letters, 67(27):3776-3779, 1991.

[13] P. A. Skordos. Initial and boundary conditions for the lattice Boltzmann method. Physical Review E, 48(6):4823-4842, 1993.

[14] Ladd A. Numerical simulations of fluid particulate suspensions via a discretised Boltzmann equation (Parts I & II). Journal of Fluid Mechanics 1994; 271:285–339.

[15] Zou, Q., and He, X. (1997). On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. Physics of Fluids 9, 1591-1598.

[16] He, X., and Luo, L.-S. (1997). Lattice Boltzmann Model for the Incompressible Navier–Stokes Equation. Journal of Statistical Physics 88, 927-944

[17] Sukop, M.C. and D.T. Thorne, Jr., 2006 (second printing 2007). Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers. Springer, Heidelberg, Berlin, New York 172 p.

[18] Xiaowen Shan and Hudong Chen. Lattice Boltzmann model for simulating flows with multiple phases and components. Phys. Rev. E 47, 1815–1819 (1993)

[19] J. Chin, E.S. Boek and P.V. Coveney, Lattice Boltzmann simulation of the flow of binary immiscible fluids with different viscosities using the Shan-Chen microscopic interaction model, Proc. R. Soc. Lond. A 360 (2002), pp. 547–558.

[20] Swift M, Osbourne W and Yeomans J.1995 Lattice Boltzmann simulations for non-ideal fluids. Phys. Rev. Lett. 75, 830–833 (1995)

[21] Li-Shi Luo. Theory of the lattice Boltzmann method: lattice Boltzmann models for nonideal gases. Phys. Rev. E 62, 4982–4996 (2000)

[22] M. Latva-Kokko and Daniel H. Rothman. Diffusion properties of gradient-based lattice Boltzmann models of immiscible fluids. Phys. Rev. E 71, 056702 (2005)

[23] Gunstensen, Andrew K. and Rothman, Daniel H. and Zaleski. Lattice Boltzmann model of immiscible fluids. Phys. Rev. A 43, 4320–4327 (1991)

[24] McNamara, Guy R. and Zanetti, Gianluigi. Use of the Boltzmann Equation to Simulate Lattice-Gas Automata. Phys. Rev. Lett. 61, 2332–2335 (1988)

[25] Rothman D H and Keller J M. Immiscible cellular-automaton fluids. Journal of Statistical Physics Volume 52, Numbers 3-4, 1119-1127, DOI: 10.1007/BF01019743

[26] Jonas Tölke, Manfred Krafczyk, Manuel Schulz and Ernst Rank. Lattice

Boltzmann Simulations of Binary Fluid Flow through Porous Media. Philosophical Transactions: Mathematical, Philosophical Transactions: Mathematical, Physical and Engineering Sciences. Vol. 360, No. 1792.

[27] T Reis and T N Phillips. Lattice Boltzmann model for simulating immiscible two-phase flows. 2007 J. Phys. A: Math. Theor. 40 4033

[28] U. Ghia, K. N. Ghia, and C. Y. Shin. High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method J. Comput. Phys. 48, 387 (1982)

[29] Christopher E. Brennen. Fundamentals of Multiphase Flow. Cambridge University Press, isbn 0521848040

[30] S. Shad, SPE, Schlumberger; B.B. Maini, SPE, I.D. Gates, Flow Regime Transfer Conditions for Two-Phase Flow in a Fracture, DOI:10.2118/137729-MS, Canadian Unconventional Resources and International Petroleum Conference

[31] F.H. Harlow, and J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow, Phys. Fluids 8, pp. 2182-2189, 1965.

[32] C.W. Hirt, and B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys., 39, pp. 201-225, 1981.

[33] J.A. Sethian, Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Cambridge University Press, 1996.

[34] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comp. Phys 100, pp. 25-37, 1992.

[35] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S.
Nas, and Y.J. Jan, A front-tracking method for the computations of multiphase flow, J.
Comput. Phys., 162, pp. 708-759, 2001.

[36] S.G. Yoon, A Fully Nonlinear Model for Atomization of High-Speed Jets, PhD thesis, Purdue University, 2002.

[37] A.R. Wadhwa, V. Magi, and J. Abraham, Numerical studies of droplet interactions, Proc. of International Conference on Liquid Atomization and Spray Systems, Sorrento, Italy, July 2003.

[38] J. M. Buick and C. A. Greated. Gravity in a lattice Boltzmann model. Phys. Rev. E 61, 5307–5320 (2000)

[39] P. Lallemand and L.-S. Luo. Theory of the lattice boltzmann method: Dispersion, dissipation, isotropy, galilean invariance, and stability. Physical Review E, 61(6):6546_6562, 2000