



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

**The Lattice Boltzmann Method for Fluid
Dynamics: Theory and Applications**

by

Chen Peng

Lausanne, January 20, 2011

submitted to the Department of Mathematics in EPFL
in partial fulfillment of the requirements for the degree of

Master of Mathematics

under the supervision of

Professor Alfio Quarteroni and Doctor Matteo Astorino

Abstract

Nowadays, the lattice Boltzmann method has been widely used by scientists and engineers as an alternative to conventional numerical solvers for the Navier-Stokes equations. The present work in this thesis aims at studying the lattice Boltzmann method for fluid dynamics from the kinetic point of view in contrast to macroscopic differential systems. And the main topics are concentrated on three aspects: theoretical background of lattice Boltzmann method, boundary conditions as well as multiblock and multigrid methods.

In the first part, the conceptual framework from statistical mechanics to fluid dynamics and the theoretical foundation for the lattice Boltzmann method are constructed. Both Grad's representation approach and Chapman-Enskog expansion approach are employed to derive the Navier-Stokes equations in fluid dynamics from Boltzmann - BGK equation. Different lattice structure in 1D, 2D and 3D are deduced from Gauss-Hermite quadrature. The general algorithm for lattice Boltzmann method as well as how to deal with compressibility effects and physical, dimensionless and lattice Boltzmann systems are summarized.

The second part is devoted to investigating different methods for imposing boundary conditions, in particular Dirichlet boundary conditions, on both straight and curved boundaries for lattice Boltzmann method. The difficulties of imposing macroscopic boundary conditions on kinetic distribution functions accurately and stably are addressed for straight and curved boundaries respectively, and two families of techniques under the name of “Distribution Modification” and “Distribution Reconstruction” are reviewed and presented with respect to these difficulties.

In order to push the lattice Boltzmann method to more practical applications, where multiscale physical problem in complex geometry and large scale parallel computation come on the stage, the use of multiblock and multigrid techniques is mandatory. Several multiblock and multigrid methods are examined and extended in the last part. Some basic ideas, such as overlapping and conservation laws, are borrowed from the field of domain decomposition and developed specifically for lattice Boltzmann method.

Poiseuille flow, lid driven cavity flow, Womersley flow, Taylor-Couette flow, blood flow applications are taken as benchmarks throughout the thesis.

Keywords: lattice Boltzmann method, fluid dynamics, Grad's moments system, kinetic representation, Chapman-Enskog expansion, Navier-Stokes equations, straight and curved boundary conditions, multiblock and multigrid methods

Acknowledgments

This thesis could not be finished without the help and support of many people who are gratefully acknowledged here.

First and foremost, I am honored to take this precious opportunity to express my deepest gratitude to my supervisor Professor Alfio Quarteroni. His enlightening instruction, constantly encouragement and trust without hesitation are the best gift for a student. His generous support and kindness are greatly appreciated. His keen and vigorous academic observation enlightens me not only in this thesis but also in my future study.

I would like to express my sincere appreciation to my supervisor Dr. Matteo Astorino, a respectable, responsible and resourceful scholar. He has provided me with valuable guidance in every stage of the writing of this thesis. Without his valuable suggestions and criticisms as well as impressive patience, I could not have completed my thesis.

I am indebted to all the members of the Chair of Modelling and Scientific Computing (CMCS). My horizon towards computational mathematics has been considerably broadened from various courses, seminars and discussions in this chair. Many aspects of what I have learned here are reflected in this thesis.

I am grateful to EPFL, especially the Department of Mathematics for the active academic atmosphere, friendly living environment and awarding me the Excellent Scholarship to support my study.

To my grandparents and parents, thanks so much for always supporting and understanding me in whatever I do.

To my girlfriend, thanks for the love and accompany, which are the first light of sunshine in the morning coming through the window of my life.

To the whole family of the Gampers in Switzerland, thanks for embracing me as one member of the warm and friendly family, I feel truly blessed and encouraged on my life and study.

Last but not least, thank all my friends in Lausanne, for the help and friendship.

Contents

1	Introduction	1
1.1	General review of the lattice Boltzmann method	1
1.2	Summary of the thesis	2
1.2.1	Theory of lattice Boltzmann method	3
1.2.2	Boundary conditions for lattice Boltzmann method	3
1.2.3	Multiblock and multigrid lattice Boltzmann methods	6
2	Theory of lattice Boltzmann method	8
2.1	Introduction	8
2.2	Statistical mechanics and Boltzmann equation	9
2.3	From Boltzmann-BGK equation to Navier-Stokes equations	11
2.4	Grad's representation approach	16
2.4.1	Hermite polynomials	16
2.4.2	Projection on Hermite bases	18
2.4.3	Euler equations	20
2.4.4	Navier-Stokes equations	21
2.4.5	Beyond Navier-Stokes equations	27
2.4.6	Discretization to numerical model	28
2.5	Chapman-Enskog expansion approach	33
2.5.1	Lattice Boltzmann-BGK Model	33
2.5.2	Chapman-Enskog expansion	34
2.5.3	Chapman-Enskog ansatz	41
2.6	Lattice Boltzmann Method	47
2.6.1	Gauss-Hermite quadrature and lattice structure	47
2.6.2	Algorithm for lattice Boltzmann method	50
2.6.3	Physical, dimensionless and lattice Boltzmann system	51
2.6.4	Compressibility effect of lattice Boltzmann method	53
2.7	Conclusion	53

3 Boundary conditions for lattice Boltzmann method	55
3.1 Introduction	55
3.2 Straight boundary conditions	57
3.2.1 Bounce-back boundary dynamics	59
3.2.2 Zou-He boundary dynamics	60
3.2.3 Finite difference boundary dynamics	62
3.2.4 Regularized boundary dynamics	64
3.3 Benchmarks with straight boundary	65
3.3.1 Poiseuille flow in 2D	65
3.3.2 Lid driven cavity flow in 2D	72
3.3.3 Womersley flow in 2D	75
3.4 Curved boundary conditions	80
3.4.1 Filippova-Hänel boundary dynamics	82
3.4.2 Bouzidi-Firdaouss-Lallemand boundary dynamics	85
3.4.3 Uniform interpolation boundary dynamics	86
3.4.4 Halfway bounceback boundary bynamics	87
3.4.5 Regularized interpolation/extrapolation boundary dynamics	87
3.5 Benchmarks with curved boundary	92
3.5.1 Laminar Taylor Couette flow in 2D	92
3.5.2 An example of blood flow in 2D	95
3.6 Conclusion	101
4 Multiblock and multigrid lattice Boltzmann method	102
4.1 Introduction	102
4.2 General ideas of multiblock and multigrid method	104
4.3 Multiblock lattice Boltzmann method	105
4.3.1 Multiblock without overlapping	105
4.3.2 Multiblock with overlapping	107
4.4 Multigrid lattice Boltzmann method	107
4.4.1 Grid refinement - Filippova and Hänel	108
4.4.2 Grid refinement - Dupuis and Chopard	110
4.4.3 Grid refinement - Lin and Lai	111
4.5 Numerical verification and application	114
4.5.1 Verification of multiblock and multigrid method	114
4.5.2 Initialization by multigrid lattice Boltzmann method	116
4.5.3 Aneurysm with a simplified geometric shape	118
4.6 Conclusion	120
5 Conclusions and Perspectives	123

List of Figures

1.2.1 organization for theory of lattice Boltzmann method	4
1.2.2 sketch for evolution of distribution function on boundary, blue region is fluid and red region is solid, flow in in middle and flow out on right	5
1.2.3 Left: curved boundary of porous media, Right: sketch of curved boundary	5
1.2.4 blood circulation in the blood vessels of human body	6
1.2.5 Tianhe-I, world's fastest supercomputer since Oct. 2010, equipped with 14,336 Xeon X5670 processors and 7,168 Nvidia Tesla M2050 general purpose GPUs.	7
2.6.1 general algorithm for lattice Boltzmann method	50
3.2.1 sketch for straight boundary condition with indexed distribution function as well as solid nodes and bulk nodes indexed with solid points and hollow points	57
3.2.2 sketch for fullway bounce back for straight boundary conditions. On the left boundary, the arrows indicate bounce back of the distribution functions, while on the right part in the bulk region, the arrows indicate how distribution functions evolve in the bulk; red arrows stands for post collision distribution functions while green ones represents distribution functions after streaming step	59
3.2.3 sketch for halfway bounce back for non-slip straight boundary	60
3.3.1 poiseuille flow driven by constant pressure gradient in 2D, with constant pressure $p = 0$ on the outlet boundary.	66
3.3.2 Error depending on N for Poiseuille flow with $\delta t = \delta x^2$	67
3.3.3 Poiseuille flow with initial velocity as 0 and constant pressure; pressure and velocity tend to be steady when the time goes long enough, e.g 100T	67
3.3.4 Poiseuille flow with $\delta t = 1/1000$ Left and Right $\delta t = \delta x/100$	68
3.3.5 testing the stability (measured by how high the Reynolds number can climb) of the four different methods for imposing boundary conditions, Poiseuille flow with $\delta t = \delta x^2$	69
3.3.6 CPU time comparison between different boundary conditions	70

3.3.7 CPU time comparison between different boundary conditions	70
3.3.8 CPU time comparison between different boundary conditions	71
3.3.9 Poiseuille flow with initial velocity as 0 and constant pressure; Left is the pressure and velocity at the initial time while Right shows at a quarter period $t = T/4$	71
3.3.10 Poiseuille flow with initial velocity as 0 and constant pressure; Left is the pressure and velocity at one period while Right shows at $t = 10T$	72
3.3.11 lid driven cavity flow in the region of a square box in 2D, with prescribed constant velocity in x-direction on the up boundary and zero velocity on the wall	73
3.3.12 Comparison of velocity by lattice Boltzmann method and finite element method when $Re = 100$. Left, $u_x, y = 0.5$ Right, $u_y, x = 0.5$	74
3.3.13 Velocity in L^2 norm, Left: $Re = 100$, Right: $Re = 1000$	74
3.3.14 Comparison of velocity by lattice Boltzmann method and finite element method when $Re = 1000$. Left, $u_x, y = 0.5$ Right, $u_y, x = 0.5$	75
3.3.15 Streamline of velocity, Left: $Re = 100$, Right: $Re = 1000$	75
3.3.16 $u_x, x = 1$ at the first quarter period with Womersley number 3.9633. Left: analytical solution. Right: numerical solution by compressible lattice Boltzmann method. Up: at time $t = T/8$. Down: at time $t = T/4$	76
3.3.17 Comparison between analytical velocity and numerical velocity $u_x, x = 1$ by compressible LBM within a quarter period with Womersley number 3.9633	77
3.3.18 $u_x, x = 1$ at the first half period with Womersley number 3.9633, by incompressible LBM	78
3.3.19 Comparison between analytical velocity and numerical velocity $u_x, x = 1$ within one period with Womersley number 3.9633 by incompressible LBM	79
3.3.20 $u_x, x = 1$ at the first half period with Womersley number 12.5331 by incompressible LBM	79
3.3.21 Comparison between analytical velocity and numerical velocity $u_x, x = 1$ within one period with Womersley number 12.5331 by incompressible LBM	80
3.4.1 sketch for curved boundary between solid and fluid (Left and Right), compared to straight boundary (Middle). The red arrow denotes the unknown information streaming from solid to fluid. Left: 1 unknown, Middle: 3 unknowns, Right: 5 unknowns	81
3.4.2 sketch for curved boundary condition, from [54]	82

3.4.3 three different configurations, black for fluid nodes, white for solid nodes, gray for boundary nodes. Left: boundary nodes locate in the fluid domain; Middle: boundary nodes sit on both the solid domain and the fluid domain, according to the distance to the boundary wall; Right: boundary nodes lie on the solid domain. P is one node taken for example. The illustration figures are taken from [58]	88
3.4.4 Left: interpolate the velocity on the boundary node P by its neighbors on the fluid domain and the boundary wall; Right: compute the velocity derivative by finite difference method for node P . The illustration figures are taken from [58]	89
3.5.1 sketch for Taylor Couette flow in between two cylinders, with radius of interior cylinder r_1 and r_2 for the exterior cylinder, steady flow counter-clockwise	92
3.5.2 numerical solution for Taylor Couette flow, viscosity $\nu = 0.01$	93
3.5.3 comparison of numerical error by different boundary dynamics	94
3.5.4 comparison of CPU time by the five methods for curved boundary	94
3.5.5 sketch for blood flow in the artery of the left arm with plaque enlarged	95
3.5.6 Computational domain is taken based on the rough sketch for blood vessel of the left arm; the part of vessel within the rectangle is narrowed down due to plaque	96
3.5.7 blood velocity within a period of one second	97
3.5.8 parabolic distribution of velocity on the inlet boundary at time $t = 0.2s$	98
3.5.9 simulation for pressure and velocity in the blood vessel at time $t = 0$	98
3.5.10 simulation for pressure and velocity in the blood vessel at time $t = 10s$	99
3.5.11 simulation for pressure and velocity in the blood vessel at time $t = 10.25s$	99
3.5.12 simulation for pressure and velocity in the blood vessel at time $t = 10.5s$	100
3.5.13 simulation for pressure and velocity in the blood vessel at time $t = 10.75s$	100
4.2.1 sketch for domain decomposition with two different layouts of interface.	104
4.2.2 sketch for multiblock and multigrid without overlap on the interface; grids on interface 1 between block 1 and block 2 are kept the same while grids on interface 2 between block 2 and block 3 are refined by ratio 2	104
4.2.3 sketch for multiblock and multigrid with one layer of overlap on the interface; grids on interface 1 between block 1 and block 2 are kept the same while grids on interface 2 between block 2 and block 3 are refined by ratio 2	105
4.3.1 sketch for multiblock without overlapping on the interface	106
4.3.2 sketch for multiblock with overlapping on the interface	107
4.4.1 sketch for grid refinement without overlapping on the interface	108

4.4.2 sketch for grid refinement with overlapping on the interface	110
4.4.3 sketch for composite grid refinement	112
4.4.4 sketch for for the process of composite grid refinement	112
4.5.1 sketch for steady numerical results of pressure and velocity for Poiseuille flow in three blocks sitting on two grids, by the scheme without overlapping	115
4.5.2 comparison between analytical pressure and numerical pressure by multi- block with overlapping and Dupuis and Chopard's multigrid with over- lapping along x axis in the middle of y direction, e.g. $y = 0.5$	116
4.5.3 sketch for multigrid lattice Boltzmann method for initialization	117
4.5.4 initialization by multigrid lattice Boltzmann method(MG-LBM) in hi- erarchical grids for Poiseuille flow with required error 10^{-6} , initial state (left) and final state (right) on each of the three levels	117
4.5.5 computational time for initialization by lattice Boltzmann method(LBM) and multigrid lattice Boltzmann method(MG-LBM) with required error 10^{-6}	118
4.5.6 sketch for simplified medical images of aneurysm	118
4.5.7 sketch for multiblock and multigrid for aneurysm in Figure 4.5.6, the gray region stands for the overlapping interface	119
4.5.8 velocity(Left) and pressure(Right) of blood flow for aneurysm in a period at time $t = 0s, 0.25s, 0.5s, 0.75s$	121
4.5.9 velocity(Left) and pressure(Right) of blood flow for aneurysm in a period at time $t = 10s, 10.25s, 10.5s, 10.75s$	122
A.0.1Gauss-Hermite quadrature formulas $E_{1,1}^1, E_{1,3}^2, E_{1,5}^3$	126
A.0.2Gauss-Hermite quadrature formulas $E_{2,5}^7$ Left: Structure Right: Mesh .	126
A.0.3Gauss-Hermite quadrature formulas $E_{2,5}^9$, Left: Structure Right: Mesh .	127
A.0.4Gauss-Hermite quadrature formulas $E_{3,5}^{15}$ Left: Structure Right: Mesh .	127
A.0.5Gauss-Hermite quadrature formulas Left : $E_{3,5}^{19}$ and Right: $E_{3,5}^{27}$	127

List of Tables

2.6.1 One-dimensional Gauss-Hermite quadrature formula	48
2.6.2 Two-dimensional Gauss-Hermite quadrature formula ($(\sqrt{3}, 0)_S$ are the full symmetric points and the same for that in three dimensional quadra- tures)	49
2.6.3 Three-dimensional Gauss-Hermite quadrature formula	49
4.5.1 Error comparison between different multigrid method	115

List of Symbols

- **Abbreviations**

- LGA: lattice gas automata
- LBM: lattice Boltzmann method
- BGK: Bhatnagar-Gross-Krook
- BBGKY: Bogoljubov-Born-Green-Kirkwood-Yvon
- FDM: finite difference method
- FEM: finite element method
- FVM: finite volume method
- SEM: spectral element method
- SRT: single relaxation time
- MRT: multiple relaxation time
- BCs: boundary conditions
- ICs: initial conditions
- Kn: Knudsen number
- Ma: Mach number
- Re: Reynolds number

- **Notations**

- t : time
- τ : relaxation time
- x : the position coordinate
- ξ : particle velocity
- p : pressure
- p : particle momentum

- \mathbf{g} : acceleration
- \mathbf{F} : external force
- f : distribution function
- f^i : distribution function on i^{th} level by Chapman-Enskog expansion $i = 0, 1, \dots$
- f^{eq} : equilibrium distribution function
- f^{neq} : non-equilibrium distribution function
- f^i : distribution function in the i^{th} direction $i = 0, \dots, q - 1$
- \tilde{f} : post collision distribution function
- Q : collision integral as $Q(f, f)$ or second order lattice symmetry as Q
- Ω : integral domain or collision operator as Ω_i
- $\sigma(\Omega)$: differential collision cross section
- ψ_k : k^{th} collision invariants $k = 0, 1, 2, 3, 4$
- ϕ : general collision invariants
- ρ : density
- \mathbf{u} : fluid physical
- $\mathbf{j} : \rho \mathbf{u}$ fluid momentum
- $\rho\epsilon$: kinetic energy
- S : pressure or stress tensor
- P : pressure or stress tensor
- \mathbf{q} : energy or heat flux
- R : third order moment
- Π^i : stress tensor on i^{th} level by Chapman-Enskog expansion $i = 0, 1, \dots$
- c_s : sound velocity
- m : mass of molecule
- T : absolute velocity or period

- k_B : Boltzmann constant
- $\theta : k_B T/m$
- \mathbf{c} : intrinsic velocity or mean velocity $\xi - \mathbf{u}$
- $I[\cdot]$: integral operator
- R^D : D dimensional space
- $w(\cdot)$: weight function for Hermit polynomials
- $\mathcal{H}^{(n)}(\cdot)$: n^{th} order Hermit polynomials
- $\boldsymbol{\alpha} : (\alpha_1, \alpha_2, \dots, \alpha_n)$ coordinate index
- I : identity matrix
- δ : $\delta_1 \delta_2 \cdots \delta_n$ Kronecker function
- $\delta_t : \Delta t$ temporal discretization step
- $\delta_x : \Delta x$ spatial discretization step
- $\mathbf{a}^{(n)}$: n^{th} order coefficient tensor of distribution function, $n = 0, 1, \dots$
- $\mathbf{a}_0^{(n)}$: n^{th} order coefficient tensor of equilibrium distribution function, $n = 0, 1, \dots$
- l : mean free length
- L : characteristic physical length
- $\varepsilon : l/L$ Knudsen number
- ν : dynamic shear viscosity
- ν' : dynamic bulk viscosity
- C : constant or correction term for Chapman-Enskog ansatz
- E : error in L_2 norm
- $E_{D,m}^d$: quadrature with d points in D dimension with m degree of precision
- $\mathbf{u}_{\cdot} : \cdot = p, d, lb$ velocity in physical, dimensionless or lattice Boltzmann system
- χ : weight for linear interpolation

- \mathbf{e}_i : i^{th} direction vector, $i = 0, 1, \dots, q - 1$

- **Operations**

- Tr : taking trace of matrix
- ∂_\cdot : partial derivative with respective to $\cdot = t, x, \xi, \rho, \mathbf{j}, \rho\epsilon$
- ∇ : nabla
- Δ : Laplace operator or ratio
- $a_i b_j$: Enstein notation $A = a_i b_j$ means that A_{ij} is the product of a_i and b_j
- $\mathbf{a} \cdot \mathbf{b}$: scalar product $\mathbf{a} \cdot \mathbf{b} = \sum_i a_i b_i$
- $\mathbf{a}\mathbf{b}$: tensor product $a_i b_j$
- $A : B$: contraction $A : B = \sum_{i,j} A_{ij} B_{ij}$

Chapter 1

Introduction

1.1 General review of the lattice Boltzmann method

During the last few years, the lattice Boltzmann method (LBM) has been developed from the cradle of statistical mechanics as a very good alternative numerical approach for modeling physical phenomena in fluid flows. In contrast to the traditional numerical schemes such as finite difference method (FDM), finite volume method (FVM), finite element method (FEM) or spectral element method (SEM), which are based on the discretization of macroscopic continuum equations, the lattice Boltzmann method is rooted in microscopic models and mesoscopic kinetic equations (see [2] [23], [31] [28] respectively). From the physical point of view, the lattice Boltzmann method can be interpolated as a microscope for fluid mechanics while a telescope for molecular dynamics [6], and it has been successfully applied for the coupling of molecular dynamics in the microscopic world and fluid dynamics in the macroscopic one [84].

Historically, the lattice Boltzmann method was derived from its forerunner lattice gas automata (LGA) and in particular from the FHP model, after the name of Frisch, Hasslacher and Pomeau [1]. The fundamental idea behind lattice gas automata is that microscopic interactions of artificial particles living on the microscopic lattice can lead to the corresponding macroscopic equations to describe the same fluid flows. During the interaction, which is consisting of collision and propagation or streaming with lattice velocities, lattice symmetry plays a key role for conserving mass and momentum as well as ensuring the angular momentum conservation. The first lattice gas automata with 4-fold rotational symmetry in square lattice was developed by Hardy, de Pazzis and Pomeau in 1973 [4], achieving only mass and momentum conservation. It was until 1986 that Frisch, Hasslacher and Pomeau triggered an avalanche with their hexagonal symmetry sufficient to conserve the angular momentum so as to retrieve not only diffusion equation, reaction-diffusion equation but also the Navier-Stokes equations [1].

However, the FHP model was proven to suffer from many diseases, such as statis-

tical noise, lack of Galilean invariance, low Reynolds number, exponential complexity and so on [6]. Initiated to get rid of statistical noise, McNamara and Zanetti [7] borrowed the idea of lattice Boltzmann equation from Frisch et al. [10] with the artificial particles replaced by continuous distribution functions. With linearization and simplification of the collision operator in lattice Boltzmann equation [3, 15], the lattice Boltzmann - BGK model was born and marked a milestone as a successful recipe for lattice gas automata in that it achieved the goal of more flexibility, Galilean invariant, high Reynolds number and much reduced complexity [73]. Later on, the lattice Boltzmann - BGK model stood on its own foot by direct phase space discretization of the continuous Boltzmann - BGK equation [43], and flourished with the emerge of multiple relaxation time model [8, 9] and entropic model [16, 19] with better performance on stability and accuracy.

Since the birth of lattice Boltzmann method, it has been witnessed that its applicable territory has spread throughout the kingdom of conventional numerical methods for fluid dynamics. The simulation of laminar and turbulent flows [44], suspension of colloidal particles [12], porous media [13], multiphase and multicomponent flows [34, 27], magnetohydrodynamics [21], more general fluid system as well as the possibility of large scale computation and coupling of problems from down to microscopic world up to mesoscopic world and to macroscopic world [84] are considerably benefited attributing to the flexibility, simplicity, intrinsic parallelism of the lattice Boltzmann method [6].

Despite the increasing popularity of the lattice Boltzmann method as an alternative numerical approach to conventional numerical methods in fluid dynamics, this novel approach still possesses several limitations [6, 11]. From the numerical point of view, it is possible [48] but very difficult in general for the lattice Boltzmann method to achieve higher than the second order of accuracy for both temporal and spatial discretization. Moreover, because there are more distribution functions than the hydrodynamic variables to keep track of, it can not avoid the relatively intensive computation as a trade-off for the simplicity and advanced parallel efficiency of the lattice Boltzmann method [32]. As for application, high-Mach number flows in aerodynamics are still difficult for the lattice Boltzmann method [17] because the expansion of equilibrium distribution function is based on small Mach number, and a consistent thermo-hydrodynamic scheme is still demanding [18, 39].

1.2 Summary of the thesis

The present work in this thesis aims at studying the lattice Boltzmann method for fluid dynamics and the main topics are concentrated on three aspects. Firstly, the theoretical

background for the lattice Boltzmann method developed as a very good alternative solver for the Navier-Stokes equations is investigated in Chapter 2. The second part, in Chapter 3 is devoted to one of the hottest topic in the lattice Boltzmann method: how to impose appropriately boundary conditions for both straight and curved boundary in a uniform grid. Lastly, multiblock and multigrid lattice Boltzmann methods are investigated as a fundamental step towards more practical applications and large scale parallel computations in Chapter 4. The benchmarks of Poiseuille flow, lid driven cavity flow, Womersley flow as well as Taylor-Couette flow are taken as numerical analysis and verification for the properties of stability, accuracy and efficiency of lattice Boltzmann method. More practical applications such as blood flow around aneurysm and through the artery located in the left arm are carried out. All these benchmarks and applications are implemented by Matlab. A brief introduction and motivation for each section is as follows:

1.2.1 Theory of lattice Boltzmann method

In Chapter 2, the Boltzmann equation will be introduced in the frame of statistical mechanics at first. After the linearization of Boltzmann equation to Boltzmann - BGK equation, the full derivation from Boltzmann - BGK equation to Navier-Stokes equations will be conducted in two approaches. The first one is to project the Boltzmann - BGK equation on Hermit basis and then truncate it up to certain order so as to obtain the corresponding conservation equations, under the name of Grad's representation approach. While the other one comes from lattice gas automata - Chapman-Enskog multiscale expansion for the lattice Boltzmann - BGK equation based on lattice symmetries. The compressible and thermal Navier-Stokes equations with independent bulk and shear viscosity will be obtained with the aid of Chapman-Enskog ansatz. In the last of this section, lattice structure will be given from Gauss-Hermit quadrature formulas and the algorithm for lattice Boltzmann method will be summarized. The organization of this chapter is shown in Figure 1.2.1.

1.2.2 Boundary conditions for lattice Boltzmann method

The central topic of Chapter 3 is how to impose boundary conditions for both straight boundary and curved boundary in the lattice Boltzmann method. Differently from the conventional methods such as finite difference method or finite element, the leading role playing on the stage of the lattice Boltzmann method is not macroscopic variables such as velocity or pressure, but rather microscopic distribution functions that stream across the boundary, see Figure 1.2.2. Since the distribution functions streaming from the solid into the bulk are generally unknown, we have to construct them based on the available

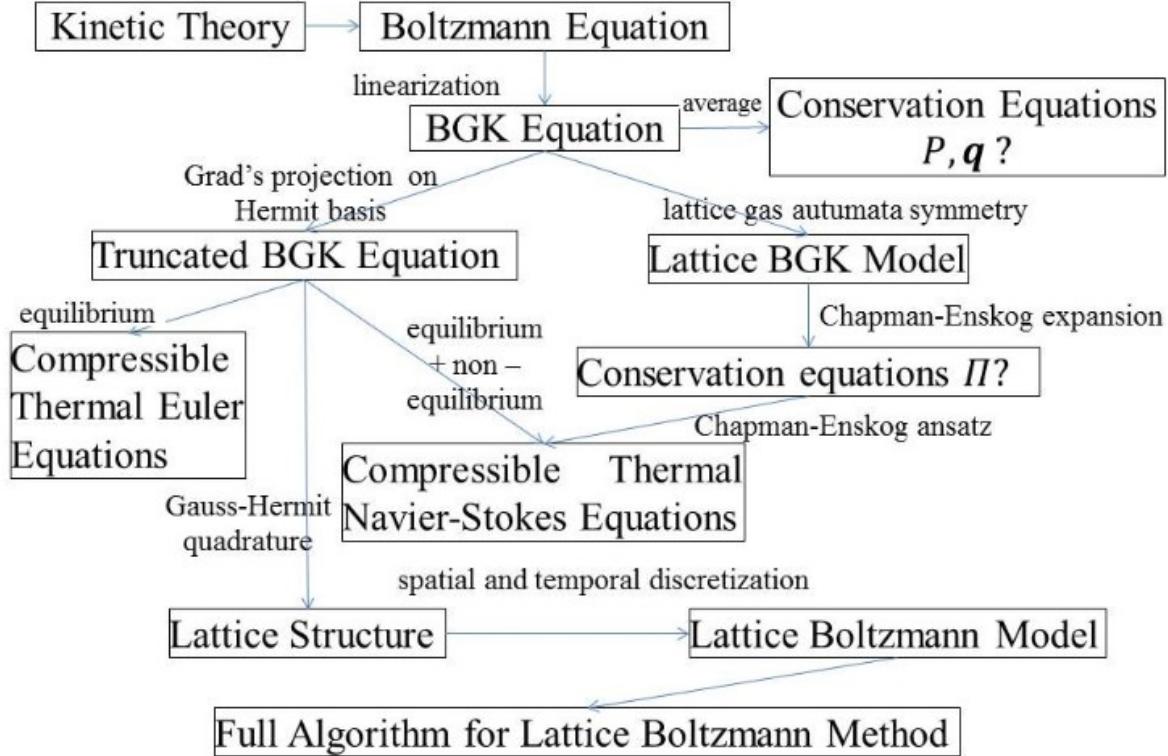


Figure 1.2.1: organization for theory of lattice Boltzmann method

macroscopic quantities on the boundary. The transformation from macroscopic to kinetic variables is one of the key difficulties in the lattice Boltzmann method.

Two distinct approaches are applied to address the difficulty of the boundary conditions for the lattice Boltzmann method. One is the so called “Distribution Modification” while the other is “Distribution Reconstruction”. By distribution modification it means that the unknown distribution functions are obtained by some physical rules such as bounce-back rule, mass and momentum conservation law or their combination. While by distribution reconstruction it stands for reconstructing all the distribution functions on the boundary by their explicit relation with the macroscopic variables - density, velocity and their derivatives, which can be approximated by the given boundary conditions.

Both straight boundary and curved boundary are considered. For straight boundary, different boundary dynamics by the two approaches are investigated and compared. As for more complex curved boundary of the computational domain, for example porous media (see left of Figure 1.2.3), interpolation or extrapolation is needed to approximate those variables sitting on the boundary nodes away from the boundary wall in the Cartesian grid (see right of Figure 1.2.3), in order to achieve precise simulation. This differs from finite element method since the nodes in the mesh of finite element

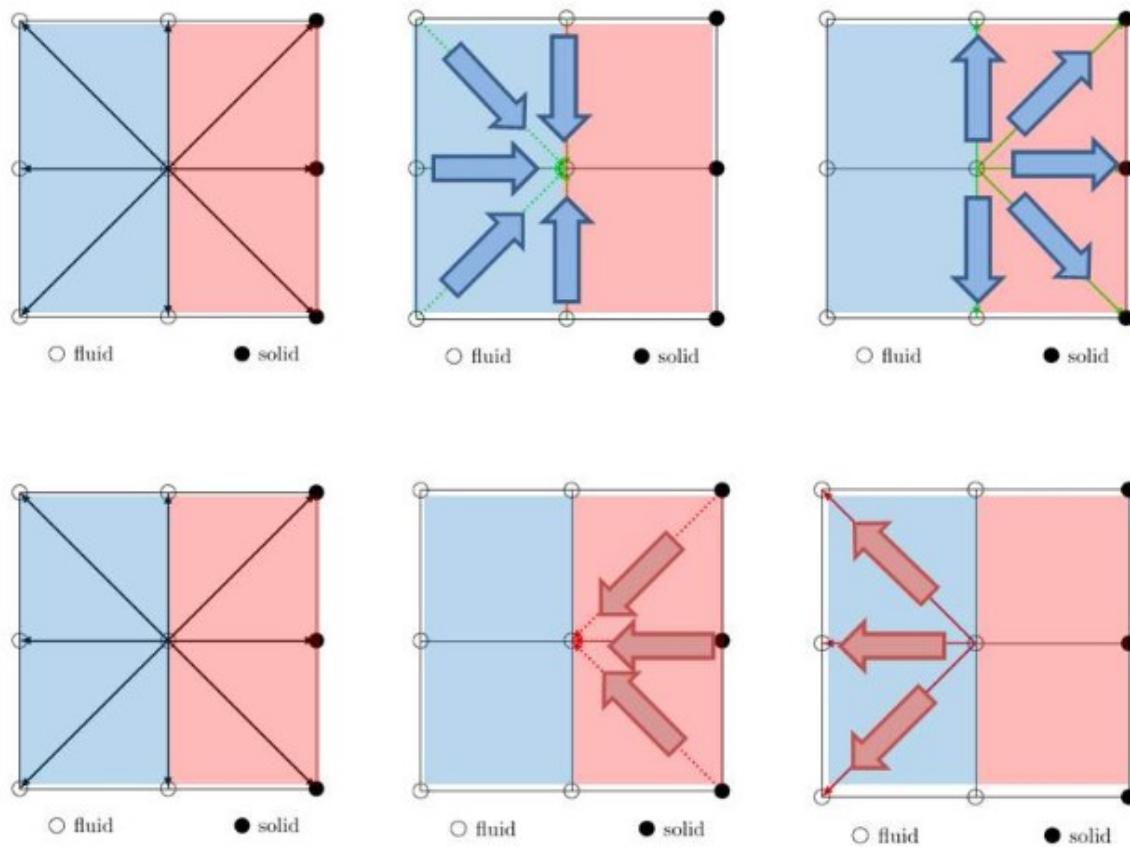


Figure 1.2.2: sketch for evolution of distribution function on boundary, blue region is fluid and red region is solid, flow in in middle and flow out on right

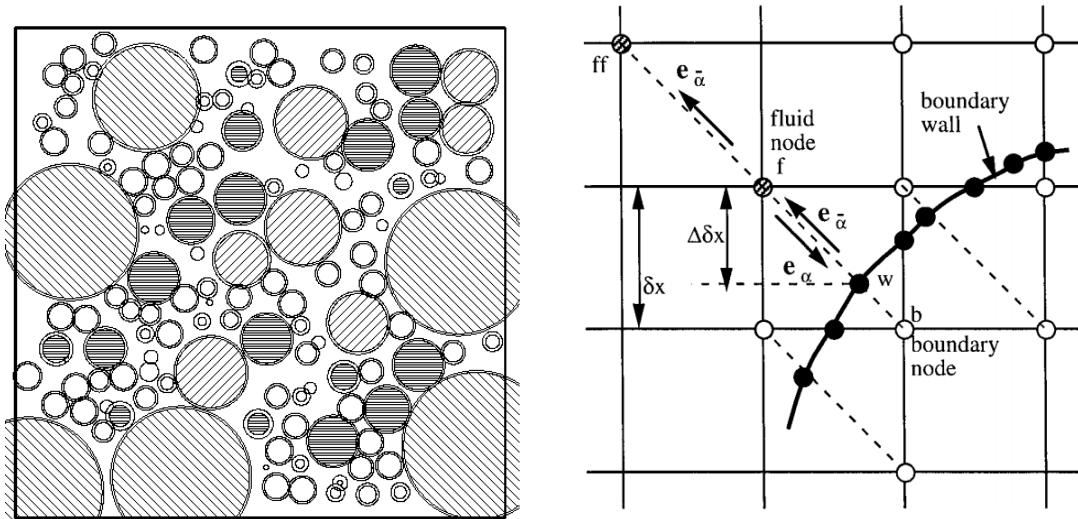


Figure 1.2.3: Left: curved boundary of porous media, Right: sketch of curved boundary

can be put on the curved boundary directly.

1.2.3 Multiblock and multigrid lattice Boltzmann methods

The lattice Boltzmann method was applied on uniform mesh in one single block at the beginning. However, this kind of mesh does not always meet our requirement in an efficient way, either in the sense of dealing with multiscale physical problem, or for memory limitations or for the sake of parallel computing.

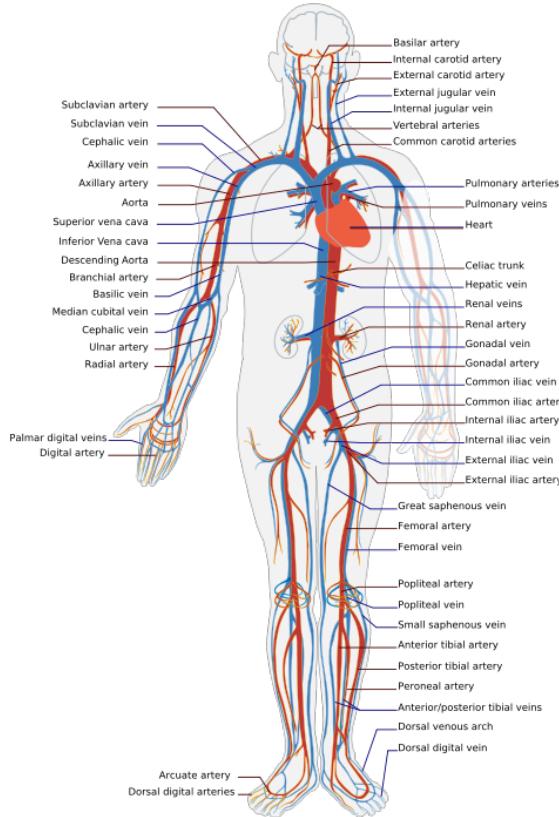


Figure 1.2.4: blood circulation in the blood vessels of human body

Take blood flow in human body for example as shown in Figure 1.2.4, the physical geometry is much more complicated, beyond the descriptive ability of uniform lattice in one single block. On the one hand, there are many branches of blood vessels sparsely distributed with relatively arbitrary geometry. Therefore, a uniform lattice covering the whole artery would cost considerable computer memory for the region without artery passing by. The recipe to deal with this problem is to design a certain number of blocks according to the shape of the artery. On the other hand, the radius of artery or vein is much larger than that of capillary. Hence, a uniform lattice would be either too coarse for capillary to capture the right behavior of the flow or too fine for artery

so that simulation will become considerable time consuming. This problem is tackled by grid refinement or multigrid method, where hierachic mesh is applied.

At the same time, from the computational point of view, these techniques represents one of the key elements for an efficient use of the lattice Boltzmann method in parallel structures of supercomputers [65], [66]. Take for example the world's fastest supercomputer (since Oct. 2010, see Figure 1.2.5[41]) "Tianhe-I" located in the National Supercomputing Center in China, it is capable of an Rmax (maximum range) of 2.566 petaFLOPS; that is, over 2.5 quadrillion (thousand million million) floating point operations per second, or at least 10^6 personal computers.



Figure 1.2.5: Tianhe-I, world's fastest supercomputer since Oct. 2010, equipped with 14,336 Xeon X5670 processors and 7,168 Nvidia Tesla M2050 general purpose GPUs.

In Chapter 4 we consider different multiblock and multigrid techniques for the lattice Boltzmann method. The main difficulty in these procedures comes from the exchange of information between the different (sub-)domains and different grid sizes. The information exchanged on interface, similar to that for boundary condition, could either be distribution function in the kinetic level or macroscopic variables such as density and velocity in the macro level. Since a direct modification of distribution function is easier and faster than reconstruction of distribution function from macroscopic variables, most of the grid refinement techniques have been developed in the kinetic level.

Chapter 2

Theory of lattice Boltzmann method

2.1 Introduction

The lattice Boltzmann method has been developed from lattice gas automata at first [3] and later on established as an independent numerical method derived from direct discretization of the Boltzmann - BGK equation [43]. The central topic of the theoretical background for the lattice Boltzmann method is how to derive from the kinetic Boltzmann - BGK equation formulated for particle distribution function to macroscopic conservation equations with thermodynamic variables such as density, momentum and energy [29] corresponding to the zeroth, first and second order moments of velocity. However, the conservation equations are not closed because of the presence of higher order moments of momentum stress tensor and energy flux.

Two different approaches have been developed to handle this problem. One is the well known Grad's 13-moments equations (Grad 1949 [24, 25]) and the other is the popular Chapman-Enskog multiscale expansion (Chapman and Cowling 1970 [26]). In the former approach, Grad claimed that the above five equations are not sufficient to describe all the properties of the fluid and another set of equations were proposed for stress tensor and energy flux, by projecting distribution function on Hermite polynomials. While in the latter approach according to Chapman and Cowling, the distribution function as well as temporal and spatial variables are expanded asymptotically in different powers of Knudsen number (denoted by ε in the following) so that the system is separated to multiple scales. By investigating the projection of distribution function on lattice symmetries (rooted in lattice gas automata) in different scales, we can further approximate the stress tensor and energy flux by the lower moments and their derivatives with respect to time and space and finally derive the macro differential equations for fluid dynamics, such as Euler equations and Navier-Stokes equations.

Comparatively, Grad's representation approach of projecting the distribution function on Hermite polynomials and truncating it at certain order is relatively easy without loss of the accuracy up to the truncated order, while Chapman-Enskog expansion can be used to analyze the accuracy of the approximation up to a certain order. It's worth to mention that Chapman-Enskog expansion is difficult to be applied for the derivation of differential system with moments higher than the second order, or in another way, it can be applied to derive Navier-Stokes equations but difficult to go beyond.

Both of the two approaches are explained in detail to have a more profound understanding of how to go from the kinetic Boltzmann - BGK equation to the macroscopic Navier-Stokes equations. In section 2.2, Boltzmann equation is introduced in the frame of statistical mechanics, followed by the derivation from Boltzmann - BGK equation to conservation equations in section 2.3. Then the Grad's kinetic representation approach is borrowed to close the conservation system in section 2.4, where both the continuous derivation to Euler equations and Navier-Stokes equations and numerical discretization to lattice Boltzmann - BGK model will be given. In section 2.5, Chapman-Enskog multiscale expansion is explored by starting directly from the lattice Boltzmann equation based on lattice symmetry and ending up with the Chapman-Enskog ansatz for the full derivation. In the last section 2.6, specific lattice structures and lattice Boltzmann models in 1D, 2D and 3D are deduced from Gauss-Hermit quadrature formulas.

2.2 Statistical mechanics and Boltzmann equation

Statistical mechanics was pioneered by the Austrian physicist Ludwig Boltzmann from the 1870s [5]. It provides a framework for the interpretation and prediction of the macroscopic bulk properties of materials by the microscopic properties of individual atoms and molecules. The atomistic dynamics that describes the motion of individual molecules by the deterministic Newton equations or Hamiltonian system and the thermodynamics that describes thermodynamic quantities such as pressure, momentum and energy by continuity, momentum and energy equations are bridged by the statistical mechanics in a probabilistic approach. The Boltzmann equation, established by Ludwig Boltzmann in 1872 is the cornerstone of the development of statistical mechanics [6].

Take 1cm^3 of air for example, there are $2.69 \cdot 10^{19}$ molecules at 0°C and a pressure of one atmosphere. The atomistic dynamics for the molecules of the air governed by

Newton equations as

$$\begin{aligned}\frac{d\mathbf{x}_i}{dt} &= \frac{\mathbf{p}_i}{m}, \\ \frac{d\mathbf{p}_i}{dt} &= \mathbf{F}_i,\end{aligned}\tag{2.2.1}$$

where $i = 1, \dots, N$, \mathbf{x}_i is the position coordinate of the i^{th} molecule, $\mathbf{p}_i := m\boldsymbol{\xi}_i$ the linear momentum and \mathbf{F}_i the external force from intermolecular interactions or external fields such as gravity or electric field. In 3 dimensional space there are totally $6N$ functions of time $(\mathbf{x}_i(t), \mathbf{p}_i(t)), i = 1, \dots, N$. Equations (2.2.1) provide a detailed description of all the N molecules, from which there is the possibility to capture the behavior of the whole system. However, as the example of air with volume 1cm^3 shows, N is generally too big for the ability of any foreseeable computer.

This problem is taken into account by a higher level of description with a system of N particles by distribution functions $f_N(\mathbf{x}_1, \mathbf{p}_1, \dots, \mathbf{x}_N, \mathbf{p}_N, t)$, where N is much smaller than the real number of molecules [11]. $f_N(\mathbf{x}_1, \mathbf{p}_1, \dots, \mathbf{x}_N, \mathbf{p}_N, t)d\mathbf{x}_1 d\mathbf{p}_1 \cdots d\mathbf{x}_N d\mathbf{p}_N$ is the probability to find a particle within the interval $[\mathbf{x}_1, \mathbf{x}_1 + d\mathbf{x}_1] \times [\mathbf{p}_1, \mathbf{p}_1 + d\mathbf{p}_1] \times \cdots \times [\mathbf{x}_N, \mathbf{x}_N + d\mathbf{x}_N] \times [\mathbf{p}_N, \mathbf{p}_N + d\mathbf{p}_N]$. f_N encompasses all statistical information of the dynamical process, and evolves according to the Liouville equation

$$\frac{\partial f_N}{\partial t} - \sum_{j=1}^{3N} \left(\frac{\partial H_N}{\partial x_j} \frac{\partial f_N}{\partial p_j} - \frac{\partial H_N}{\partial p_j} \frac{\partial f_N}{\partial x_j} \right) = 0,\tag{2.2.2}$$

where H_N is the Hamiltonian of the system. Integrating over part of the phase space, we define the reduced densities as

$$F_s(\mathbf{x}_1, \mathbf{p}_1, \dots, \mathbf{x}_s, \mathbf{p}_s, t) := V^s \int f_N(\mathbf{x}_1, \mathbf{p}_1, \dots, \mathbf{x}_N, \mathbf{p}_N, t) d\mathbf{x}_{s+1} d\mathbf{p}_{s+1} \cdots d\mathbf{x}_N d\mathbf{p}_N,\tag{2.2.3}$$

where V^s is a normalization factor. A coupled system of differential equations for $F_s (1 \leq s \leq N)$ with the name of BBGKY [14] after Bogoljubov, Born, Green, Kirkwood and Yvon has been shown to be equivalent to the Liouville equation. The Boltzmann equation has been derived from BBGKY by the assumption of two-partial local collisions with uncorrelated velocities before collision and free of external forces. The integro-differential Boltzmann equation describes a single partial distribution function $f(\mathbf{x}, \boldsymbol{\xi}, t) \propto F_1(\mathbf{x}_1, \mathbf{p}_1, t)$ as

$$(\partial_t + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} + \mathbf{g} \cdot \nabla_{\boldsymbol{\xi}})f(\mathbf{x}, \boldsymbol{\xi}, t) = Q(f, f),\tag{2.2.4}$$

where $\mathbf{g}(\mathbf{x}, t)$ is the acceleration depending only on space and time, with $F(\boldsymbol{\xi}) = -\mathbf{g} \cdot \nabla_{\boldsymbol{\xi}} f$ as the body force. The collision integral $Q(f, f)$ with $\sigma(\Omega)$ the differential

collision cross section for the two-particle collision which transforms from the incoming velocity (ξ, ξ_1) to the outgoing velocity (ξ', ξ'_1) is defined as

$$Q(f, f) = \int d\xi_1 \int d\Omega \sigma(\Omega) |\xi - \xi_1| (f(\xi') f(\xi'_1) - f(\xi) f(\xi_1)). \quad (2.2.5)$$

It can be shown [20] that the collision integral has five elementary collision invariants $\psi_k(\xi)$, $k = 0, 1, 2, 3, 4$ as

$$\int Q(f, f) \psi_k(\xi) d\xi = 0. \quad (2.2.6)$$

where $\psi_0 = 1$, $(\psi_1, \psi_2, \psi_3) = \xi$, $\psi_4 = \xi^2$ related to mass, momentum and kinetic energy. Written in a linear combination, the general collision invariants $\phi(\xi)$ is as

$$\phi(\xi) = a + \mathbf{b} \cdot \xi + c \xi^2. \quad (2.2.7)$$

It can also be shown that only the functions with the form

$$f(\xi) = \exp(a + \mathbf{b} \cdot \xi + c \xi^2) \quad (c < 0) \quad (2.2.8)$$

can satisfy the null collision integral $Q(f, f) = 0$. The special case of the Maxwell-Boltzmann equilibrium distribution function is defined as

$$f^{(0)}(\mathbf{x}, \xi, t) = \frac{\rho(\mathbf{x}, t)}{(2\pi\theta(\mathbf{x}, t))^{D/2}} \exp\left(-\frac{(\xi - \mathbf{u})^2}{2\theta(\mathbf{x}, t)}\right), \quad (2.2.9)$$

where $\rho(\mathbf{x}, t)$ and $\mathbf{u}(\mathbf{x}, t)$ are the density and velocity of the fluid at spatial position \mathbf{x} and time t ; θ is defined as $k_B T / m$ with k_B the Boltzmann constant, m the molecular mass of gas particles and T the absolute temperature under our consideration (one notices that for isothermal fluid T is constant). As can be seen from (2.2.5), the complicated collision integral is a big obstacle when dealing with Boltzmann equation. In order to simply the collision integral $Q(f, f)$ to a convenient integral $J(f)$, two rules have to be obeyed. The first one is that $J(f)$ conserves all the collision invariants $\psi_k(\xi)$ of $Q(f, f)$. While the other one is that the collision term should bear the tendency of distribution function to the Maxwell-Boltzmann distribution function by H-theorem [11]. Both of the two rules are fulfilled by the widely used BGK model developed by Bhatnagar, Gross and Krook [22], which will be defined in the next section.

2.3 From Boltzmann-BGK equation to Navier-Stokes equations

The Boltzmann - BGK equation in D dimensional space is defined as

$$(\partial_t + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} + \mathbf{g} \cdot \nabla_{\boldsymbol{\xi}}) f(\mathbf{x}, \boldsymbol{\xi}, t) = -\frac{1}{\tau} (f - f^{(0)}), \quad (2.3.1)$$

where τ is the relaxation time and $f^{(0)}$ is the Maxwell-Boltzmann distribution function in equilibrium state as defined in (2.2.9). Here and throughout the following context, we rescale all the velocity $\boldsymbol{\xi}$, \mathbf{u} by the sound velocity $c_s = \sqrt{k_B T/m}$ and θ by c_s^2 so that all the variables become dimensionless. Especially, $\theta = 1$ for isothermal fluid. The rescaled equilibrium distribution function is still given by (2.2.9).

Compared to the macroscopic thermodynamic variables, the kinetic distribution function provides much more detailed information. In fact, the macroscopic thermodynamic variables can be expressed as the moments of the distribution function with different order with respect to velocity. Specifically, we have

$$\text{mass density: } \rho(\mathbf{x}, t) = \int f(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi}, \quad (2.3.2)$$

$$\text{momentum density: } \rho \mathbf{u}(\mathbf{x}, t) = \int \boldsymbol{\xi} f(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi}, \quad (2.3.3)$$

$$\text{kinetic energy density: } \rho \epsilon(\mathbf{x}, t) = \frac{1}{2} \int \mathbf{c}^2 f(\mathbf{x}, \boldsymbol{\xi}, t) d\mathbf{c}, \quad (2.3.4)$$

$$\text{pressure or stress tensor: } P = \int \mathbf{c} \mathbf{c} f(\mathbf{x}, \boldsymbol{\xi}, t) d\mathbf{c}, \quad (2.3.5)$$

$$\text{energy or heat flux: } \mathbf{q} = \frac{1}{2} \int \mathbf{c} \mathbf{c}^2 f(\mathbf{x}, \boldsymbol{\xi}, t) d\mathbf{c}, \quad (2.3.6)$$

where $\mathbf{c} = \boldsymbol{\xi} - \mathbf{u}$ is defined as the intrinsic velocity or mean velocity. The form of \mathbf{c}^2 stands for the inner product $\mathbf{c}^2 = c_i c_i$ while $\mathbf{c}\mathbf{c}$ represents the tensor product such that $(\mathbf{c}\mathbf{c})_{ij} = c_i c_j$, where the subscripts i, j denote Cartesian components and Einstein summation convention is used for repeated indices in the component notation, with the meaning of $c_i c_i = \sum_i c_i^2$.

If stories for fluid in the mesoscopic world are all about distribution function f subjected to Boltzmann-BGK equation, then the bridge to go to the macroscopic world from the mesoscopic one are velocity average (integral) in different order of moments with respect to the distribution function. When we are in the macroscopic world with its own variables such as density ρ and velocity \mathbf{u} , what we would play should be submitted to Newton's laws, including mass conservation, momentum conservation and energy conservation in the realm of classical fluid mechanics. Now let's cross the bridge, bearing in mind our goal to go from Boltzmann-BGK equation to Euler equations to Navier-Stokes equations and beyond.

Define an integral operator $I[\cdot]$ - as the rule to cross the bridge as follows

$$I[h](\mathbf{x}, t) = \frac{1}{\rho(\mathbf{x}, t)} \int_{R^D} h(\boldsymbol{\xi}) f(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi}, \quad (2.3.7)$$

where $h(\boldsymbol{\xi})$ is an integrable function dependent only on the velocity $\boldsymbol{\xi}$. In particular, $1, \boldsymbol{\xi}, \boldsymbol{\xi}^2/2$, corresponding to mass, momentum and energy are chosen among the moments of velocity. In general, multiplying by $h(\boldsymbol{\xi})$ and integrating both sides of the Boltzmann-BGK equation (2.4.16) over the velocity space, we obtain

$$\int (\partial_t f + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} f + \mathbf{g} \cdot \nabla_{\boldsymbol{\xi}} f) h d\boldsymbol{\xi} = -\frac{1}{\tau} \int (f - f^{(0)}) h d\boldsymbol{\xi}. \quad (2.3.8)$$

When $h(\boldsymbol{\xi}) = 1, \boldsymbol{\xi}, \boldsymbol{\xi}^2/2$ respectively, the right hand side of (2.3.8) vanishes thanks to the conservation laws of mass, momentum and kinetic energy for ideal collision. While for the left hand side, term by term we have

$$\int (\partial_t f) h(\boldsymbol{\xi}) d\boldsymbol{\xi} = \partial_t \left(\frac{\rho}{\rho} \int f h(\boldsymbol{\xi}) d\boldsymbol{\xi} \right) = \partial_t (\rho I[h(\boldsymbol{\xi})]), \quad (2.3.9)$$

$$\begin{aligned} \int (\boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} f) h(\boldsymbol{\xi}) d\boldsymbol{\xi} &= \nabla_{\mathbf{x}} \cdot \left(\int h(\boldsymbol{\xi}) \boldsymbol{\xi} f d\boldsymbol{\xi} \right) - \int f \underbrace{\nabla_{\mathbf{x}} \cdot (\boldsymbol{\xi} h(\boldsymbol{\xi}))}_{=0} d\boldsymbol{\xi} \\ &= \nabla_{\mathbf{x}} \cdot \left(\frac{\rho}{\rho} \int h(\boldsymbol{\xi}) \boldsymbol{\xi} f d\boldsymbol{\xi} \right) = \nabla_{\mathbf{x}} \cdot (\rho I[h(\boldsymbol{\xi}) \boldsymbol{\xi}]), \end{aligned} \quad (2.3.10)$$

$$\begin{aligned} \int (\mathbf{g}(\mathbf{x}, t) \cdot \nabla_{\boldsymbol{\xi}} f) h(\boldsymbol{\xi}) d\boldsymbol{\xi} &= \int (h(\boldsymbol{\xi}) \mathbf{g}(\mathbf{x}, t)) \cdot \nabla_{\boldsymbol{\xi}} f d\boldsymbol{\xi} \quad (\text{by Green's formula}) \\ &= \int (h(\boldsymbol{\xi}) \mathbf{g}(\mathbf{x}, t)) \cdot \underbrace{(f \mathbf{n}) d\gamma}_{=0} - \int \nabla_{\boldsymbol{\xi}} \cdot (h(\boldsymbol{\xi}) \mathbf{g}(\mathbf{x}, t)) f d\boldsymbol{\xi} \\ &= -\frac{\rho}{\rho} \int \mathbf{g}(\mathbf{x}, t) \cdot \nabla_{\boldsymbol{\xi}} h(\boldsymbol{\xi}) f d\boldsymbol{\xi} = -\rho I[\mathbf{g}(\mathbf{x}, t) \cdot \nabla_{\boldsymbol{\xi}} h(\boldsymbol{\xi})]. \end{aligned} \quad (2.3.11)$$

Substituting the above three terms into (2.3.8) and noticing that the right hand side vanishes, the following integral-differential equation holds for $h(\boldsymbol{\xi}) = 1, \boldsymbol{\xi}, \boldsymbol{\xi}^2/2$:

$$\partial_t (\rho I[h(\boldsymbol{\xi})]) + \nabla_{\mathbf{x}} \cdot (\rho I[h(\boldsymbol{\xi}) \boldsymbol{\xi}]) - \rho I[\mathbf{g}(\mathbf{x}, t) \cdot \nabla_{\boldsymbol{\xi}} h(\boldsymbol{\xi})] = 0. \quad (2.3.12)$$

In addition, we can easily verify that the following equalities for $h(\xi) = 1, \xi, \xi^2/2$,

$$\left\{ \begin{array}{l} I[1] = 1, I[\xi] = \mathbf{u}, I[\mathbf{c}] = I[\xi] - I[\mathbf{u}] = \mathbf{u} - \mathbf{u} = 0, \\ I[\xi^2] = I[(\mathbf{c} + \mathbf{u})^2] = I[\mathbf{c}^2] - 2I[\mathbf{c}] \cdot \mathbf{u} + \mathbf{u}^2 = 2\epsilon + \mathbf{u}^2, \\ I[\xi\xi] = I[(\mathbf{c} + \mathbf{u})(\mathbf{c} + \mathbf{u})] = I[\mathbf{cc}] - I[\mathbf{c}]\mathbf{u} - \mathbf{u}I[\mathbf{c}] + \mathbf{u}\mathbf{u} = P/\rho + \mathbf{u}\mathbf{u}, \\ I[\xi^2\xi] = I[(\mathbf{c} + \mathbf{u})^2(\mathbf{c} + \mathbf{u})] = I[\mathbf{c}^2\mathbf{c}] + 2I[\mathbf{cc}] \cdot \mathbf{u} + I[\mathbf{c}]\mathbf{u}^2 + I[\xi^2]\mathbf{u} \\ \quad = 2\mathbf{q}/\rho + 2P/\rho \cdot \mathbf{u} + (2\epsilon + \mathbf{u}^2)\mathbf{u}. \end{array} \right. \quad (2.3.13)$$

With (2.3.13) at hand, equation (2.3.12) for $h(\xi) = 1, \xi$ leads to the continuity equation and momentum equation respectively ($\nabla := \nabla_{\mathbf{x}}$ for simplicity):

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (2.3.14)$$

$$\partial_t(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla \cdot P = \rho \mathbf{g}. \quad (2.3.15)$$

With the constraint of the continuity equation (2.3.14), the momentum equation (2.3.15) can be simplified in the Eulerian frame as

$$\rho \partial_t \mathbf{u} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla \cdot P = \rho \mathbf{g}. \quad (2.3.16)$$

Equations (2.3.14) and (2.3.15) or (2.3.16) define the full compressible Navier-Stokes equations. Notice that stress tensor P is unknown. When $h(\xi) = \xi^2/2$, the energy equation is obtained from (2.3.12) as follows

$$\begin{aligned} 0 &= \frac{1}{2} \partial_t(\rho I[\xi^2]) + \frac{1}{2} \nabla_{\mathbf{x}} \cdot (\rho I[\xi^2 \xi]) - \frac{1}{2} \rho I[\mathbf{g}(\mathbf{x}, t) \cdot \nabla_{\xi} \xi^2] \\ (\text{by } 2.3.13) &= \partial_t(\rho \epsilon + \frac{1}{2} \rho \mathbf{u}^2) + \nabla \cdot ((\rho \epsilon + \frac{1}{2} \rho \mathbf{u}^2) \mathbf{u} + P \cdot \mathbf{u} + \mathbf{q}) - \rho \mathbf{g} \cdot \mathbf{u} \\ (\text{reorder}) &= \underbrace{(\partial_t(\rho \epsilon) + \nabla \cdot (\rho \mathbf{u} \epsilon))}_{\text{Term 1}} + \underbrace{\left(\frac{1}{2} \partial_t(\rho \mathbf{u}^2) + \frac{1}{2} \nabla \cdot ((\rho \mathbf{u}^2) \mathbf{u}) + P \cdot \mathbf{u} \right)}_{\text{Term 2}} \\ &\quad + \nabla \cdot \mathbf{q} - \rho \mathbf{g} \cdot \mathbf{u}. \end{aligned} \quad (2.3.17)$$

$$\begin{aligned} \text{Term 1} &= (\partial_t \rho) \epsilon + \rho \partial_t \epsilon + (\nabla \cdot (\rho \mathbf{u})) \epsilon + \rho \mathbf{u} \cdot (\nabla \epsilon) \\ (\text{by } 2.3.14) &= -(\nabla \cdot (\rho \mathbf{u})) \epsilon + (\nabla \cdot (\rho \mathbf{u})) \epsilon + (\rho \partial_t \epsilon + \rho \mathbf{u} \cdot (\nabla \epsilon)) \\ &= \rho (\partial_t + (\nabla \epsilon) \cdot \mathbf{u}) = \rho \frac{d\epsilon}{dt}. \end{aligned} \quad (2.3.18)$$

$$\begin{aligned}
\text{Term 2} &= \frac{1}{2} \left((\partial_t \rho) \mathbf{u}^2 + \rho (\partial_t \mathbf{u}^2) + (\nabla \cdot (\rho \mathbf{u})) \mathbf{u}^2 + \rho \mathbf{u} \cdot (\nabla \mathbf{u}^2) \right) \\
&\quad + (\mathbf{u} \cdot (\nabla \cdot P) + P : (\nabla \mathbf{u})) \quad \left(\text{where } P : (\nabla \mathbf{u}) = \sum_{i,j} P_{i,j} (\nabla \mathbf{u})_{i,j} \right) \\
(\text{by 2.3.14}) &= \frac{1}{2} \left(-(\nabla \cdot (\rho \mathbf{u})) \mathbf{u}^2 + (\nabla \cdot (\rho \mathbf{u})) \mathbf{u}^2 + 2\rho \mathbf{u} \cdot (\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u}) \right) \\
&\quad + \mathbf{u} \cdot (\nabla \cdot P) + P : (\nabla \mathbf{u}) \\
(\text{by 2.3.16}) &= P : (\nabla \mathbf{u}) + \rho \mathbf{g} \cdot \mathbf{u}.
\end{aligned} \tag{2.3.19}$$

Substituting Term 2.3.18 and Term 2.3.19 into equation 2.3.17, we get the energy equation in the Lagrangian formulation as

$$\rho \frac{d\epsilon}{dt} + P : (\nabla \mathbf{u}) + \nabla \cdot \mathbf{q} = 0. \tag{2.3.20}$$

Briefly, equations 2.3.14, 2.3.15 and 2.3.20 are the hydrodynamic equations for a fluid, which take both thermal and compressible effects into account. From the definition of stress tensor and kinetic energy density we have the following relation

$$P_{ii} = 2\rho\epsilon, \tag{2.3.21}$$

while the average of the trace of the stress tensor is borrowed to define the hydrostatic pressure as

$$p = \frac{P_{ii}}{D}. \tag{2.3.22}$$

Meanwhile, by the perfect gas law, the kinetic energy density is related to temperature according to the equipartition theorem as

$$p = \rho\theta = \rho c_s^2. \tag{2.3.23}$$

Therefore, from macroscopic point of view, there are 13 moments in the flow system. Namely one moment for the mass density ρ , three for the velocity \mathbf{u} , six for the stress tensor P and three for the energy flux \mathbf{q} . Correspondingly, 13 equations are needed to close this system. However, we have only five equations in total. In detail, there is one equation in 2.3.14, three in 2.3.15 and one in 2.3.20. Hence, the problem becomes how to close the system with 13 unknowns by 5 equations. More specifically, the problem focuses on how to determine the stress tensor P and the energy flux \mathbf{q} . The first approach goes to the Grad's representation approach in the next section.

2.4 Grad's representation approach

In this section, we will first introduce Hermite Polynomials and some of their basic properties. Then the distribution function will be projected on the Hermite bases. After a brief investigation of the projected Boltzmann - BGK equation, we will derive the Euler equations and Navier-Stokes equations and then explain why we choose certain order of expansion and truncation of distribution function to get Euler equations, Navier-Stokes equations and beyond. Lastly, the continuous Boltzmann - BGK equation will be discretized to the lattice Boltzmann - BGK model.

2.4.1 Hermite polynomials

As Shan et al. claimed in [29], “A unique feature in using the Hermite polynomials as the expansion basis rather than any other functions is that the expansion coefficients correspond precisely to the velocity moments up to the given degrees.” That’s the reason why we choose Hermite polynomials as the basis for projection of Boltzmann - BGK equations.

In a D-dimensional space, the Hermite polynomial of degree n is defined by the n^{th} derivative of the weight function, which is given as

$$w(\boldsymbol{\xi}) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{\boldsymbol{\xi}^2}{2}\right). \quad (2.4.1)$$

Therefore, we obtain the Hermite polynomial of degree n as

$$\mathcal{H}_{\alpha}^{(n)}(\boldsymbol{\xi}) = \frac{(-1)^n}{w(\boldsymbol{\xi})} \partial_{\xi_{\alpha}}^n w(\boldsymbol{\xi}) = \frac{(-1)^n}{w(\boldsymbol{\xi})} \frac{\partial^n w(\boldsymbol{\xi})}{\partial \xi_{\alpha_1} \partial \xi_{\alpha_2} \cdots \partial \xi_{\alpha_n}}. \quad (2.4.2)$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ and $\alpha_i = 1, 2, \dots, D; i = 1, 2, \dots, n$. In the tensorial form, the n^{th} rank symmetric tensor of n^{th} degree polynomial is denoted as

$$\mathcal{H}^{(n)}(\boldsymbol{\xi}) = \frac{(-1)^n}{w(\boldsymbol{\xi})} \nabla^n w(\boldsymbol{\xi}) \quad (2.4.3)$$

Hence, the first few polynomials are given explicitly as

$$\left\{ \begin{array}{l} \mathcal{H}^{(0)}(\boldsymbol{\xi}) = \frac{(-1)^0}{w(\boldsymbol{\xi})} w(\boldsymbol{\xi}) = 1, \\ \mathcal{H}_{\alpha}^{(1)}(\boldsymbol{\xi}) = \frac{(-1)}{w(\boldsymbol{\xi})} \partial_{\xi_{\alpha}} w(\boldsymbol{\xi}) = \xi_{\alpha}, \\ \mathcal{H}_{\alpha_1 \alpha_2}^{(2)}(\boldsymbol{\xi}) = \frac{(-1)^2}{w(\boldsymbol{\xi})} \partial_{\xi_{\alpha_1} \xi_{\alpha_2}}^2 w(\boldsymbol{\xi}) = \xi_{\alpha_1} \xi_{\alpha_2} - \delta_{\alpha_1 \alpha_2}, \\ \mathcal{H}_{\alpha_1 \alpha_2 \alpha_3}^{(3)}(\boldsymbol{\xi}) = \frac{(-1)^3}{w(\boldsymbol{\xi})} \partial_{\xi_{\alpha_1} \xi_{\alpha_2} \xi_{\alpha_3}}^3 w(\boldsymbol{\xi}) = \xi_{\alpha_1} \xi_{\alpha_2} \xi_{\alpha_3} - (\delta_{\alpha_1 \alpha_2} \xi_{\alpha_3} + \delta_{\alpha_1 \alpha_3} \xi_{\alpha_2} + \delta_{\alpha_2 \alpha_3} \xi_{\alpha_1}), \\ \mathcal{H}_{\alpha_1 \alpha_2 \alpha_3 \alpha_4}^{(4)}(\boldsymbol{\xi}) = \frac{(-1)^4}{w(\boldsymbol{\xi})} \partial_{\xi_{\alpha_1} \xi_{\alpha_2} \xi_{\alpha_3} \xi_{\alpha_4}}^4 w(\boldsymbol{\xi}) = \xi_{\alpha_1} \xi_{\alpha_2} \xi_{\alpha_3} \xi_{\alpha_4} - (\delta_{\alpha_1 \alpha_2} \xi_{\alpha_3} \xi_{\alpha_4} + \delta_{\alpha_1 \alpha_3} \xi_{\alpha_2} \xi_{\alpha_4} \\ \quad + \delta_{\alpha_1 \alpha_4} \xi_{\alpha_2} \xi_{\alpha_3} + \delta_{\alpha_2 \alpha_3} \xi_{\alpha_1} \xi_{\alpha_4} + \delta_{\alpha_2 \alpha_4} \xi_{\alpha_1} \xi_{\alpha_3} + \delta_{\alpha_3 \alpha_4} \xi_{\alpha_1} \xi_{\alpha_2}) \\ \quad + (\delta_{\alpha_1 \alpha_2} \delta_{\alpha_3 \alpha_4} + \delta_{\alpha_1 \alpha_3} \delta_{\alpha_2 \alpha_4} + \delta_{\alpha_1 \alpha_4} \delta_{\alpha_2 \alpha_3}), \end{array} \right. \quad (2.4.4)$$

where $\delta_{\alpha_1 \alpha_2}$ is the Kronecker delta function with value 1 if $\alpha_1 = \alpha_2$ and 0 otherwise. Denoting by \tilde{n} the cyclic index permutation with n different forms, we have the general n th rank symmetric tensor given in a simple way as (I is identity matrix)

$$\left\{ \begin{array}{l} \mathcal{H}^{(0)}(\boldsymbol{\xi}) = 1, \\ \mathcal{H}^{(1)}(\boldsymbol{\xi}) = \boldsymbol{\xi}, \\ \mathcal{H}^{(2)}(\boldsymbol{\xi}) = \boldsymbol{\xi} \boldsymbol{\xi} - I, \\ \mathcal{H}^{(3)}(\boldsymbol{\xi}) = \boldsymbol{\xi} \boldsymbol{\xi} \boldsymbol{\xi} - \tilde{3} \boldsymbol{\xi} I, \\ \mathcal{H}^{(4)}(\boldsymbol{\xi}) = \boldsymbol{\xi} \boldsymbol{\xi} \boldsymbol{\xi} \boldsymbol{\xi} - \tilde{6} \boldsymbol{\xi} \boldsymbol{\xi} I + \tilde{3} II. \end{array} \right. \quad (2.4.5)$$

A useful recurrence relation is deduced as follows. By the definition of Hermite polynomial (2.4.3), we have

$$\mathcal{H}^{(n)} w(\boldsymbol{\xi}) = (-1)^n \nabla_{\boldsymbol{\xi}}^n w(\boldsymbol{\xi}),$$

so that

$$\begin{aligned} -\mathcal{H}^{(n+1)} w(\boldsymbol{\xi}) &= -(-1)^{n+1} \nabla_{\boldsymbol{\xi}}^{n+1} w(\boldsymbol{\xi}) = \nabla_{\boldsymbol{\xi}} (\mathcal{H}^{(n)} w(\boldsymbol{\xi})) \\ &= \mathcal{H}^{(n)} (\nabla_{\boldsymbol{\xi}} w(\boldsymbol{\xi})) + \nabla_{\boldsymbol{\xi}} (\mathcal{H}^{(n)}) w(\boldsymbol{\xi}) \quad (2.4.6) \\ &= -\boldsymbol{\xi} \mathcal{H}^{(n)} w(\boldsymbol{\xi}) + \nabla_{\boldsymbol{\xi}} (\mathcal{H}^{(n)}) w(\boldsymbol{\xi}). \end{aligned}$$

Therefore,

$$\boldsymbol{\xi} \mathcal{H}^{(n)} = \mathcal{H}^{(n+1)} + \nabla_{\boldsymbol{\xi}} (\mathcal{H}^{(n)}). \quad (2.4.7)$$

More explicitly, we have

$$\begin{aligned}\xi_{\alpha_0} \mathcal{H}_{\alpha_1, \alpha_2, \dots, \alpha_n}^{(n)} &= \mathcal{H}_{\alpha_0, \alpha_1, \dots, \alpha_n}^{(n+1)} + \partial_{\xi_{\alpha_0}} \mathcal{H}_{\alpha_1, \alpha_2, \dots, \alpha_n}^{(n)} \\ &= \mathcal{H}_{\alpha_0, \alpha_1, \dots, \alpha_n}^{(n+1)} + \sum_{i=1}^n \delta_{\alpha_0, \alpha_i} \mathcal{H}_{\alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n}^{(n-1)}.\end{aligned}\quad (2.4.8)$$

For the derivative of Hermite polynomials with respect to velocity in the compact form, we have

$$\nabla_{\boldsymbol{\xi}}(\mathcal{H}^{(n)}) = \tilde{n} \mathcal{H}^{(n-1)} I \quad (2.4.9)$$

Moreover, Hermite polynomials form a set of orthonormal bases of the Hilbert space equipped with the inner product $\langle f, g \rangle = \int w f g d\boldsymbol{\xi}$ and the following relation holds as

$$\int w(\boldsymbol{\xi}) \mathcal{H}_{\boldsymbol{\alpha}}^{(m)} \mathcal{H}_{\boldsymbol{\beta}}^{(n)} d\boldsymbol{\xi} = \delta_{mn} \boldsymbol{\delta}_{\boldsymbol{\alpha}\boldsymbol{\beta}}, \quad (2.4.10)$$

where $\boldsymbol{\delta}_{\boldsymbol{\alpha}\boldsymbol{\beta}}^n$ is 1 if $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$ is one permutation of $\boldsymbol{\beta} = (\beta_1, \dots, \beta_n)$, vanishing otherwise. With the first few Hermite polynomials, we can project the distribution function on Hermite bases explicitly in the next section.

2.4.2 Projection on Hermite bases

The distribution function projected on Hermite bases reads

$$f(\mathbf{x}, \boldsymbol{\xi}, t) = w(\boldsymbol{\xi}) \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{a}^{(n)}(\mathbf{x}, t) \mathcal{H}^{(n)}(\boldsymbol{\xi}), \quad (2.4.11)$$

and the truncated projection up to the N^{th} order ($N = 1, 2, \dots$) reads

$$f(\mathbf{x}, \boldsymbol{\xi}, t) = w(\boldsymbol{\xi}) \sum_{n=0}^N \frac{1}{n!} \mathbf{a}^{(n)}(\mathbf{x}, t) \mathcal{H}^{(n)}(\boldsymbol{\xi}), \quad (2.4.12)$$

where the coefficient $\mathbf{a}^{(n)}$ is defined by

$$\mathbf{a}^{(n)}(\mathbf{x}, t) = \int f(\mathbf{x}, \boldsymbol{\xi}, t) \mathcal{H}^{(n)}(\boldsymbol{\xi}) d\boldsymbol{\xi}. \quad (2.4.13)$$

Both $\mathbf{a}^{(n)}$ and $\mathcal{H}^{(n)}$ are n^{th} -rank tensors and the product between them stands for full contraction. In order to express the projected distribution truncated up to the n^{th} moment explicitly, we need to know the specific expression of the coefficients up to the n^{th} degree. From the definition of Hermite polynomials (2.4.5), the first five coefficients

are given as

$$\left\{ \begin{array}{l} \mathbf{a}^{(0)} = \rho, \\ \mathbf{a}^{(1)} = \rho \mathbf{u}, \\ \mathbf{a}^{(2)} = P + \rho(\mathbf{u}\mathbf{u} - I), \\ \mathbf{a}^{(3)} = K + \tilde{3}a^{(2)} + (1 - D)\rho\mathbf{u}\mathbf{u}\mathbf{u}, \\ \mathbf{a}^{(4)} = L + \tilde{4}\mathbf{u}K + \tilde{6}\mathbf{u}\mathbf{u}P - \tilde{6}PI - \tilde{6}\rho\mathbf{u}\mathbf{u}I + \tilde{3}\rho II, \end{array} \right. \quad (2.4.14)$$

where

$$K = \int cccf(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi} \text{ and } L = \int cccc f(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi}$$

are the third and fourth moments. In a similar way for the equilibrium distribution (2.2.9), we also get the corresponding coefficients as

$$\left\{ \begin{array}{l} \mathbf{a}_0^{(0)} = \rho, \\ \mathbf{a}_0^{(1)} = \rho \mathbf{u}, \\ \mathbf{a}_0^{(2)} = \rho \mathbf{u}\mathbf{u} + \rho(\theta - 1)I, \\ \mathbf{a}_0^{(3)} = \rho \mathbf{u}\mathbf{u}\mathbf{u} + \tilde{3}\rho(\theta - 1)\mathbf{u}I, \\ \mathbf{a}_0^{(4)} = \rho \mathbf{u}\mathbf{u}\mathbf{u}\mathbf{u} + \tilde{3}\rho(\theta - 1)^2II + \tilde{6}\rho(\theta - 1)\mathbf{u}\mathbf{u}I. \end{array} \right. \quad (2.4.15)$$

With all these coefficients expressed by different moments of velocity up to the 4th order, it's time to project the Boltzmann-BGK equation on the Hermite bases. Considering the Boltzmann-BGK equation

$$(\partial_t + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} + \mathbf{g} \cdot \nabla_{\boldsymbol{\xi}})f(\mathbf{x}, \boldsymbol{\xi}, t) = -\frac{1}{\tau}(f - f^{(0)}), \quad (2.4.16)$$

the projection of the left hand side of equation (2.4.16) on Hermite polynomials yields

$$\begin{aligned} & \int (\partial_t + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} + \mathbf{g} \cdot \nabla_{\boldsymbol{\xi}})f(\mathbf{x}, \boldsymbol{\xi}, t) \mathcal{H}^{(n)}(\boldsymbol{\xi}) d\boldsymbol{\xi} \\ &= \partial_t \left(\int \underbrace{f \mathcal{H}^{(n)}(\boldsymbol{\xi}) d\boldsymbol{\xi}}_{\text{by (2.4.13)}} \right) + \int \nabla_{\mathbf{x}} \cdot \left(\underbrace{\boldsymbol{\xi} \mathcal{H}^{(n)}(\boldsymbol{\xi}) f}_{\text{by 2.4.7}} \right) d\boldsymbol{\xi} + \int \underbrace{(\nabla_{\boldsymbol{\xi}} \cdot (\mathbf{g}(\mathbf{x}, t)f)) \mathcal{H}^{(n)}(\boldsymbol{\xi})}_{\text{by Green's formula}} d\boldsymbol{\xi} \\ &= \partial_t (\mathbf{a}^{(n)}) + \nabla_{\mathbf{x}} \cdot \left(\int (\mathcal{H}^{(n+1)}(\boldsymbol{\xi})f + \tilde{n}\mathcal{H}^{(n-1)}(\boldsymbol{\xi})If) d\boldsymbol{\xi} \right) \\ & \quad + \int (\mathbf{g}(\mathbf{x}, t) \underbrace{f}_{=0 \text{ on } \partial R^D}) \mathcal{H}^{(n)}(\boldsymbol{\xi}) \cdot \mathbf{n} d\gamma - \int (\mathbf{g}(\mathbf{x}, t)f) \cdot \underbrace{\nabla_{\boldsymbol{\xi}} \mathcal{H}^{(n)}(\boldsymbol{\xi})}_{\text{by 2.4.9}} d\boldsymbol{\xi} \\ &= \partial_t (\mathbf{a}^{(n)}) + \nabla_{\mathbf{x}} \cdot (\mathbf{a}^{(n+1)}) + \tilde{n} \nabla_{\mathbf{x}} \cdot (\mathbf{a}^{(n-1)}I) - \tilde{n} \mathbf{g} \cdot (\mathbf{a}^{(n-1)}I), \end{aligned}$$

while the left hand side of the Boltzmann-BGK equation (2.4.16) becomes

$$\int -\frac{1}{\tau} (f - f^{(0)}) \mathcal{H}^{(n)}(\boldsymbol{\xi}) d\boldsymbol{\xi} = -\frac{1}{\tau} (\mathbf{a}^{(n)} - \mathbf{a}_0^{(n)}).$$

Therefore, the projected Boltzmann-BGK equation reads as

$$\partial_t (\mathbf{a}^{(n)}) + \nabla_{\mathbf{x}} \cdot (\mathbf{a}^{(n+1)}) + \tilde{n} \nabla_{\mathbf{x}} \cdot (\mathbf{a}^{(n-1)} I) - \tilde{n} \mathbf{g} \cdot (\mathbf{a}^{(n-1)} I) = -\frac{1}{\tau} (\mathbf{a}^{(n)} - \mathbf{a}_0^{(n)}). \quad (2.4.17)$$

Remark 2.4.1 Equation (2.4.17) implies that the Boltzmann equation is potentially related to the macroscopic equation through the procedure of Hermite projection, since the coefficients are nothing else but certain moments of velocity-the macroscopic variables.

2.4.3 Euler equations

In the equilibrium state where distribution function are given as Maxwell-Boltzmann distribution, the kinetic Boltzmann equation leads to the macroscopic Euler equation, as shown by setting all the distribution function to be that in (2.2.9). Going through the same procedure as for that results in (2.4.17), we have the equilibrium projected Boltzmann-BGK equation given as

$$\partial_t (\mathbf{a}_0^{(n)}) + \nabla_{\mathbf{x}} \cdot (\mathbf{a}_0^{(n+1)}) + \tilde{n} \nabla_{\mathbf{x}} \cdot (\mathbf{a}_0^{(n-1)} I) - \tilde{n} \mathbf{g} \cdot (\mathbf{a}_0^{(n-1)} I) = -\frac{1}{\tau} (\mathbf{a}_0^{(n)} - \mathbf{a}_0^{(n)}) \quad (2.4.18)$$

With the coefficients corresponding to the equilibrium distribution given by (2.4.15), the explicit equations for $n = 0, 1, 2$ in (2.4.18) are obtained as follows

$$\begin{aligned} n = 0 & \quad \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \\ n = 1 & \quad \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \mathbf{u} + \rho(\theta - 1) I) + \nabla \cdot (\rho I) - \rho \mathbf{g} = 0 \\ \Rightarrow & \quad \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) + \nabla \cdot (\rho \theta I) = \rho \mathbf{g}, \\ n = 2 & \quad \underbrace{\partial_t (\rho \mathbf{u} \mathbf{u} + \rho(\theta - 1) I)}_{(Term1)} + \underbrace{\nabla \cdot (\rho \mathbf{u} \mathbf{u} \mathbf{u} + \tilde{3}\rho(\theta - 1) \mathbf{u} I)}_{(Term2)} \\ & \quad + \underbrace{2\nabla \cdot (\rho \mathbf{u} I)}_{(Term3)} - \underbrace{2\mathbf{g} \cdot (\rho \mathbf{u} I)}_{(Term4)} = 0, \end{aligned} \quad (2.4.19)$$

where we assume $n = 0$, $\mathbf{a}_0^{(-1)} = 0$. For $n = 2$, taking trace ($Tr(\cdot)$), we have

$$\begin{aligned} \text{Tr(Term1)} &= \partial_t (\rho \mathbf{u}^2 + \rho D(\theta - 1)) \\ &= \underbrace{\partial_t (\rho \theta D)}_{\text{by } 2.3.21|2.3.22|2.3.23} - \underbrace{\partial_t \rho D + (\partial_t \rho) \mathbf{u}^2}_{\text{continuity equation}} + \underbrace{2\rho \mathbf{u} \cdot \partial_t \mathbf{u}}_{\text{momentum equation}} \end{aligned}$$

$$\begin{aligned}
&= 2\partial_t(\rho\epsilon) - \nabla \cdot (\rho\mathbf{u})(\mathbf{u}^2 - D) - 2\rho\mathbf{u} \cdot ((\mathbf{u} \cdot \nabla)\mathbf{u}) - 2\mathbf{u} \cdot \nabla(\rho\theta) + 2\rho\mathbf{g} \cdot \mathbf{u}, \\
\text{Tr}(\text{Term2+Term3}) &= \nabla \cdot (\rho\mathbf{u})\text{Tr}(\mathbf{u}\mathbf{u}) + 2(\nabla \cdot \mathbf{u})\text{Tr}(\rho\mathbf{u}\mathbf{u}) + (D+2)\nabla \cdot (\rho\theta\mathbf{u}) - D\nabla \cdot (\rho\mathbf{u}) \\
&= \nabla \cdot (\rho\mathbf{u})(\mathbf{u}^2 - D) + 2(\rho\mathbf{u} \cdot \mathbf{u})(\nabla \cdot \mathbf{u}) + (D+2)\nabla \cdot (\rho\theta\mathbf{u}), \\
\text{Tr}(\text{Term4}) &= 2\rho\mathbf{g} \cdot \mathbf{u},
\end{aligned}$$

$$\frac{1}{2}\text{Tr}(\text{Term1+Term2+Term3+Term4}) = \partial_t(\rho\epsilon) - \mathbf{u} \cdot \nabla(\rho\theta) + \frac{D+2}{2}\nabla \cdot (\rho\theta\mathbf{u}) = 0.$$

By noticing that $\rho\theta = (2/D)\rho\epsilon$, from the equilibrium distribution of Boltzmann-BGK equation, we can get the full Euler equations as

$$\left\{
\begin{array}{l}
\partial_t\rho + \nabla \cdot (\rho\mathbf{u}) = 0, \\
\partial_t(\rho\mathbf{u}) + \nabla \cdot (\rho\mathbf{u}\mathbf{u}) + \frac{2}{D}\nabla \cdot (\rho\epsilon I) = \rho\mathbf{g}, \\
\partial_t(\rho\epsilon) - \frac{2}{D}\mathbf{u} \cdot \nabla(\rho\epsilon) + \frac{D+2}{D}\nabla \cdot (\rho\epsilon\mathbf{u}) = 0.
\end{array}
\right. \quad (2.4.20)$$

Euler equations were born for the fluids in ideal state without viscosity or heat conduction [35]. However, most of the fluids are far away from the ideal state, which corresponds, from the kinetic point of view, to the non-equilibrium state for Boltzmann equation. In the following section, we will see that in order to derive the full Navier-Stokes equations, this non-equilibrium state has to be taken into account.

2.4.4 Navier-Stokes equations

In order to derive the Navier-Stokes equations that are used to describe more general fluids, let's move a step further to the non-equilibrium distribution function. From the kinetic point of view, the distribution function can be separated into two parts: the equilibrium part and the non-equilibrium part. The physical phenomenon that molecules in the fluid system tend to move to the equilibrium state is described by the collision operator. While local oscillations or Brownian motion of molecules arising from the random collision under the molecular force such as van der Waals force, as well as other motion diverging from the equilibrium state can be collectively described by the non-equilibrium distribution. From the macroscopic point of view, all these diverging motions appears in the form of either the temporal or spatial derivative of the variables as density, momentum and so on. Formally, the full distribution function can be decomposed into two parts as

$$f(\mathbf{x}, \boldsymbol{\xi}, t) = f^{(eq)}(\mathbf{x}, \boldsymbol{\xi}, t) + f^{(neq)}(\mathbf{x}, \boldsymbol{\xi}, t), \quad (2.4.21)$$

where $f^{(eq)}$ is the equilibrium part given by (2.2.9) and $f^{(neq)}$ is the non-equilibrium(or off-equilibrium) part which requires more attention. Asymptotically speaking, the non-equilibrium distribution $f^{(neq)}$ can be regarded as small perturbation from equilibrium state, with the scale of Knudsen number $O(\varepsilon)$, which is defined as the ratio of the length l of mean free path of molecular to a characteristic physical length scale L , i.e.

$$\varepsilon = l/L. \quad (2.4.22)$$

If the Knudsen number is close to or larger than one, the mean free path of a molecule is comparable to the length scale of the physical problem, and the continuum assumption of fluid mechanics is not a reasonable approximation. Since the continuum assumption shoud be hold under our consideration, the Knudsen number is supposed to be much small than 1 ($\varepsilon \ll 1$). In order to clarity the role of the non-equilibrium distribution function in the Boltzmann - BGK equation, let's rewrite (2.4.16) as

$$(\partial_t + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} + \mathbf{g} \cdot \nabla_{\boldsymbol{\xi}})(f^{(eq)}(\mathbf{x}, \boldsymbol{\xi}, t) + f^{(neq)}(\mathbf{x}, \boldsymbol{\xi}, t)) = -\frac{1}{\tau} f^{(neq)}(\mathbf{x}, \boldsymbol{\xi}, t). \quad (2.4.23)$$

Under the assumption that $\varepsilon \ll 1$, the above Boltzmann-BGK equation (2.4.23) is approximated by neglecting the non-equilibrium distribution on the left hand side as

$$(\partial_t + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} + \mathbf{g} \cdot \nabla_{\boldsymbol{\xi}})f^{(eq)}(\mathbf{x}, \boldsymbol{\xi}, t) \approx -\frac{1}{\tau} f^{(neq)}(\mathbf{x}, \boldsymbol{\xi}, t). \quad (2.4.24)$$

Projecting (2.4.24) on Hermite bases, we obtain the coefficients of the n^{th} moment of velocity as in (2.4.17)

$$-\frac{1}{\tau} \mathbf{a}_1^{(n)} \approx \partial_t (\mathbf{a}_0^{(n)}) + \nabla_{\mathbf{x}} \cdot (\mathbf{a}_0^{(n+1)}) + \tilde{n} \nabla_{\mathbf{x}} \cdot (\mathbf{a}_0^{(n-1)} I) - \tilde{n} \mathbf{g} \cdot (\mathbf{a}_0^{(n-1)} I), \quad (2.4.25)$$

where we have

$$\begin{cases} \mathbf{a}_1^{(n)}(\mathbf{x}, t) = \int f^{(neq)}(\mathbf{x}, \boldsymbol{\xi}, t) \mathcal{H}^{(n)}(\boldsymbol{\xi}) d\boldsymbol{\xi}, \\ f^{(neq)}(\mathbf{x}, \boldsymbol{\xi}, t) = \sum_{n=0}^{\infty} \frac{1}{n!} \mathcal{H}^{(n)}(\boldsymbol{\xi}) \mathbf{a}_1^{(n)}(\mathbf{x}, t). \end{cases} \quad (2.4.26)$$

In order to get $f^{(neq)}$ from (2.4.26), the non-equilibrium coefficients $\mathbf{a}_1^{(n)}$ is needed by (2.4.25). The first five tensors (2.4.15) imply that the coefficients for equilibrium distribution up to fourth order depend on density ρ , momentum $\mathbf{j} := \rho \mathbf{u}$ and energy $e := \rho \epsilon$, but not explicitly on time. Therefore, the last three terms on the right hand side of (2.4.25) are determined. For the first term involving temporal derivative we apply the chain rule as

$$\partial_t (\mathbf{a}_0^{(n)}) = \partial_{\rho} (\mathbf{a}_0^{(n)}) \partial_t \rho + \partial_{\mathbf{j}} (\mathbf{a}_0^{(n)}) \cdot \partial_t (\rho \mathbf{u}) + \partial_e (\mathbf{a}_0^{(n)}) \partial_t (\rho \epsilon). \quad (2.4.27)$$

which can be rewritten by employing Euler equations (2.4.20) as

$$\begin{aligned} \partial_t (\mathbf{a}_0^{(n)}) &= -\partial_\rho (\mathbf{a}_0^{(n)}) \nabla \cdot (\rho \mathbf{u}) + \partial_j (\mathbf{a}_0^{(n)}) \cdot \left(\rho \mathbf{g} - \nabla \cdot (\rho \mathbf{u} \mathbf{u}) - \frac{2}{D} \nabla \cdot (\rho \epsilon I) \right) \\ &\quad + \partial_e (\mathbf{a}_0^{(n)}) \left(\frac{D+1}{2} \rho \mathbf{g} \cdot \mathbf{u} + \frac{2}{D} \mathbf{u} \cdot \nabla (\rho \epsilon) - \frac{D+2}{D} \nabla \cdot (\rho \epsilon \mathbf{u}) \right). \end{aligned} \quad (2.4.28)$$

where the terms of $\partial_\rho, \partial_j, \partial_e$ remain to be computed. Firstly, replacing the velocity \mathbf{u} by \mathbf{j}/ρ , energy $\rho\epsilon$ by e in (2.4.15), we have

$$\left\{ \begin{array}{l} \mathbf{a}_0^{(0)} = \rho \\ \mathbf{a}_0^{(1)} = \mathbf{j} \\ \mathbf{a}_0^{(2)} = \mathbf{j}\mathbf{j}/\rho + (2/D)eI - \rho I \\ \mathbf{a}_0^{(3)} = \mathbf{j}\mathbf{j}\mathbf{j}/\rho^2 + \tilde{3}((2/D)e\mathbf{j}/\rho - \mathbf{j})I. \end{array} \right. \quad (2.4.29)$$

Taking derivative for all the coefficient tensors in (2.4.29) with respect to ρ , \mathbf{j} and e respectively, we have

$$\left\{ \begin{array}{l} \partial_\rho \mathbf{a}_0^{(0)} = 1, \quad \partial_j \mathbf{a}_0^{(0)} = 0, \quad \partial_e \mathbf{a}_0^{(0)} = 0, \\ \partial_\rho \mathbf{a}_0^{(1)} = 0, \quad \partial_j \mathbf{a}_0^{(1)} = 1, \quad \partial_e \mathbf{a}_0^{(1)} = 0, \\ \partial_\rho \mathbf{a}_0^{(2)} = -\mathbf{u}\mathbf{u} - I, \quad \partial_j \mathbf{a}_0^{(2)} = \tilde{2}\mathbf{u}I, \quad \partial_e \mathbf{a}_0^{(2)} = (2/D)I, \\ \partial_\rho \mathbf{a}_0^{(3)} = -2\mathbf{u}\mathbf{u}\mathbf{u} - \tilde{3}\theta\mathbf{u}I, \quad \partial_j \mathbf{a}_0^{(3)} = \tilde{3}\mathbf{u}\mathbf{u}I + \tilde{3}(\theta - 1)II, \quad \partial_e \mathbf{a}_0^{(3)} = \tilde{3}(2/D)\mathbf{u}I. \end{array} \right. \quad (2.4.30)$$

Combining the coefficient tensors for equilibrium distribution (2.4.15), the chain rule for temporal derivative (2.4.28), the partial derivative with respect to density, momentum and energy (2.4.30), we obtain the explicit expression for coefficient tensors of the non-equilibrium distribution (2.4.25) as

$$\left\{ \begin{array}{l} \mathbf{a}_1^{(0)} = 0, \\ \mathbf{a}_1^{(1)} = 0, \\ \mathbf{a}_1^{(2)} = -\tau\rho\theta \left(\tilde{2}\nabla\mathbf{u} - (2/D)(\nabla \cdot \mathbf{u})I \right), \\ \mathbf{a}_1^{(3)} = -\tau\rho\theta \left(\tilde{3}\mathbf{u}(\tilde{2}\nabla\mathbf{u} - (2/D)(\nabla \cdot \mathbf{u})I) + \tilde{3}\nabla\theta I \right). \end{array} \right. \quad (2.4.31)$$

More explicitly we have,

$$\begin{cases} \mathbf{a}_{1\alpha_1\alpha_2}^{(2)} = -\tau\rho\theta\Gamma_{\alpha_1\alpha_2}, & \text{where } \Gamma_{\alpha_1\alpha_2} = \partial_{\alpha_1}u_{\alpha_2} + \partial_{\alpha_2}u_{\alpha_1} - (2/D)\partial_{\alpha_3}u_{\alpha_3}\delta_{\alpha_1\alpha_2} \\ \mathbf{a}_{1\alpha_1\alpha_2\alpha_3}^{(3)} = -\tau\rho\theta(\Gamma_{\alpha_1\alpha_2}u_{\alpha_3} + \Gamma_{\alpha_1\alpha_3}u_{\alpha_2} + \Gamma_{\alpha_2\alpha_3}u_{\alpha_1} + \delta_{\alpha_1\alpha_2}\partial_{\alpha_3}\theta + \delta_{\alpha_1\alpha_3}\partial_{\alpha_2}\theta + \delta_{\alpha_2\alpha_3}\partial_{\alpha_1}\theta). \end{cases} \quad (2.4.32)$$

Truncated up to the third order in (2.4.26)₂, the non-equilibrium distribution becomes

$$f^{(neq)}(\mathbf{x}, \boldsymbol{\xi}, t) = \sum_{n=0}^3 \frac{1}{n!} \mathcal{H}^{(n)}(\boldsymbol{\xi}) \mathbf{a}_1^{(n)}(\mathbf{x}, t). \quad (2.4.33)$$

Currently, the full distribution function can be explicitly approximated by the equilibrium distribution and the non-equilibrium distribution function, so that the stress tensor and energy flux can be obtained as

$$\begin{aligned} P &= \int \mathbf{c} \mathbf{c} f(\mathbf{x}, \boldsymbol{\xi}, t) d\mathbf{c} = \int (\boldsymbol{\xi} - \mathbf{u})(\boldsymbol{\xi} - \mathbf{u})(f^{(eq)}(\mathbf{x}, \boldsymbol{\xi}, t) + f^{(neq)}(\mathbf{x}, \boldsymbol{\xi}, t)) d\boldsymbol{\xi} \\ \text{from (2.4.15)} &= \rho\theta I + \int (\boldsymbol{\xi}\boldsymbol{\xi} - \boldsymbol{\xi}\mathbf{u} - \mathbf{u}\boldsymbol{\xi} + \mathbf{u}\mathbf{u}) f^{(neq)}(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi} \quad (2.4.34) \\ \text{from (2.4.33)} &= \rho\theta I - \tau\rho\theta\Gamma, \end{aligned}$$

and

$$\begin{aligned} \mathbf{q} &= \frac{1}{2} \int \mathbf{c} \mathbf{c}^2 f(\mathbf{x}, \boldsymbol{\xi}, t) d\mathbf{c} = \frac{1}{2} \int (\boldsymbol{\xi} - \mathbf{u})(\boldsymbol{\xi} - \mathbf{u})^2 (f^{(eq)}(\mathbf{x}, \boldsymbol{\xi}, t) + f^{(neq)}(\mathbf{x}, \boldsymbol{\xi}, t)) d\boldsymbol{\xi} \\ \text{from (2.4.15)} &= 0 + \frac{1}{2} \int (\boldsymbol{\xi} - \mathbf{u})(\boldsymbol{\xi} - \mathbf{u})^2 f^{(neq)}(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi} \\ \text{from (2.4.33)} &= -\tau\rho\theta\left(\frac{D+2}{2}\right)\nabla\theta. \quad (2.4.35) \end{aligned}$$

Finally, we have the full Navier-Stokes equations from (2.3.14), (2.3.16), (2.3.20) together with (2.4.34) and (2.4.35):

$$\begin{cases} \partial_t\rho + \nabla \cdot (\rho\mathbf{u}) = 0, \\ \partial_t(\rho\mathbf{u}) + \nabla \cdot (\rho\mathbf{u}\mathbf{u} + pI - \mu_s\Gamma) = \rho\mathbf{g}, \\ \rho\frac{d\epsilon}{dt} + (pI - \mu_s\Gamma) : (\nabla\mathbf{u}) - \frac{D+2}{2}\nabla \cdot (k\nabla\theta) = 0, \end{cases} \quad (2.4.36)$$

Remark 2.4.2 *The dynamic shear viscosity and the thermal diffusivity are the same as $\mu_s = k = \rho\theta\tau$ while the dynamic bulk viscosity is null. These conditions produce the disadvantage of limiting the applicability of (2.4.36) with respect to the general compressible Navier-Stokes equations (where $\nu_s \neq k$ and $\nu' \neq 0$). This is mainly due to the fact that the Boltzmann - BGK modes use a simplified collision term which involves*

only a single relaxation time. One recipe to handle this disadvantage is to employ more relaxation time, for example the multiple relaxation time model [36, 37], where the relaxation time is not the single τ but involve more relaxation time parameters (up to 9 for 9 moments of velocity).

Remark 2.4.3 An alternative approach to hunt for the dynamic bulk viscosity is to modify artificially the forcing term during the Chapman-Enskog ansatz for the Boltzmann-BGK equation. The highlight point for this approach is that different macroscopic equations such as Advection-Diffusion equation and Poisson equation could also be achieved from the Boltzmann-BGK equation [42]. Nevertheless, there is no generic way to design an appropriate modification for the force term. More details will be given in the next section, where the Chapman-Enskog expansion will be used to travel from the Boltzmann-BGK equation to the Navier-Stokes equations.

The computation by the above procedure to derive macroscopic equations from Boltzmann - BGK equation is considerably exhausting but straightforward. In summary, to obtain the coefficient tensors of the non-equilibrium distribution function up to the third order, only the fourth order and the lower orders of coefficient tensors of the equilibrium distribution are required. However, if we are only interested in the density and momentum rather than energy or higher moments, the last equation in (2.4.36) is not needed any more. Hence, only the coefficient tensor of the equilibrium distribution function up to $n = 2$ could play certain role in the projection of the Boltzmann-BGK equation, while the temperature can be assumed to be a constant ($\theta = 1$), without loss of generality.

According to this simplification, the system of equations that we retrieved is

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0 \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \mathbf{u} + pI - \boldsymbol{\tau}) = \rho \mathbf{g} \end{cases} \quad (2.4.37)$$

where $\boldsymbol{\tau} = 2\mu_s S$, $S_{\alpha_1 \alpha_2} = (\partial_{\alpha_1} u_{\alpha_2} + \partial_{\alpha_2} u_{\alpha_1})/2$. Notice that S is different from Γ in (2.4.36), which is defined in (2.4.32). In practice, this simplified Navier-Stokes equations are widely used to describe the evolution of velocity and pressure for fluid dynamics problems.

Note finally that in this simplified Navier-Stokes equations, the second order truncation of the equilibrium distribution function takes the form as

$$\begin{aligned}
f^{(0)}(\mathbf{x}, \boldsymbol{\xi}, t) &= \frac{\rho(\mathbf{x}, t)}{(2\pi)^{D/2}} \exp\left(-\frac{(\boldsymbol{\xi} - \mathbf{u}(\mathbf{x}, t))^2}{2}\right) \\
&= \frac{\rho}{(2\pi)^{D/2}} \exp\left(-\frac{\boldsymbol{\xi}^2}{2}\right) \exp\left(\frac{2\boldsymbol{\xi} \cdot \mathbf{u} - \mathbf{u}^2}{2}\right) \\
&= \rho w(\boldsymbol{\xi}) \left(1 + \boldsymbol{\xi} \cdot \mathbf{u} + \frac{1}{2}(\boldsymbol{\xi} \cdot \mathbf{u})^2 - \frac{1}{2}\mathbf{u}^2\right) + O(\mathbf{u}^3).
\end{aligned} \tag{2.4.38}$$

However, this approximation only holds when the macroscopic velocity \mathbf{u} is small compared to the sound velocity c_s . In another way, the ratio of macroscopic velocity and sound velocity which is defined as the Mach number Ma should be smaller than unity. More physically, we have the rescaled sound velocity c_s as (see [38])

$$c_s = \sqrt{\frac{\gamma k_B T}{m}} / \sqrt{\frac{k_B T}{m}} = \sqrt{\gamma} \text{ and } Ma = \frac{\sqrt{\mathbf{u}^2}}{c_s} < 1, \tag{2.4.39}$$

where γ is the specific heat ratio related to the number of degrees of freedom of gas molecules β as $\gamma = (\beta + 2)/\beta \rightarrow 1$ as $\beta \rightarrow \infty$, when the sound velocity becomes $c_s = 1$.

It's worth to mention that with equilibrium distribution function truncated up to third order, we can achieve the Navier-Stokes equations in (2.4.36) accurate up to the momentum equations with Γ in (2.4.32) instead of S for the simplified Navier-Stokes equations. Notice that comparing (2.4.37) to the standard incompressible Navier-Stokes equations, we have that the incompressibility constraint $\nabla \cdot \mathbf{u} = 0$ is replaced by the continuity equation (2.4.37)₁. In this sense, the Boltzmann-BGK equation leads to weak compressible Navier-Stokes equations, which should always be bear in mind when applying the kinetic results to explain the macroscopic equations.

Remark 2.4.4 *To this point, with Euler equations and Navier Stokes equations as examples, the reader might have two questions. The first one could be “if there is a general rule for up to which level of expansion of distribution function we need to capture the right macroscopic equations.” ($f \approx f^{(eq)} = \varepsilon^0 f^{(0)}$ for Euler equations, $f = f^{(eq)} + f^{(neq)} \approx \varepsilon^0 f^{(0)} + \varepsilon^1 f^{(1)}$ for Navier-Stokes equations and so on). While the second one would go to “if there is a general rule for up to which order of truncation of distribution function we need in order not to lose accuracy for different moments such as density, momentum, energy and so on.” (third order truncation of equilibrium part and second order truncation of non-equilibrium part of distribution function for Navier-Stokes equations with accurate density, momentum and energy). Fortunately, the answers for both questions are positive, as proved in the next section.*

2.4.5 Beyond Navier-Stokes equations

In general, by expanding the distribution function on different powers of Knudsen number as

$$f = f^{(0)} + \varepsilon f^{(1)} + \varepsilon^2 f^{(2)} + \dots, \quad (2.4.40)$$

while expanding temporal derivatives on differential powers of Knudsen number corresponding to relaxation time on different scales (one order higher than expansion for distribution function) as

$$\partial_t = \varepsilon \partial_t^{(1)} + \varepsilon^2 \partial_t^{(2)} + \dots. \quad (2.4.41)$$

Together with one order expansion for spatial derivative as $\nabla_{\mathbf{x}} = \varepsilon \nabla_{\mathbf{x}}^{(1)}$ ¹, we have by (2.4.16)

$$f^{(k+1)} = -\tau \left(\sum_{m=0}^k \partial_t^{(k-m+1)} f^{(m)} + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}}^{(1)} f^{(k)} + \mathbf{g} \cdot \nabla_{\boldsymbol{\xi}} f^{(k)} \right). \quad (2.4.42)$$

Similar to (2.4.25), (2.4.42) projected on n^{th} order Hermit polynomials we obtain the relation between the coefficient tensors as

$$\mathbf{a}_{k+1}^{(n)} = -\tau \left(\sum_{m=0}^k \partial_t^{(k-m+1)} (\mathbf{a}_m^{(n)}) + \nabla_{\mathbf{x}}^{(1)} \cdot (\mathbf{a}_k^{(n+1)}) + \tilde{n} \nabla_{\mathbf{x}}^{(1)} \cdot (\mathbf{a}_k^{(n-1)} I) - \tilde{n} \mathbf{g} \cdot (\mathbf{a}_k^{(n-1)} I) \right). \quad (2.4.43)$$

From (2.4.43) we can see that the n^{th} Hermite coefficient in $f^{(k+1)}$ depends only on the temporal and spatial derivatives of the Hermite coefficients in $f^{(k)}$ of orders up to $n+1$. In particular, this implies that if the Hermite terms of order up to n are retained in the expansion of $f^{(0)}$ in (2.4.16), the first $n-k$ Hermite coefficients in the k^{th} order of the multiscale expansion will be the same as if the full Maxwell-Boltzmann equilibrium distribution (2.2.9) is used in (2.4.16). Or in another way, in order to have a n^{th} hydrodynamic moment accurate up to the k^{th} level, there is only the need for Maxwell-Boltzmann equilibrium distribution expanded up to $(n+k)^{th}$ order. For example, for the zeroth level Euler equations, the density, momentum, energy are recovered accurately provided the expansion for Maxwell-Boltzmann equilibrium distribution will achieve 1st, 2nd and 3rd order correspondingly. While for Navier-Stokes equations which is located on the first level of the expansion, the similar expansion would go up to 2nd, 3rd and 4th order.

Remark 2.4.5 It is worth to mention that the analysis above provides a general theoretical procedure for systematically formulating kinetic models beyond the Navier-Stokes

¹It will be explained by calculation in next section why it is enough up to one order for the expansion of spatial derivative, or see [29]

equations. Moving a step forward, by retaining up to the fourth order terms in the Hermite expansion, we can satisfy the Burnett level accuracy pertaining to the fluid momentum evolution for isothermal systems, whereas the accuracy for the energy evolution and thermohydrodynamics is achieved at the Burnett level of accuracy if the fifth order terms in Hermite expansion are retained [29].

2.4.6 Discretization to numerical model

As the conclusion has been drawn that the macroscopic Navier-Stokes equations can be traced all the way back to the kinetic model - the Boltzmann-BGK equation in the continuous system, discretization for the model is immediately demanded to solve the lattice Boltzmann equation numerically. Basically, there are three different kinds of independent variables needed to be discretized, namely the kinetic velocity ξ , the spatial variable x and the temporal variable t .

First of all, the velocity is mainly involved in the integral for projection of the Boltzmann-BGK equation on Hermite bases in the velocity space. Accordingly, the appropriate quadrature rule is needed for approximating the continuous integral up to desired order of accuracy. Another place where the velocity appears is in the force term $\mathbf{g} \cdot \nabla_{\xi} f$, for which the derivative with respect to velocity needs to be handled.

Secondly, discretization for the temporal and spatial variables in the convective form $\partial_t f + \xi \cdot \nabla_x f$ are welcoming in field of the computational fluid dynamics, where numerous numerical methods including finite difference, finite volume as well as finite element methods could be applied. A more detailed analysis and comments on these methods can be found in Succi's book [6]. In what follows, finite difference method is chosen preferably due to its simplicity.

Discretization of Velocity Space

Let $f^N(x, \xi, t)$ denote the truncated approximation for $f(x, \xi, t)$, defined by

$$f^N(x, \xi, t) = w(\xi) \sum_{n=0}^N \frac{1}{n!} \mathbf{a}^{(n)}(x, t) \mathcal{H}^{(n)}(\xi). \quad (2.4.44)$$

From (2.4.14) we know that the coefficients $\mathbf{a}^{(n)}$ in (2.4.44) are precisely given by f^N for $n = 1 \dots N$ as

$$\mathbf{a}^{(n)}(x, t) = \int f^N(x, \xi, t) \mathcal{H}^{(n)}(\xi) d\xi. \quad (2.4.45)$$

Since $f^N(x, \xi, t)/w(\xi)$ is a polynomial of velocity ξ up to no more than the N^{th} order and $\mathcal{H}^{(n)}(\xi)$ is also a polynomial of velocity ξ up to the N^{th} order, there exists $r(x, \xi, t)$, a polynomial of velocity ξ up to the $2N^{th}$ order that

$$f^N(x, \xi, t) \mathcal{H}^{(n)}(\xi) = w(\xi) r(x, \xi, t). \quad (2.4.46)$$

Hence, $\mathbf{a}^{(n)}$ can be calculated by using Gauss-Hermite quadrature² in multiple dimensions with a degree $\geq 2N^{th}$ order accuracy as

$$\mathbf{a}^{(n)}(\mathbf{x}, t) = \int w(\boldsymbol{\xi}) r(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi} = \sum_{i=0}^{q-1} w_i r(\mathbf{x}, \boldsymbol{\xi}_i, t) = \sum_{i=0}^{q-1} \frac{w_i}{w(\boldsymbol{\xi}_i)} f^N(\mathbf{x}, \boldsymbol{\xi}_i, t) \mathcal{H}^{(n)}(\boldsymbol{\xi}_i), \quad (2.4.47)$$

where w_i and $\boldsymbol{\xi}_i (i = 0, 1, \dots, q-1)$ are the weights and abscissae of the Gauss-Hermite quadrature respectively. Similarly, we have the coefficients $\mathbf{a}_0^{(n)}(\mathbf{x}, t)$ for the equilibrium distribution function $f^{(0)}$. In order to have the expression for each variable in the discretized lattice Boltzmann system, we need to rescale the spatial variable by δx and temporal variable by δt . As a result, all the velocities are rescaled by lattice velocity $c_l = \delta x / \delta t$, which is dependent on the sound velocity c_s up to different lattice structure [43]. It can be shown that, in the lattice Boltzmann system, the Hermite polynomials become

$$\left\{ \begin{array}{l} \mathcal{H}^{(0)}(\boldsymbol{\xi}) = 1, \\ \mathcal{H}^{(1)}(\boldsymbol{\xi}) = \boldsymbol{\xi}, \\ \mathcal{H}^{(2)}(\boldsymbol{\xi}) = \boldsymbol{\xi}\boldsymbol{\xi} - c_s^2 I, \\ \mathcal{H}^{(3)}(\boldsymbol{\xi}) = \boldsymbol{\xi}\boldsymbol{\xi}\boldsymbol{\xi} - c_s^2 3\boldsymbol{\xi}I, \\ \mathcal{H}^{(4)}(\boldsymbol{\xi}) = \boldsymbol{\xi}\boldsymbol{\xi}\boldsymbol{\xi}\boldsymbol{\xi} - \tilde{c}_s^2 \boldsymbol{\xi}\boldsymbol{\xi}I + \tilde{c}_s^4 II. \end{array} \right. \quad (2.4.48)$$

While the corresponding coefficient tensors for the distribution function $f(\mathbf{x}, \boldsymbol{\xi}, t)$ computed with (2.4.48), reads

$$\left\{ \begin{array}{l} \mathbf{a}^{(0)} = \rho, \\ \mathbf{a}^{(1)} = \rho \mathbf{u}, \\ \mathbf{a}^{(2)} = P + \rho(\mathbf{u}\mathbf{u} - c_s^2 I), \\ \mathbf{a}^{(3)} = K + \tilde{3}\mathbf{a}^{(2)} + (1 - D)\rho \mathbf{u}\mathbf{u}\mathbf{u}, \\ \mathbf{a}^{(4)} = L + \tilde{4}\mathbf{u}K + \tilde{6}\mathbf{u}\mathbf{u}P - \tilde{6}c_s^2 PI - \tilde{6}c_s^4 \rho \mathbf{u}\mathbf{u}I + \tilde{3}\rho II. \end{array} \right. \quad (2.4.49)$$

²more details about Gauss-Hermite quadrature and lattice structure can be found in the section 2.6.1

For the equilibrium distribution, we also get the coefficient tensors as

$$\begin{cases} \mathbf{a}_0^{(0)} = \rho, \\ \mathbf{a}_0^{(1)} = \rho \mathbf{u}, \\ \mathbf{a}_0^{(2)} = \rho \mathbf{u} \mathbf{u} + \rho(\theta - 1) c_s^2 I, \\ \mathbf{a}_0^{(3)} = \rho \mathbf{u} \mathbf{u} \mathbf{u} + \tilde{3}\rho(\theta - 1) \mathbf{u} c_s^2 I, \\ \mathbf{a}_0^{(4)} = \rho \mathbf{u} \mathbf{u} \mathbf{u} \mathbf{u} + \tilde{3}\rho(\theta - 1)^2 c_s^4 II + \tilde{6}\rho(\theta - 1) \mathbf{u} \mathbf{u} c_s^2 I. \end{cases} \quad (2.4.50)$$

All these coefficients are used the same as those before rescaling, as stated later on. For the sake of simplicity, define a new function f_i for $i = 0, 1, \dots, q - 1$ as

$$f_i(\mathbf{x}, t) = \frac{w_i}{w(\xi_i)} f(\mathbf{x}, \xi_i, t),$$

so that we have the N^{th} order truncated discrete equilibrium distribution function as

$$f_i^{(0)}(\mathbf{x}, t) = w_i \sum_{n=0}^N \frac{1}{c_s^{2n} n!} \mathbf{a}_0^{(n)}(\mathbf{x}, t) \mathcal{H}^{(n)}(\xi_i). \quad (2.4.51)$$

In order to derive the Navier-Stokes equations, the second order moment in the equilibrium distribution function is required, according to (2.4.50), it is

$$f_i^{(0)}(\mathbf{x}, t) = w_i \rho \left(1 + \frac{\xi_i \cdot \mathbf{u}}{c_s^2} + \frac{(\xi_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} \right), \quad (2.4.52)$$

while the N^{th} order truncated body force term is given as

$$F_i(\mathbf{x}, t) = w_i \sum_{n=1}^{N-1} \frac{1}{c_s^{2n} n!} \tilde{n} \mathbf{g}(\mathbf{x}, t) \cdot (\mathbf{a}_0^{(n-1)}(\mathbf{x}, t) c_s^2 I) \mathcal{H}^{(n)}(\xi_i). \quad (2.4.53)$$

Equation (2.4.53) up to the second order can be explicitly written as

$$F_i(\mathbf{x}, t) = w_i \rho \left(\frac{\xi_i \cdot \mathbf{g}}{c_s^2} + \frac{(\xi_i \cdot \mathbf{g})(\xi_i \cdot \mathbf{u})}{c_s^4} - \frac{\mathbf{g} \cdot \mathbf{u}}{c_s^2} \right). \quad (2.4.54)$$

With all the functions discretized with respect to the velocity at hand, the discrete BGK-Boltzmann equation is given for $i = 0, 1, \dots, q - 1$ as

$$\partial_t f_i + \xi_i \cdot \nabla f_i = -\frac{1}{\tau} (f_i - f_i^{(0)}) + F_i \quad (2.4.55)$$

which can be written in Lagrangian formulation as

$$\frac{df_i}{dt} = -\frac{1}{\tau} (f_i - f_i^{(s)}), \quad (2.4.56)$$

where for $i = 0, 1, \dots, q - 1$,

$$f_i^{(s)} := f_i^{(0)} + \tau F_i. \quad (2.4.57)$$

Time and Space Discretization

Now it's left to look at the temporal and spatial discretization for $f_i(\mathbf{x}, t)$. The first order approximation is the explicit approximation for the discretization of time and space, reading as for $i = 0, 1, \dots, q - 1$,

$$f_i(\mathbf{x} + \boldsymbol{\xi}_i \delta t, t + \delta t) - f_i(\mathbf{x}, t) = -\frac{\delta t}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(s)}(\mathbf{x}, t) \right), \quad (2.4.58)$$

separated into collision step and streaming step for $i = 0, 1, \dots, q - 1$ as

$$\text{collision: } \tilde{f}_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) - \frac{\delta t}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(s)}(\mathbf{x}, t) \right) \quad (2.4.59)$$

$$\text{streaming: } f_i(\mathbf{x} + \boldsymbol{\xi}_i \delta t, t + \delta t) = \tilde{f}_i(\mathbf{x}, t). \quad (2.4.60)$$

When we have the discrete distribution functions, the macro thermodynamic variables can be expressed in the following formulas

$$\text{mass density: } \rho = \sum_{i=0}^{q-1} f_i, \quad (2.4.61)$$

$$\text{momentum density: } \rho \mathbf{u} = \sum_{i=0}^{q-1} f_i \boldsymbol{\xi}_i, \quad (2.4.62)$$

$$\text{kinetic energy density: } \rho \epsilon = \frac{1}{2} \sum_{i=0}^{q-1} f_i \mathbf{c}_i^2, \quad (2.4.63)$$

$$\text{pressure or stress tensor: } P = \sum_{i=0}^{q-1} f_i \mathbf{c}_i \mathbf{c}_i, \quad (2.4.64)$$

$$\text{energy or heat flux: } \mathbf{q} = \frac{1}{2} \sum_{i=0}^{q-1} f_i \mathbf{c}_i \mathbf{c}_i^2, \quad (2.4.65)$$

where $\mathbf{c}_i = \boldsymbol{\xi}_i - \mathbf{u}$ for $i = 0, 1, \dots, q - 1$ is the discrete counterpart of the continuous intrinsic or mean velocity.

The second order approximation is an implicit approximation by integrating (2.4.56)

for once with respect to time t and applying the trapezoidal rule as

$$\begin{aligned}
 & f_i(\mathbf{x} + \boldsymbol{\xi}_i \delta t, t + \delta t) - f_i(\mathbf{x}, t) \\
 &= -\frac{1}{\tau} \int_0^{\delta t} (f_i(\mathbf{x} + \boldsymbol{\xi}_i s, t + s) - f_i^{(eq)}(\mathbf{x} + \boldsymbol{\xi}_i s, t + s)) ds \\
 &= -\frac{\delta t}{2\tau} \left(f_i(\mathbf{x} + \boldsymbol{\xi}_i \delta t, t + \delta t) - f_i^{(s)}(\mathbf{x} + \boldsymbol{\xi}_i \delta t, t + \delta t) \right) \\
 &\quad - \frac{\delta t}{2\tau} \left(f_i(\mathbf{x}, t) - f_i^{(s)}(\mathbf{x}, t) \right) + \mathcal{O}(\delta t^2).
 \end{aligned} \tag{2.4.66}$$

To change this implicit approximation discrete iteration into a explicit formulation, we define a new function

$$\hat{f}_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) + \frac{\delta t}{2\tau} \left(f_i(\mathbf{x}, t) - f_i^{(s)}(\mathbf{x}, t) \right). \tag{2.4.67}$$

Therefore, (2.4.66) becomes

$$\hat{f}_i(\mathbf{x} + \boldsymbol{\xi}_i \delta t, t + \delta t) - \hat{f}_i(\mathbf{x}, t) = -\frac{\delta t}{\hat{\tau}} \left(\hat{f}_i(\mathbf{x}, t) - f_i^{(s)}(\mathbf{x}, t) \right), \tag{2.4.68}$$

where the relaxation time as well as the dynamic shear viscosity and thermal diffusivity turn out to be

$$\hat{\tau} = \tau + \frac{1}{2}, \quad \mu_s = c_s^2 \rho \theta (\hat{\tau} - \frac{1}{2}), \text{ and } k = c_s^4 \rho \theta (\hat{\tau} - \frac{1}{2}). \tag{2.4.69}$$

Remark 2.4.6 It is important to observe that in the system represented by the new distribution function \hat{f}_i , we have

$$\hat{\rho} = \sum_{i=0}^{q-1} \hat{f}_i = \sum_{i=0}^{q-1} f_i = \rho \tag{2.4.70}$$

$$\hat{\rho} \hat{\mathbf{u}} = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \hat{f}_i = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \left(f_i + \frac{1}{2\tau} (f_i - f_i^{(0)}) - \frac{1}{2} F_i \right) = \rho \mathbf{u} - \frac{1}{2} \rho \mathbf{g}. \tag{2.4.71}$$

Equation (2.4.70) and (2.4.71) lead to

$$\hat{\mathbf{u}} = \mathbf{u} - \frac{1}{2} \mathbf{g}. \tag{2.4.72}$$

Finally, we have from (2.4.57), (2.4.68) and (2.4.69) that

$$\hat{f}_i(\mathbf{x} + \boldsymbol{\xi}_i \delta t, t + \delta t) - \hat{f}_i(\mathbf{x}, t) = -\frac{\delta t}{\hat{\tau}} \left(\hat{f}_i(\mathbf{x}, t) - f_i^{(0)}(\mathbf{x}, t) \right) + \delta t \left(1 - \frac{1}{2\hat{\tau}} \right) F_i(\rho, \mathbf{u}). \tag{2.4.73}$$

For the sake of simplicity, we still denote $f_i = \hat{f}_i$, $\tau = \hat{\tau}$ and set $F_i = (1 - \frac{1}{2\hat{\tau}})F_i$ so that by (2.4.54), we have

$$F_i(\mathbf{x}, t) = w_i \left(1 - \frac{1}{2\tau} \right) \left(\frac{\boldsymbol{\xi}_i \cdot \mathbf{F}}{c_s^2} + \frac{(\boldsymbol{\xi}_i \cdot \mathbf{F})(\boldsymbol{\xi}_i \cdot \mathbf{u})}{c_s^4} - \frac{\mathbf{F} \cdot \mathbf{u}}{c_s^2} \right). \quad (2.4.74)$$

Rescaling the time step δt by itself into lattice Boltzmann system, we have that (2.4.73) becomes

$$f_i(\mathbf{x} + \boldsymbol{\xi}_i, t + 1) - f_i(\mathbf{x}, t) = -\frac{1}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(0)}(\mathbf{x}, t) \right) + F_i(\rho, \mathbf{u}). \quad (2.4.75)$$

Separating (2.4.75) into the following two steps we obtain the classical lattice Boltzmann formulation for $i = 0, 1, \dots, q-1$ as

$$\text{collision: } \tilde{f}_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) - \frac{1}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(0)}(\mathbf{x}, t) \right) + F_i(\rho, \mathbf{u}), \quad (2.4.76)$$

$$\text{streaming: } f_i(\mathbf{x} + \boldsymbol{\xi}_i, t + 1) = \tilde{f}_i(\mathbf{x}, t),$$

where a collision step and a streaming step can be identified.

2.5 Chapman-Enskog expansion approach

As stated before, the Grad's approach is appropriate to achieve high order of kinetic representation of hydrodynamics, from Euler equations to Navier-Stokes equations and beyond. In the Chapman-Enskog expansion approach, the derivation from the Boltzmann-BGK equation to Navier-Stokes equations can find its roots in lattice gas automata, in which field many results are ready to be used. Moreover, the Chapman-Enskog expansion approach could also be used to analyze the numerical accuracy of initial condition and boundary conditions(see[40]). Roughly speaking, the Chapman-Enskog expansion is based on the multi-scale expansion of the distribution function, spatial and temporal variables as well as the projection of the lattice Boltzmann equation on the lattice symmetries (see later) in different scales so as to get the corresponding macroscopic equations [42].

2.5.1 Lattice Boltzmann-BGK Model

Unlike the Grad's representation approach, where the derivation from the Boltzmann-BGK equation to Navier-Stokes equations is carried out in continuous system and then discretization is applied to the continuous system to obtain the numerical model, the Chapman-Enskog expansion starts from the discrete system - lattice Boltzmann system directly³. The lattice structure of the discrete system is originated from the

³in lattice Boltzmann system, space step δx and time step δt are rescaled to be 1, so is the lattice velocity $c_l := \delta x / \delta t = 1$

Lattice-Gas Cellular Automata model (see [11]), while it can also be obtained from the Gauss-Hermite quadrature for the weights as well as discrete velocity - the abscissae of the quadrature (see section 2.6.1).

The lattice Boltzmann - BGK model is given in the lattice Boltzmann system (all variables have lattice units, in particular $\delta x = 1, \delta t = 1$) as

$$f_i(\mathbf{x} + \boldsymbol{\xi}_i, t + 1) - f_i(\mathbf{x}, t) = -\frac{1}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(0)}(\mathbf{x}, t) \right) + F_i(\mathbf{x}, t), \quad (2.5.1)$$

where $f_i(\mathbf{x} + \boldsymbol{\xi}_i, t + 1), f_i(\mathbf{x}, t)$ are the discrete distribution functions and $F_i(\mathbf{x}, t)$ is the unknown discrete body force along the spatial direction $\boldsymbol{\xi}_i$ for $i = 0, 1, \dots, q - 1$. τ is the same as in (2.4.75) - the single relaxation time. And the lattice is constructed such that the two adjacent nodes in the lattice are connected by the spatial discrete velocity $\boldsymbol{\xi}_i$.

The equilibrium distribution up to the second order moment of the macroscopic velocity is directly given the same as (2.4.52). The force term $F_i(\mathbf{x}, t)$ will be specified as a modification in deriving Navier-Stokes equations. The macroscopic density and momentum are defined under the conservation laws as

$$\rho(\mathbf{x}, t) = \sum_{i=0}^{q-1} f_i(\mathbf{x}, t) = \sum_{i=0}^{q-1} f_i^{(0)}(\mathbf{x}, t), \quad (2.5.2)$$

$$\mathbf{j}(\mathbf{x}, t) = \rho(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t) = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i(\mathbf{x}, t) = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)}(\mathbf{x}, t). \quad (2.5.3)$$

2.5.2 Chapman-Enskog expansion

The basic idea behind Chapman-Enskog expansion is to separate the physical time and space as well as distribution function into multiple scales with respect to the order of Knudsen number ε [42]. Physical properties of the macroscopic variables, including the density and momentum, are automatically separated into the corresponding different scales. By applying Taylor expansion for the convection term of the lattice Boltzmann-BGK equation and analyzing it in different scales, the physical properties of mass conservation and momentum conservation are formulated into continuity equation and momentum equation respectively, ending up with the Navier-Stokes equations.

Recall the lattice Boltzmann-BGK equation with the fictitious lattice collision Ω_i defined as

$$\Omega_i(\mathbf{x}, t) := -\frac{1}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(0)}(\mathbf{x}, t) \right) + F_i(\mathbf{x}, t) = f_i(\mathbf{x} + \boldsymbol{\xi}_i, t + 1) - f_i(\mathbf{x}, t). \quad (2.5.4)$$

The Chapman-Enskog expansion for time t , space \mathbf{x} , distribution function f_i together with collision Ω_i in different scales of Knudsen number, in two different scales are given as

$$\begin{cases} \partial_t = \varepsilon \partial_{t_1} + \varepsilon^2 \partial_{t_2} + O(\varepsilon^3), \\ \nabla = \varepsilon \nabla_1 + \varepsilon^2 \nabla_2 + O(\varepsilon^3), \\ f_i = f_i^{(0)} + \varepsilon f_i^{(1)} + O(\varepsilon^2), \\ \Omega_i = \varepsilon \Omega_i^{(1)} + \varepsilon^2 \Omega_i^{(2)} + O(\varepsilon^3). \end{cases} \quad (2.5.5)$$

In most of the literatures, the spatial derivative is expanded only up to the first order $\nabla = \varepsilon \nabla_1 + O(\varepsilon^2)$. The reason for this could be intuitively explained as that in order to achieve the Navier-Stokes equations, involving both the second order derivative - diffusion term $\Delta \mathbf{u}$ and the first order derivative - convection term $(\mathbf{u} \cdot \nabla) \mathbf{u}$ while the temporal derivative is only the first order $\partial_t(\rho \mathbf{u})$, we need to parallelize the spatial derivative and temporal derivative to the same order, i.e. second order in time and first order in space. In the following multi-scale analysis, we first stick to the second order expansion for space as in 2.5.5

Applying Taylor expansion with respect to $f_i(\mathbf{x}, t)$ for the lattice Boltzmann equation (2.5.4) up to the second order, we have

$$\begin{aligned} \Omega_i(\mathbf{x}, t) &= f_i(\mathbf{x} + \boldsymbol{\xi}_i, t + 1) - f_i(\mathbf{x}, t) \\ &= (\partial_t + \nabla \cdot \boldsymbol{\xi}_i) f_i + \frac{1}{2} (\partial_t^2 + 2\partial_t \nabla \cdot \boldsymbol{\xi}_i + \nabla \nabla : \boldsymbol{\xi}_i \boldsymbol{\xi}_i) f_i + O(d_t^3) f_i. \end{aligned} \quad (2.5.6)$$

where $O(d_t^3) f_i$ denote higher order of derivatives. Substituting (2.5.5) into (2.5.6), we obtain the multi-scale expansion as

$$\begin{aligned} \varepsilon \Omega_i^{(1)} + \varepsilon^2 \Omega_i^{(2)} + O(\varepsilon^3) &= (\varepsilon \partial_{t_1} + \varepsilon^2 \partial_{t_2} + \varepsilon \nabla_1 \cdot \boldsymbol{\xi}_i + \varepsilon^2 \nabla_2 \cdot \boldsymbol{\xi}_i + O(\varepsilon^3)) \\ &\quad \left(f_i^{(0)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} + O(\varepsilon^3) \right) \\ &\quad + \frac{1}{2} \left((\varepsilon \partial_{t_1} + \varepsilon^2 \partial_{t_2} + O(\varepsilon^3))^2 + (\varepsilon \nabla_1 + \varepsilon^2 \nabla_2 + O(\varepsilon^3))^2 : \boldsymbol{\xi}_i \boldsymbol{\xi}_i \right. \\ &\quad \left. + 2(\varepsilon \partial_{t_1} + \varepsilon^2 \partial_{t_2} + O(\varepsilon^3))(\varepsilon \nabla_1 \cdot \boldsymbol{\xi}_i + \varepsilon^2 \nabla_2 \cdot \boldsymbol{\xi}_i + O(\varepsilon^3)) \right) \\ &\quad \left(f_i^{(0)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} + O(\varepsilon^3) \right) \\ &= \underbrace{(\varepsilon(\partial_{t_1} + \nabla_1 \cdot \boldsymbol{\xi}_i) + \varepsilon^2(\partial_{t_2} + \nabla_2 \cdot \boldsymbol{\xi}_i))}_{1^{st} order} + \underbrace{\varepsilon^2(\partial_{t_2} + \nabla_2 \cdot \boldsymbol{\xi}_i)}_{2^{nd} order} \end{aligned}$$

$$\begin{aligned}
& + \frac{1}{2} \varepsilon^2 \underbrace{(\partial_{t_1}^2 + 2\partial_{t_1}\nabla_1 \cdot \boldsymbol{\xi}_i + \nabla_1\nabla_1 : \boldsymbol{\xi}_i \boldsymbol{\xi}_i)}_{2^{nd} order} + O(\varepsilon^3) \\
& \left(f_i^{(0)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} + O(\varepsilon^3) \right). \tag{2.5.7}
\end{aligned}$$

Separating the multi-scale expansion (2.5.7) on different scales of ε , we have the following two equations on the scales of ε and ε^2 :

$$\begin{aligned}
\text{for } \varepsilon : \quad \Omega_i^{(1)} &= (\partial_{t_1} + \nabla_1 \cdot \boldsymbol{\xi}_i) f_i^{(0)}, \\
\text{for } \varepsilon^2 : \quad \Omega_i^{(2)} &= (\partial_{t_1} + \nabla_1 \cdot \boldsymbol{\xi}_i) f_i^{(1)} + (\partial_{t_2} + \nabla_2 \cdot \boldsymbol{\xi}_i) f_i^{(0)} \\
& + \frac{1}{2} (\partial_{t_1}^2 + 2\partial_{t_1}\nabla_1 \cdot \boldsymbol{\xi}_i + \nabla_1\nabla_1 : \boldsymbol{\xi}_i \boldsymbol{\xi}_i) f_i^{(0)}. \tag{2.5.8}
\end{aligned}$$

Corresponding to the expansion of distribution function, from (2.5.2) and (2.5.3), we have the expansion for momentum as

$$\mathbf{j} = \mathbf{j}^{(0)} + \varepsilon \mathbf{j}^{(1)} + O(\varepsilon^2). \tag{2.5.9}$$

The force term $F(\mathbf{x}, t)$, which is responsible for the change of momentum is located on the first order scale as

$$\mathbf{F} = \varepsilon \mathbf{F}^{(1)} + O(\varepsilon^2), \tag{2.5.10}$$

and suppose

$$\mathbf{j}^{(1)} = -\mathbf{F}^{(1)}/2, \tag{2.5.11}$$

which will be highly depended for derivation later on. Therefore, the zeroth order and first order collision invariants can be written as

$$\begin{cases} 0 = \sum_{i=0}^{q-1} \Omega_i = \sum_{i=0}^{q-1} \left(\varepsilon \Omega_i^{(1)} + \varepsilon^2 \Omega_i^{(2)} + O(\varepsilon^3) \right), \\ \mathbf{F} = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \Omega_i = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \left(\varepsilon \Omega_i^{(1)} + \varepsilon^2 \Omega_i^{(2)} + O(\varepsilon^3) \right). \end{cases} \tag{2.5.12}$$

Hence, the zeroth order (with respect to velocity $\boldsymbol{\xi}$) macroscopic equations are given on the scales of ε and ε^2 respectively by collision invariant from (2.5.8) as

$$\begin{aligned}
\text{for } \varepsilon : \quad 0 &= \sum_{i=0}^{q-1} \Omega_i^{(1)} = \sum_{i=0}^{q-1} (\partial_{t_1} + \nabla_1 \cdot \boldsymbol{\xi}_i) f_i^{(0)} \\
& = \partial_{t_1} \left(\sum_{i=0}^{q-1} f_i^{(0)} \right) + \nabla_1 \cdot \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)} \right) \\
& = \partial_{t_1} \rho + \nabla_1 \cdot \mathbf{j}^{(0)}. \tag{2.5.13}
\end{aligned}$$

Noticing $\boldsymbol{\xi}_i \boldsymbol{\xi}_i = Q_i + c_s^2 I$ and defining $\Pi := \sum_{i=0}^{q-1} Q_i f_i$, we have

$$\begin{aligned}
\text{for } \varepsilon^2 : \quad 0 &= \sum_{i=0}^{q-1} \Omega_i^{(2)} = \sum_{i=0}^{q-1} \left((\partial_{t_1} + \nabla_1 \cdot \boldsymbol{\xi}_i) f_i^{(1)} + (\partial_{t_2} + \nabla_2 \cdot \boldsymbol{\xi}_i) f_i^{(0)} \right. \\
&\quad \left. + \frac{1}{2} (\partial_{t_1}^2 + 2\partial_{t_1} \nabla_1 \cdot \boldsymbol{\xi}_i + \nabla_1 \nabla_1 : \boldsymbol{\xi}_i \boldsymbol{\xi}_i) f_i^{(0)} \right) \\
&= \partial_{t_1} \left(\sum_{i=0}^{q-1} f_i^{(1)} \right) + \nabla_1 \cdot \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(1)} \right) + \partial_{t_2} \left(\sum_{i=0}^{q-1} f_i^{(0)} \right) \\
&\quad + \nabla_2 \cdot \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)} \right) + \frac{1}{2} \partial_{t_1}^2 \left(\sum_{i=0}^{q-1} f_i^{(0)} \right) + \partial_{t_1} \nabla_1 \cdot \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)} \right) \\
&\quad + \frac{1}{2} \nabla_1 \nabla_1 : \left(\sum_{i=0}^{q-1} (Q_i + c_s^2 I) f_i^{(0)} \right) \\
&= \nabla_1 \cdot \mathbf{j}^{(1)} + \partial_{t_2} \rho + \nabla_2 \cdot \mathbf{j}^{(0)} \\
&\quad + \underbrace{\frac{1}{2} \partial_{t_1}^2 \rho + \partial_{t_1} \nabla_1 \cdot \mathbf{j}^{(0)} + \frac{1}{2} \nabla_1 \nabla_1 : \Pi^{(0)} + \frac{1}{2} c_s^2 \nabla_1^2 \rho}_{\text{by (2.5.13)}} \\
&= \nabla_1 \cdot \mathbf{j}^{(1)} + \partial_{t_2} \rho + \nabla_2 \cdot \mathbf{j}^{(0)} \\
&\quad + \frac{1}{2} \partial_{t_1} \nabla_1 \cdot \mathbf{j}^{(0)} + \frac{1}{2} \nabla_1 \nabla_1 : \Pi^{(0)} + \frac{1}{2} c_s^2 \nabla_1^2 \rho. \tag{2.5.14}
\end{aligned}$$

While the first order (with respect to velocity $\boldsymbol{\xi}$) macroscopic equations are given on the scales of ε from (2.5.8) as

$$\begin{aligned}
\text{for } \varepsilon : \quad \mathbf{F}^{(1)} &= \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \Omega_i^{(1)} = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i (\partial_{t_1} + \nabla_1 \cdot \boldsymbol{\xi}_i) f_i^{(0)} \\
&= \partial_{t_1} \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)} \right) + \nabla_1 \cdot \left(\sum_{i=0}^{q-1} (Q_i + c_s^2 I) f_i^{(0)} \right) \\
&= \partial_{t_1} \mathbf{j}^{(0)} + \nabla_1 \cdot \Pi^{(0)} + c_s^2 \nabla_1 \rho, \tag{2.5.15}
\end{aligned}$$

from which we have

$$\begin{cases} \partial_{t_1} \nabla_1 \cdot \mathbf{j}^{(0)} = \nabla_1 \cdot \mathbf{F}^{(1)} - \nabla_1 \nabla_1 : \Pi^{(0)} - c_s^2 \nabla_1^2 \rho, \\ \partial_{t_1}^2 \mathbf{j}^{(0)} = \partial_{t_1} \mathbf{F}^{(1)} - \partial_{t_1} \nabla_1 \cdot \Pi^{(0)} - c_s^2 \partial_{t_1} \nabla_1 \rho. \end{cases} \tag{2.5.16}$$

Substituting (2.5.16)₁ and (2.5.11) into (2.5.14), we obtain the simplified macroscopic equation with zeroth order of moment on the second scale (2.5.14) as

$$\begin{aligned} 0 = & -\frac{1}{2}\nabla_1 \cdot \mathbf{F}^{(1)} + \partial_{t_2}\rho + \nabla_2 \cdot \mathbf{j}^{(0)} \\ & + \frac{1}{2}\left(\nabla_1 \cdot \mathbf{F}^{(1)} - \nabla_1 \nabla_1 : \Pi^{(0)} - c_s^2 \nabla_1^2 \rho\right) \\ & + \frac{1}{2}\nabla_1 \nabla_1 : \Pi^{(0)} + \frac{1}{2}c_s^2 \nabla_1^2 \rho \\ & = \partial_{t_2}\rho + \nabla_2 \cdot \mathbf{j}^{(0)}. \end{aligned} \quad (2.5.17)$$

Equations (2.5.13) and (2.5.17) lead to the continuity equation as (noticing $\mathbf{j}^{(0)} = \rho \mathbf{u}$)

$$\partial_t\rho + \nabla \cdot (\rho \mathbf{u}) = 0. \quad (2.5.18)$$

Defining

$$R := \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i f_i,$$

we have the first order (with respect to velocity $\boldsymbol{\xi}$) macroscopic equations given on the scales of ε^2 from (2.5.8) as

$$\begin{aligned} \text{for } \varepsilon^2 : \quad 0 = & \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \Omega_i^{(2)} = \sum_{i=0}^{q-1} \left(\boldsymbol{\xi}_i (\partial_{t_1} + \nabla_1 \cdot \boldsymbol{\xi}_i) f_i^{(1)} + \boldsymbol{\xi}_i (\partial_{t_2} + \nabla_2 \cdot \boldsymbol{\xi}_i) f_i^{(0)} \right. \\ & \left. + \frac{1}{2} \boldsymbol{\xi}_i (\partial_{t_1}^2 + 2\partial_{t_1} \nabla_1 \cdot \boldsymbol{\xi}_i + \nabla_1 \nabla_1 : \boldsymbol{\xi}_i \boldsymbol{\xi}_i) f_i^{(0)} \right) \\ = & \partial_{t_1} \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(1)} \right) + \nabla_1 \cdot \left(\sum_{i=0}^{q-1} (\Omega_i + c_s^2 I) f_i^{(1)} \right) + \partial_{t_2} \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)} \right) \\ & + \nabla_2 \cdot \left(\sum_{i=0}^{q-1} (\Omega_i + c_s^2 I) f_i^{(0)} \right) + \frac{1}{2} \partial_{t_1}^2 \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)} \right) \\ & + \partial_{t_1} \nabla_1 \cdot \left(\sum_{i=0}^{q-1} (\Omega_i + c_s^2 I) f_i^{(0)} \right) + \frac{1}{2} \nabla_1 \nabla_1 : \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i f_i^{(0)} \right) \\ = & \underbrace{\partial_{t_1} \mathbf{j}^{(1)}}_{\text{by (2.5.11)}} + \nabla_1 \cdot \Pi^{(1)} + \partial_{t_2} \mathbf{j}^{(0)} + \nabla_2 \cdot \Pi^{(0)} + c_s^2 \nabla_2 \rho \\ & + \frac{1}{2} \underbrace{\partial_{t_1}^2 \mathbf{j}^{(0)}}_{\text{by (2.5.16)}} + \partial_{t_1} \nabla_1 \cdot \Pi^{(0)} + c_s^2 \partial_{t_1} \nabla_1 \rho + \frac{1}{2} \nabla_1 \nabla_1 : R^{(0)} \\ = & -\frac{1}{2} \partial_{t_1} \mathbf{F}^{(1)} + \nabla_1 \cdot \Pi^{(1)} + \partial_{t_2} \mathbf{j}^{(0)} + \nabla_2 \cdot \Pi^{(0)} \end{aligned}$$

$$\begin{aligned}
& + c_s^2 \nabla_2 \rho + \frac{1}{2} \left(\partial_{t_1} \mathbf{F}^{(1)} - \partial_{t_1} \nabla_1 \cdot \Pi^{(0)} - c_s^2 \partial_{t_1} \nabla_1 \rho \right) \\
& + \partial_{t_1} \nabla_1 \cdot \Pi^{(0)} + c_s^2 \partial_{t_1} \nabla_1 \rho + \frac{1}{2} \nabla_1 \nabla_1 : R^{(0)} \\
& = \nabla_1 \cdot \Pi^{(1)} + \partial_{t_2} \mathbf{j}^{(0)} + \nabla_2 \cdot \Pi^{(0)} + c_s^2 \nabla_2 \rho \\
& + \frac{1}{2} \left(\partial_{t_1} \nabla_1 \cdot \Pi^{(0)} + c_s^2 \partial_{t_1} \nabla_1 \rho + \nabla_1 \nabla_1 : R^{(0)} \right). \tag{2.5.19}
\end{aligned}$$

Therefore, for the first order of moment we have the full macroscopic equation by taking $\varepsilon \times (2.5.15)$ + $\varepsilon^2 \times (2.5.19)$:

$$\begin{aligned}
\varepsilon \mathbf{F}^{(1)} &= (\varepsilon \partial_{t_1} + \varepsilon^2 \partial_{t_2}) \mathbf{j}^{(0)} + \frac{1}{2} (\varepsilon \nabla_1 + \varepsilon^2 \nabla_2) \cdot (\varepsilon \partial_{t_1} \Pi^{(0)} + c_s^2 \varepsilon \partial_{t_1} \rho I + \varepsilon \nabla_1 \cdot R^{(0)}) \\
& - \frac{1}{2} \varepsilon^2 \nabla_2 \cdot (\varepsilon \partial_{t_1} \Pi^{(0)} + c_s^2 \varepsilon \partial_{t_1} \rho I + \varepsilon \nabla_1 \cdot R^{(0)}) + (c_s^2 (\varepsilon \nabla_1 + \varepsilon^2 \nabla_2) \rho) \\
& + (\varepsilon \nabla_1 \cdot (\Pi^{(0)} + \varepsilon \Pi^{(1)}) + \varepsilon^2 \nabla_2 \cdot \Pi^{(0)} + \varepsilon^2 \nabla_2 \cdot (\varepsilon \Pi^{(1)}) - \varepsilon^2 \nabla_2 \cdot (\varepsilon \Pi^{(1)})) \\
& = \partial_t \mathbf{j}^{(0)} + \frac{1}{2} \nabla \cdot (\varepsilon \partial_{t_1} \Pi^{(0)} + c_s^2 \varepsilon \partial_{t_1} \rho I + \varepsilon \nabla_1 \cdot R^{(0)}) + c_s^2 \nabla \rho + \nabla \cdot \Pi + O(\varepsilon^3). \tag{2.5.20}
\end{aligned}$$

From (2.5.20) we can see that if the spatial derivative is expanded up to the first order $\nabla = \varepsilon \nabla_1$, all of the terms with ∇_2 vanish, and we can also have the accuracy up to the second order of Knudsen number. Moreover, in order to have the second order of accuracy, temporal derivative up to the second order is necessary. This multi-scale derivation could explain theoretically why we may only provide expansion of the spatial derivative to the first order of Knudsen number while the second order is needed for temporal expansion.

By the definition of equilibrium distribution function (2.4.52), we can compute $\mathbf{j}^{(0)}$, $\Pi^{(0)}$, $R^{(0)}$ explicitly provided the following lattice symmetries (which are related to the orthonormal properties of Hermite polynomials)

$$\left\{
\begin{aligned}
& \sum_{i=0}^{q-1} w_i = 1, \quad \sum_{i=0}^{q-1} w_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i = c_s^2 I = c_s^2 \delta_{\alpha_1 \alpha_2}, \\
& \sum_{i=0}^{q-1} w_i \boldsymbol{\xi}_i = 0, \quad \sum_{i=0}^{q-1} w_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i = 0, \quad \sum_{i=0}^{q-1} w_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i = 0, \\
& \sum_{i=0}^{q-1} w_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i = \tilde{3} c_s^4 II = c_s^4 (\delta_{\alpha_1 \alpha_2} \delta_{\alpha_3 \alpha_4} + \delta_{\alpha_1 \alpha_3} \delta_{\alpha_2 \alpha_4} + \delta_{\alpha_1 \alpha_4} \delta_{\alpha_2 \alpha_3}). \tag{2.5.21}
\end{aligned}
\right.$$

Therefore $\mathbf{j}^{(0)}, \Pi^{(0)}, R^{(0)}$ are given by

$$\begin{cases} \mathbf{j}^{(0)} = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)} = \rho \mathbf{u}, & \Pi^{(0)} = \sum_{i=0}^{q-1} Q_i f_i^{(0)} = \rho \mathbf{u} \mathbf{u}, \\ \nabla_1 \cdot R^{(0)} = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i f_i^{(0)} = c_s^2 (\nabla_1(\rho \mathbf{u}) + (\nabla_1(\rho \mathbf{u}))^T + \nabla_1 \cdot (\rho \mathbf{u}) I). \end{cases} \quad (2.5.22)$$

(2.5.20) is simplified with accuracy up to the second order of Knudsen number as

$$\begin{aligned} \mathbf{F} &= \partial_t \mathbf{j}^{(0)} + \nabla \cdot \left(\Pi^{(0)} + \varepsilon \Pi^{(1)} + c_s^2 \rho I + \frac{\varepsilon}{2} (\partial_{t_1}(\Pi^{(0)} + c_s^2 \rho I) + \nabla_1 \cdot R^{(0)}) \right) \\ &= \partial_t(\rho \mathbf{u}) + \nabla \cdot \left(\rho \mathbf{u} \mathbf{u} + c_s^2 \rho I + (\varepsilon \Pi^{(1)} + \frac{\varepsilon}{2} \partial_{t_1}(\Pi^{(0)})) \right) \\ &\quad + \nabla \cdot \left(\frac{1}{2} c_s^2 (\nabla(\rho \mathbf{u}) + (\nabla(\rho \mathbf{u}))^T) + \frac{1}{2} c_s^2 (\varepsilon \partial_{t_1} \rho + \nabla \cdot (\rho \mathbf{u})) I \right), \end{aligned} \quad (2.5.23)$$

which is explicit expressed except for the first order stress tensor $\Pi^{(1)}$, left to be approximated later on. Recall the full Navier-Stokes equations in the following form

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \\ \partial_t(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \mathbf{u} + \rho I - \boldsymbol{\tau}) = \rho \mathbf{g}, \end{cases} \quad (2.5.24)$$

where the deviatoric stresses $\boldsymbol{\tau}$ is defined as

$$\boldsymbol{\tau} = \rho \nu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T), \quad (2.5.25)$$

where ν is the dynamic shear viscosity, or with an additional term

$$\boldsymbol{\tau} = -\rho \nu' (\nabla \cdot \mathbf{u}) I + \rho \nu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T), \quad (2.5.26)$$

where ν' is the dynamic bulk viscosity. Comparatively, the continuity equation are the same as we can see from (2.5.18) and (2.5.24)₁, while the momentum equations are the same in (2.5.23) and (2.5.24)₂ except for the item of deviatoric stresses. In order to retrieve the exact Navier-Stokes equations, the so called Chapman-Enskog ansatz is applied to the collision operator to achieve the term with dynamic shear viscosity and an additional term with dynamic bulk viscosity respectively, in the following section.

Remark 2.5.1 *The Navier-Stokes equations (2.5.24) with deviatoric stresses given by (2.5.25) is essentially the same as those derived by Grad's representation approach in (2.4.37). While the additional term involving dynamic bulk viscosity in (2.5.26) hasn't been captured by Grad's representation approach.*

2.5.3 Chapman-Enskog ansatz

Suppose that the collision operator is given in the form of a BGK relaxation term as

$$\Omega_i(\mathbf{x}, t) := -\frac{1}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(0)}(\mathbf{x}, t) \right) + F_i(\mathbf{x}, t) + C_i(\mathbf{x}, t), \quad (2.5.27)$$

where F_i is the force term and C_i is the correction term and they are given by

$$\begin{cases} F_i(\mathbf{x}, t) = \frac{a}{c_s^2} w_i \boldsymbol{\xi}_i \cdot \mathbf{F} + \frac{b}{2c_s^4} w_i Q_i : (\mathbf{F}\mathbf{u} + \mathbf{u}\mathbf{F}), \\ C_i(\mathbf{x}, t) = \frac{c}{2c_s^4} w_i Q_i : \delta\Pi. \end{cases} \quad (2.5.28)$$

where the coefficients a, b, c are left to be determined. We will show in several steps that the force term F_i is responsible for the dynamic shear viscosity ν while the additional correction term C_i is responsible for the dynamic bulk viscosity ν' . Firstly, corresponding to the scale expansion of force \mathbf{F} and second order tensor $\delta\Pi$, we have on the first scale of the expansion for force term and correction term that

$$\begin{cases} F_i^{(1)}(\mathbf{x}, t) = \frac{a}{c_s^2} w_i \boldsymbol{\xi}_i \cdot \mathbf{F}^{(1)} + \frac{b}{2c_s^4} w_i Q_i : (\mathbf{F}^{(1)}\mathbf{u} + \mathbf{u}\mathbf{F}^{(1)}), \\ C_i^{(1)}(\mathbf{x}, t) = \frac{c}{2c_s^4} w_i Q_i : \delta\Pi^{(1)}. \end{cases} \quad (2.5.29)$$

In order to determine what the coefficients should be, let's examine the mass conservation law, the momentum conservation law and beyond for the higher order moments.

For the zeroth order moment $\boldsymbol{\xi}^0$, we have

$$\begin{aligned} 0 &= \sum_{i=0}^{q-1} \Omega_i = -\frac{1}{\tau} \sum_{i=0}^{q-1} \left(f_i(\mathbf{x}, t) - f_i^{(0)}(\mathbf{x}, t) \right) + \sum_{i=0}^{q-1} F_i(\mathbf{x}, t) + \sum_{i=0}^{q-1} C_i(\mathbf{x}, t) \\ &= -\frac{1}{\tau} (\rho - \rho) + \underbrace{\frac{a}{c_s^2} \sum_{i=0}^{q-1} w_i \boldsymbol{\xi}_i \cdot \mathbf{F}}_{\text{Term 1}} + \underbrace{\frac{b}{2c_s^4} \sum_{i=0}^{q-1} w_i Q_i : (\mathbf{F}\mathbf{u} + \mathbf{u}\mathbf{F})}_{\text{Term2}} \\ &\quad + \underbrace{\frac{c}{2c_s^4} \sum_{i=0}^{q-1} w_i Q_i : \delta\Pi}_{\text{Term3}} = 0, \end{aligned} \quad (2.5.30)$$

where Term 1 = Term 2 = Term 3 = 0 because of the lattice symmetries (2.5.21).

For the zeroth order moment $\boldsymbol{\xi}^1$, we have

$$\begin{aligned}
\mathbf{F} &= \sum_{i=0}^{q-1} \boldsymbol{\xi} \Omega_i = -\frac{1}{\tau} \sum_{i=0}^{q-1} (\boldsymbol{\xi} f_i(\mathbf{x}, t) - \boldsymbol{\xi} f_i^{(0)}(\mathbf{x}, t)) + \sum_{i=0}^{q-1} \boldsymbol{\xi} F_i(\mathbf{x}, t) + \sum_{i=0}^{q-1} \boldsymbol{\xi} C_i(\mathbf{x}, t) \\
&= -\frac{1}{\tau} \underbrace{(\mathbf{j} - \mathbf{j}^{(0)})}_{\text{by (2.5.11)} = -\mathbf{F}/2} + \underbrace{\frac{a}{c_s^2} \sum_{i=0}^{q-1} w_i \boldsymbol{\xi} \boldsymbol{\xi}_i \cdot \mathbf{F}}_{\text{Term1}} + \underbrace{\frac{b}{2c_s^4} \sum_{i=0}^{q-1} w_i \boldsymbol{\xi} Q_i : (\mathbf{F} \mathbf{u} + \mathbf{u} \mathbf{F})}_{\text{Term2}} \\
&\quad + \underbrace{\frac{c}{2c_s^4} \sum_{i=0}^{q-1} w_i \boldsymbol{\xi} Q_i : \delta \Pi}_{\text{Term3}} = (a + \frac{1}{2\tau}) \mathbf{F},
\end{aligned} \tag{2.5.31}$$

where $\text{Term1} = c_s^2 I$, $\text{Term2} = \text{Term3} = 0$ because of lattice symmetries (2.5.21).

From (2.5.31) we have for the first coefficient $a = 1 - 1/(2\tau)$.

With the definition $\Pi = \sum_{i=0}^{q-1} Q_i f_i$, we have that $\Pi^{(1)} = \sum_{i=0}^{q-1} Q_i f_i^{(1)}$ holds on the first order of expansion. It is left to compute $f_i^{(1)}$ in order to obtain $\Pi^{(1)}$ for (2.5.23). By the expansion of distribution function and operator, we have

$$\varepsilon \Omega_i^{(1)} + O(\varepsilon^2) = \Omega_i = -\frac{1}{\tau} \varepsilon f_i^{(1)} + \varepsilon F_i^{(1)} + \varepsilon C_i^{(1)} + O(\varepsilon^2), \tag{2.5.32}$$

together with (2.5.8)₁ and equilibrium distribution function (2.4.52) we have

$$\begin{aligned}
f_i^{(1)} &= -\tau \Omega_i^{(1)} + \tau (F_i^{(1)} + C_i^{(1)}) = -\tau (\partial_{t_1} + \nabla_1 \cdot \boldsymbol{\xi}_i) f_i^{(0)} + \tau (F_i^{(1)} + C_i^{(1)}) \\
&= -w_i \tau (\partial_{t_1} + \nabla_1 \cdot \boldsymbol{\xi}_i) \left(\rho + \frac{\boldsymbol{\xi}_i \cdot (\rho \mathbf{u})}{c_s^2} + \frac{1}{2c_s^4} Q_i : \rho \mathbf{u} \mathbf{u} \right) + \tau (F_i^{(1)} + C_i^{(1)}) \\
&= -w_i \tau \left(\partial_{t_1} \rho + \frac{1}{c_s^2} \partial_{t_1} (\boldsymbol{\xi}_i \cdot \rho \mathbf{u}) + \frac{1}{2c_s^4} \partial_{t_1} (Q_i : \rho \mathbf{u} \mathbf{u}) \right. \\
&\quad \left. + \boldsymbol{\xi}_i \cdot \nabla_1 \rho + \frac{1}{c_s^2} \boldsymbol{\xi}_i \boldsymbol{\xi}_i : \nabla_1 (\rho \mathbf{u}) + \frac{1}{2c_s^4} (\boldsymbol{\xi}_i \cdot \nabla_1) (Q_i : \rho \mathbf{u} \mathbf{u}) \right) + \tau (F_i^{(1)} + C_i^{(1)}).
\end{aligned} \tag{2.5.33}$$

For the three temporal derivatives, by virtue of continuity equation (2.5.13) and momentum equation (2.5.15) as well as the assumption of weak compressibility, i.e. $\partial_t \rho =$

$O(Ma)$ we have

$$\begin{aligned}
 & \text{by (2.5.13), } \partial_{t_1} \rho = -\nabla_1 \cdot (\rho \mathbf{u}), \\
 & \text{by (2.5.15), } \frac{1}{c_s^2} \partial_{t_1} (\boldsymbol{\xi}_i \cdot \rho \mathbf{u}) = \frac{1}{c_s^2} \boldsymbol{\xi}_i \cdot F^{(1)} - \frac{1}{c_s^2} \boldsymbol{\xi}_i \nabla_1 : \rho \mathbf{u} \mathbf{u} - \boldsymbol{\xi}_i \cdot \nabla_1 \rho, \\
 & \text{by (2.5.15), } \frac{1}{2c_s^4} \partial_{t_1} (Q_i : \rho \mathbf{u} \mathbf{u}) = \frac{1}{2c_s^4} Q_i : \left(\partial_{t_1} (\rho \mathbf{u}) \mathbf{u} + \mathbf{u} \partial_{t_1} (\rho \mathbf{u}) - \underbrace{\partial_{t_1} (\rho) \mathbf{u} \mathbf{u}}_{O(Ma^{1+2})} \right) \\
 & \quad = \frac{1}{c_s^4} Q_i : \partial_{t_1} (\rho \mathbf{u}) \mathbf{u} \quad (Q_i \text{ is symmetric }) \\
 & \quad = \frac{1}{c_s^4} Q_i : \left(F^{(1)} - \underbrace{\nabla_1 \cdot (\rho \mathbf{u} \mathbf{u})}_{O(Ma^2)} - c_s^2 \nabla_1 \rho \right) \underbrace{\mathbf{u}}_{O(Ma)} \\
 & \quad = \frac{1}{c_s^4} Q_i : \left(F^{(1)} - c_s^2 \nabla_1 \rho \right) \mathbf{u}. \tag{2.5.34}
 \end{aligned}$$

By substituting the three temporal derivatives into (2.5.33), neglecting the terms of $O(Ma^3)$, the distribution function on the scale of first order of Knudsen number becomes

$$\begin{aligned}
 f_i^{(1)} &= -\frac{w_i \tau}{c_s^2} \left(Q_i : \rho \nabla_1 \mathbf{u} - \boldsymbol{\xi}_i \nabla_1 : \rho \mathbf{u} \mathbf{u} + \frac{1}{2c_s^2} (\boldsymbol{\xi}_i \cdot \nabla_1) (Q_i : \rho \mathbf{u} \mathbf{u}) \right) \\
 &\quad - \frac{3w_i \tau}{2c_s^2} \boldsymbol{\xi}_i \cdot \mathbf{F}^{(1)} + \frac{(b-1)w_i \tau}{2c_s^4} Q_i : (\mathbf{F}^{(1)} \mathbf{u} + \mathbf{u} \mathbf{F}^{(1)}) + \frac{cw_i \tau}{2c_s^4 \varepsilon} Q_i : \delta \Pi. \tag{2.5.35}
 \end{aligned}$$

Therefore, with the lattice symmetries (2.5.21) we obtain the explicit expression for $\Pi^{(1)}$ as

$$\Pi^{(1)} = \sum_{i=0}^{q-1} Q_i f_i^{(1)} = -\frac{c_s^2 \tau}{\varepsilon} \rho (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + (b-1) \tau (\mathbf{F}^{(1)} \mathbf{u} + \mathbf{u} \mathbf{F}^{(1)}) + \frac{c \tau}{\varepsilon} \delta \Pi. \tag{2.5.36}$$

From the calculation of (2.5.34)₃, we can also obtain by (2.5.13)

$$\partial_{t_1} \Pi^{(0)} = \partial_{t_1} (\rho \mathbf{u} \mathbf{u}) = \mathbf{F}^{(1)} \mathbf{u} + \mathbf{u} \mathbf{F}^{(1)} - c_s^2 ((\nabla_1 \rho) \mathbf{u} + \mathbf{u} (\nabla_1 \rho)). \tag{2.5.37}$$

$\varepsilon * (2.5.36) + \varepsilon * (2.5.37) / 2$ leads to

$$\begin{aligned}
 \varepsilon \Pi^{(1)} + \frac{\varepsilon}{2} \partial_{t_1} \Pi^{(0)} &= \left(\underbrace{\tau(b-1) + \frac{1}{2}}_{\text{Term1}} \right) (\mathbf{F} \mathbf{u} + \mathbf{u} \mathbf{F}) - c_s^2 \tau \rho (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \\
 &\quad - \frac{1}{2} c_s^2 ((\nabla \rho) \mathbf{u} + \mathbf{u} (\nabla \rho)) + c \tau \delta \Pi. \tag{2.5.38}
 \end{aligned}$$

Taking $b = 1 - 1/(2\tau)$ so as to eliminate Term1, substituting (2.5.38) into (2.5.23), we get

$$\begin{aligned}
 \mathbf{F} &= \partial_t(\rho\mathbf{u}) + \nabla \cdot \left(\rho\mathbf{u}\mathbf{u} + c_s^2\rho I - c_s^2\tau\rho (\nabla\mathbf{u} + (\nabla\mathbf{u})^T) + c\tau\delta\Pi \right) \\
 &\quad + \nabla \cdot \left(\frac{1}{2}c_s^2 (\nabla(\rho\mathbf{u}) + (\nabla(\rho\mathbf{u}))^T) - \frac{1}{2}c_s^2((\nabla\rho)\mathbf{u} + \mathbf{u}(\nabla\rho)) \right) \\
 &\quad + \nabla \cdot \left(\frac{1}{2}c_s^2 \left(\underbrace{\partial_t\rho + \nabla \cdot (\rho\mathbf{u})}_{\text{continuity equation } = 0} - \varepsilon^2 \underbrace{\partial_{t_2}\rho}_{O(Ma)} \right) I \right) \\
 &= \partial_t(\rho\mathbf{u}) + \nabla \cdot \left(\rho\mathbf{u}\mathbf{u} + c_s^2\rho I - c_s^2(\tau - \frac{1}{2})\rho (\nabla\mathbf{u} + (\nabla\mathbf{u})^T) + c\tau\delta\Pi \right) + O(\varepsilon^2 Ma) \\
 &= \partial_t(\rho\mathbf{u}) + \nabla \cdot \left(\rho\mathbf{u}\mathbf{u} + c_s^2\rho I - \boldsymbol{\tau} \right) (\text{with accuracy order up to } \varepsilon^2 \times Ma),
 \end{aligned} \tag{2.5.39}$$

where we denote

$$\boldsymbol{\tau} = 2\nu\rho S - c\tau\delta\Pi, \quad S = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T), \quad \nu = c_s^2(\tau - \frac{1}{2}). \tag{2.5.40}$$

If $c = 0$, we recover by (2.5.39) and (2.5.40) the same Navier-Stokes equations as in (2.4.37). As the side product, we obtain the force term as

$$F_i = w_i \left(1 - \frac{1}{2\tau} \right) \left(\frac{\boldsymbol{\xi}_i \cdot \mathbf{F}}{c_s^2} + \frac{Q_i : (\mathbf{F}\mathbf{u})}{c_s^4} \right), \tag{2.5.41}$$

which is the same as that in Grad's representation approach (2.4.74) given that $\mathbf{F} = \rho\mathbf{g}$.

Equation (2.5.39) is the Navier-Stokes equations with dynamic shear viscosity but not bulk viscosity. To retrieve the dynamic bulk viscosity, the correction term C_i has to be taken into account. Suppose the correction term as

$$\delta\Pi = \frac{\varepsilon}{D} \left(Tr(\Pi^{(1)}) + k\mathbf{F}^{(1)} \cdot \mathbf{u} \right) I, \tag{2.5.42}$$

where D is the dimension of the system, $Tr(\Pi^{(1)})$ is the trace of the tensor $\Pi^{(1)}$ and k is a coefficient to be determined. In order to have $Tr(\Pi^{(1)})$, taking the trace of (2.5.36) yields,

$$\begin{aligned}
 Tr(\Pi^{(1)}) &= \frac{c\tau}{\varepsilon} Tr(\delta\Pi) - 2c_s^2\tau\rho\nabla_1 \cdot \mathbf{u} - \mathbf{F}^{(1)} \cdot \mathbf{u} \\
 &= \frac{c\tau}{\varepsilon} \left(Tr(\Pi^{(1)}) + k\mathbf{F}^{(1)} \cdot \mathbf{u} \right) - 2c_s^2\tau\rho\nabla_1 \cdot \mathbf{u} - \mathbf{F}^{(1)} \cdot \mathbf{u}.
 \end{aligned} \tag{2.5.43}$$

From (2.5.43), we have

$$Tr(\Pi^{(1)}) = \frac{1}{1 - c\tau} \left((c\tau k - 1)\mathbf{F}^{(1)} \cdot \mathbf{u} - 2c_s^2\tau\rho\nabla_1 \cdot \mathbf{u} \right). \tag{2.5.44}$$

Hence, the partial correction term turns out to be

$$c\tau\delta\Pi = \frac{\varepsilon}{1-c\tau} \frac{c\tau}{D} \left(\underbrace{((1+c\tau)k-1)}_{\text{Term1}} \mathbf{F}^{(1)} \cdot \mathbf{u} - 2c_s^2\tau\rho\nabla_1 \cdot \mathbf{u} \right) I. \quad (2.5.45)$$

When

$$k = 1/(1+c\tau) \quad (2.5.46)$$

Term1 vanishes, and choose c as

$$c = \frac{1}{\tau - 2c_s^2\tau^2/(d\nu')}, \quad (2.5.47)$$

we have eventually

$$c\tau\delta\Pi = \rho\nu' \nabla \cdot \mathbf{u} I, \quad (2.5.48)$$

which is responsible for the dynamic bulk viscosity in (2.5.39) such that

$$\boldsymbol{\tau} = -\rho\nu' (\nabla \cdot \mathbf{u}) I + 2\rho\nu S. \quad (2.5.49)$$

As a result, the full correction term is

$$\begin{aligned} C_i &= \frac{cw_i}{2Dc_s^4} \left(Tr(\varepsilon\Pi^{(1)}) + k\varepsilon \mathbf{F}^{(1)} \cdot \mathbf{u} \right) (Q_i : I) \\ &= \frac{cw_i}{2Dc_s^4} \left(Tr(\Pi - \Pi^{(0)}) + k\mathbf{F} \cdot \mathbf{u} \right) (\xi_i^2 - c_s^2 D) + O(\varepsilon^2) \\ &= \frac{cw_i}{2Dc_s^4} \left(Tr \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i f_i - \rho \mathbf{u} \mathbf{u} - \rho c_s^2 I \right) + k\mathbf{F} \cdot \mathbf{u} \right) (\xi_i^2 - c_s^2 D) + O(\varepsilon^2) \\ &= \frac{cw_i}{2Dc_s^4} \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i^2 f_i - \rho \mathbf{u}^2 - \rho c_s^2 d + k\mathbf{F} \cdot \mathbf{u} \right) (\xi_i^2 - c_s^2 D) + O(\varepsilon^2) \end{aligned} \quad (2.5.50)$$

with k and c given in (2.5.46) and (2.5.47). Equation (2.5.50) shows that the correction term finds itself up to the scale of the first order of the Knudsen number.

In summary, Chapman-Enskog approach is based on the asymptotic multi-scale analysis of the lattice Boltzmann equation and sheds light on the second order accuracy with respect to Mach number of retrieving Navier-Stokes equations up to the scale of the first order of Knudsen number. In order to obtain the continuity equation for mass conservation and the momentum equation for momentum conservation as in the full Navier-Stokes equations, the following three main steps are carried out. First of all, expand the temporal and spatial variables up to the second order of Knudsen number while the distribution function, density as well as momentum up to the first order of

Knudsen number. Secondly, by examining the lattice Boltzmann - BGK equation on the scale of zeroth order, first order and second order of Knudsen number with the truncated equilibrium distribution to the second order of Mach number, we have the exact continuity equation corresponding to mass conversation and momentum conservation for the Navier-Stokes equations. However the stress tensor for Navier-Stokes equations is unknown on the scale of first order of Knudsen number. Finally, by ansatz of the collision operator, including the force term and the correction term for dynamic bulk viscosity, we succeed in closing the full Navier-Stokes equations with not only the shear viscosity but also the desirable bulk viscosity from appropriate choice of force term and correction term.

2.6 Lattice Boltzmann Method

2.6.1 Gauss-Hermite quadrature and lattice structure

Both the discretization of velocity for the integral in the velocity space by Grad's representation approach and the lattice structure for the lattice Boltzmann - BGK equation by Chapman-Enskog expansion approach are related directly to the discrete velocity in direction and magnitude as well as the weights for the discrete velocity. Different discretization of velocity in various dimensions results in distinct lattice structure for lattice Boltzmann - BGK model with certain order of accuracy to approximate the Boltzmann equation. The principle to choose efficient lattice structure is to achieve as high order of accuracy as possible with fixed number of discrete velocities or in another way, to choose as less the number of discrete velocities as possible with fixed order of accuracy. However, the location of the discrete velocities in the lattice should be convenient for streaming the distribution function as well. Therefore, the preferable lattice structures are not the ones with the least number of discrete velocities but the ones with discrete velocities lying on the lattice nodes.

The fundamental theorem [45] claims that one optimal quadrature rule to achieve as high order of accuracy as possible with certain fixed number of abscissae for one dimensional integral is the Gaussian quadrature. It provides an algebraic degree of precision of $2q - 1$ with only q quadrature nodes, reading as

$$\int_a^b w(\xi) f(\xi) d\xi \approx \sum_{i=1}^{q-1} w_i f(\xi_i) \quad (2.6.1)$$

with exact discretization for polynomials with algebraic degree of precision up to $2q - 1$. The abscissae are the roots of the q th corresponding orthogonal polynomial $f^{(q)}(\xi)$, with the weights given as

$$w_i = \frac{\int_a^b w(\xi) f^{(q-1)}(\xi) f^{(q-1)}(\xi) d\xi}{f^{(q-1)}(\xi_i) df^{(q)}(\xi_i)/d\xi} \quad (2.6.2)$$

For one dimensional q th order Hermite polynomials, we have for the so called Gauss-Hermite quadrature rule that the q abscissae are the roots of $\mathcal{H}^{(q)}(\xi)$ and the weights from (2.4.9) and (2.4.7) in one dimension are

$$w_i = \frac{q!}{(q\mathcal{H}^{(q-1)}(\xi_i))^2} \quad (2.6.3)$$

Denoting by $E_{D,m}^d$ the quadrature formula, by d the number of nodes for the quadrature, by D the dimension of space and by m the algebraic degree of precision, we have for one dimension the quadrature formulas as in Table 2.6.1.

Table 2.6.1: One-dimensional Gauss-Hermite quadrature formula

Quadrature	ξ_i	w_i
$E_{1,1}^1$	0	1
$E_{1,3}^2$	± 1	$1/2$
$E_{1,5}^3$	0	$2/3$
	$\pm\sqrt{3}$	$1/6$

No general Gauss quadrature formulas are known for integral in high dimensional spaces. However, by the so-called production formula we can derive a class of high dimensional Gauss quadrature formulas from the one-dimensional formula. In particular, suppose we have the D dimensional polynomial of degree q in the following form

$$f(\boldsymbol{\xi}) = \sum_{q_1 + \dots + q_D \leq q} a_{q_1 q_2 \dots q_D} \prod_{i=1}^D \xi_i^{q_i} \quad (2.6.4)$$

and the integral under our consideration for Hermite polynomials holds for the following relation (see [29])

$$\frac{1}{(2\pi)^{D/2}} \int \exp\left(-\frac{\boldsymbol{\xi}^2}{2}\right) f(\boldsymbol{\xi}) d\boldsymbol{\xi} = \sum_{k_1=0}^{q-1} \dots \sum_{k_D=0}^{q-1} w_{k_1} \dots w_{k_D} f(\boldsymbol{\xi}_{k_1, \dots, k_D}), \quad (2.6.5)$$

where $\boldsymbol{\xi}_{k_1, \dots, k_D} = (\xi_{k_1}^{q_1}, \dots, \xi_{k_D}^{q_D})$. From this relation, it is possible to derive 2D and 3D Gauss-Hermite quadrature formulas as $E_{2,5}^9$ and $E_{3,5}^{27}$ respectively, as shown in Figure A.0.3 and A.0.5. More general formulas in 2D and 3D are given in Table 2.6.2 and 2.6.3 respectively. It's worth to mention that $E_{2,5}^9, E_{3,5}^{15}, E_{3,5}^{19}$ are widely used in 2D.

Remark 2.6.1 As one notice that though the grid points for the three dimensional Gauss Hermite quadrature is 27 the precision of the quadrature rule is only still of algebraic degree 5. In order to reduce the number of quadrature points, we denote by $w_{(0,0,0)}$ for the abscissa $(0,0,0)$ and $w_{(1,1,1)}$ for $(\sqrt{3}, \sqrt{3}, \sqrt{3})$ and so on for the other abscissas. Given that $q_1 + q_2 + q_3 \leq 5$ and $q_1 \leq q_2 \leq q_3$ without loss of generality, we have that $q_1 = 0$ since $q_1 \leq 1$ and for $q_1 = 1$ the integral in (2.6.5) is 0, which means that the three dimensional Gauss-Hermite quadrature $E_{3,5}^{27}$ degenerates to the two dimensional one with the same precision degree provided

$$\begin{aligned} w_{(0,0,0)} + 2w_{(0,0,1)} &= 4/9 \\ w_{(0,0,1)} + 2w_{(0,1,1)} &= 1/9 \\ w_{(0,1,1)} + 2w_{(1,1,1)} &= 1/36 \end{aligned} \quad (2.6.6)$$

Table 2.6.2: Two-dimensional Gauss-Hermite quadrature formula ($(\sqrt{3}, 0)_S$ are the full symmetric points and the same for that in three dimensional quadratures)

Quadrature	ξ_i	Group	p	w_i
$E_{2,4}^6$	(0, 0)	A	1	1/2
	$2(\cos(2n\pi/5), \sin(2n\pi/5))$		5	1/10
$E_{2,5}^7$	(0, 0)	A	1	1/2
	$2(\cos(n\pi/3), \sin(n\pi/3))$		6	1/12
$E_{2,5}^9$	(0, 0)	A	1	4/9
	$\sqrt{3}(1, 0)_S$		4	1/9
	$\sqrt{3}(\pm 1, \pm 1)$		4	1/36

Table 2.6.3: Three-dimensional Gauss-Hermite quadrature formula

Quadrature	ξ_i	Group	p	w_i
$E_{3,5}^{13}$	(0, 0, 0)	A	1	2/5
	$(\pm r, \pm s, 0)$		4	$1/20$ $r^2 = (5 + \sqrt{5})/2$
	$(0, \pm r, \pm s)$		4	$1/20$ $s^2 = (5 - \sqrt{5})/2$
	$(\pm s, 0, \pm r)$		4	$1/20$
$E_{3,5}^{15}$	(0, 0, 0)	A	1	2/9
	$\sqrt{3}(1, 0, 0)_S$		6	1/9
	$\sqrt{3}(\pm 1, \pm 1, \pm 1)$		8	1/72
$E_{3,5}^{19}$	(0, 0, 0)	A	1	1/3
	$\sqrt{3}(1, 0, 0)_S$		6	1/18
	$\sqrt{3}(1, 1, 0)_S$		12	1/36
$E_{3,5}^{27}$	(0, 0, 0)	A	1	8/27
	$\sqrt{3}(0, 0, 0)_S$		6	2/27
	$\sqrt{3}(1, 1, 0)_S$		12	1/54
	$\sqrt{3}(\pm 1, \pm 1, \pm 1)$		8	1/216

where the weights $w_{a,b,c}$ are the same when $a+b+c$ is the same because of the symmetric property of the weights. The parametric solution for (2.6.6) reading as

$$\begin{pmatrix} w_{(0,0,0)} \\ w_{(0,0,1)} \\ w_{(0,1,1)} \\ w_{(1,1,1)} \end{pmatrix} = \frac{1}{72} \begin{pmatrix} 8(2-t) \\ 4(t-2) \\ 2t \\ 1-t \end{pmatrix} \quad (2.6.7)$$

from which $E_{3,5}^{15}$ (see Figure A.0.4) and $E_{3,5}^{19}$ (see Figure A.0.5) are constructed by setting $t = 0$ and $t = 1$, while $E_{3,5}^{27}$ corresponds to $t = 2/3$. Another three interesting quadratures were given by Stoud [29] as $E_{2,4}^6$, $E_{2,5}^7$ and $E_{3,5}^{13}$. And they turned out to be the least number of nodes for corresponding algebraic degree of precision. For general hierarchical models, see [40].

2.6.2 Algorithm for lattice Boltzmann method

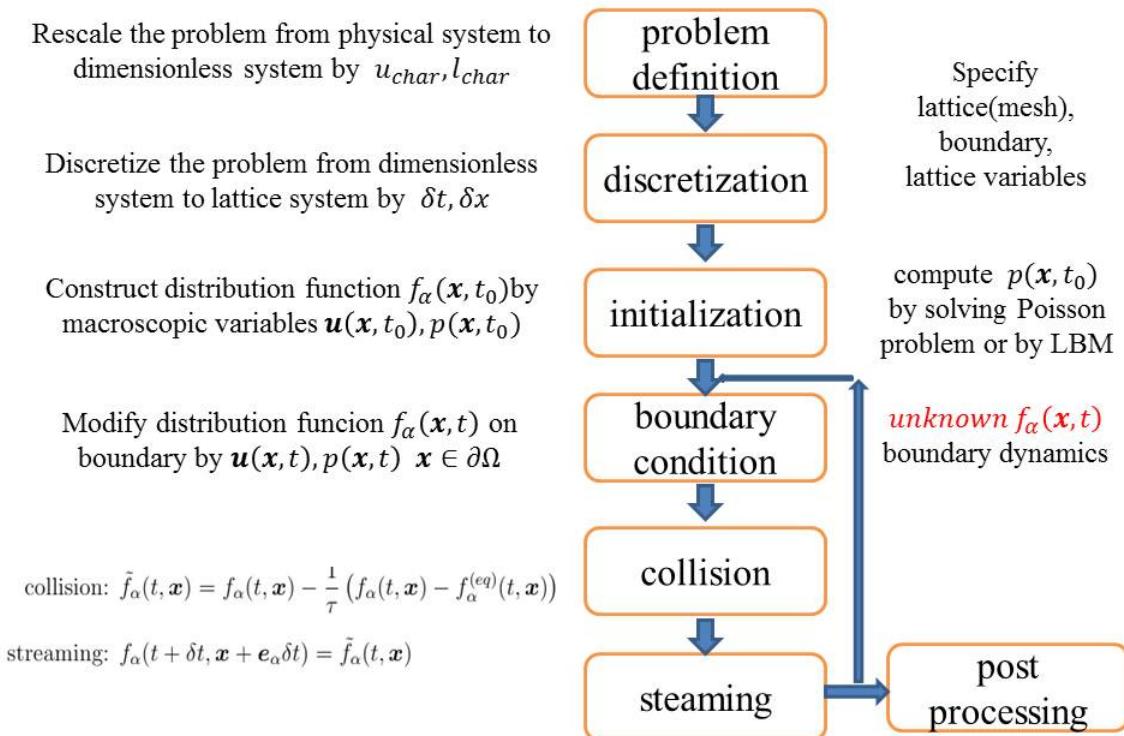


Figure 2.6.1: general algorithm for lattice Boltzmann method

Once the lattice structure is specified, the lattice Boltzmann - BGK model is determined. And the general algorithm can be given as follows: (flow chart is shown in Figure 2.6.1)

1. In order to have a consistent problem definition for lattice Boltzmann method, we first rescale the physical problem from physical system to dimensionless system by characteristic length l_0 and velocity u_0 . (This will be discussed in the next section).

2. In the dimensionless system, mesh in the form of uniform grid in Cartesian coordinate is generated by spatial step δx according to the geometry of the computational domain. The time step δt is also chosen according to δx with the restriction that the Mach number $Ma := (\delta t / \delta x) / c_s < 1$. (A common choice is $\delta t = O(\delta x^2)$).
3. In order to initialize distribution functions, not only lattice structure should be specified, initial velocity as well as initial pressure are also required. However, pressure is not always given for initial conditions so that, in order to get it, we have to solve Poisson problem or apply particular techniques based on lattice Boltzmann method, namely the multiple relaxation time (MRT) and multigrid (MG) lattice Boltzmann method. For the latter an overview will be given in section 4.5.2
4. Enter the loop for evolution. Impose boundary conditions at the macroscopic level and construct the unknown distribution functions streaming from the solid to the bulk on the kinetic level based on the imposed boundary conditions. Different boundary dynamics are chosen according to the requirement of stability, accuracy and efficiency, investigated in Chapter 3.
5. Collision according to (2.4.76)₁. Collision are the same for nodes sitting both in the inner region and on the boundary.
6. Streaming according to (2.4.76)₂. Stream in all directions for the nodes in the inner region and the directions pointing inside the boundary.
7. Stop and go to the step of post processing if certain stopping condition is satisfied. Otherwise go back to step 4 of imposing boundary condition.

In the next two sections, we will briefly discuss two critical aspects that has to be taken into account when lattice Boltzmann simulation are performed. The first one in section 2.6.3 is related to how to move from the physical system (SI units) to the lattice Boltzmann system. The second on in section 2.6.4 deals with the issue of the compressibility effect.

2.6.3 Physical, dimensionless and lattice Boltzmann system

In order to simulate fluid in physical system by lattice Boltzmann method, proper treatment of the physical problem for discretization into the lattice Boltzmann system, in which the evolution of lattice Boltzmann model is carried out, remains untouched. One can either start directly from the physical problem with dimensional units to the

lattice Boltzmann system by discretization, or firstly rescale the physical problem into dimensionless system by characteristic time t_0 and length l_0 with dimensionless units, and then go to the lattice Boltzmann system by discretization time step δt and spatial step δx . In order to make the simulation consistent for different problem, and to get rid of physical units, it's better to rescale the problem from physical system to dimensionless system, in which we only care about values of variables and not their units. Denote with index x_p, x_d, x_{lb} for spatial variables in physical system, dimensionless system and lattice Boltzmann system respectively, we have (see [72]) the incompressible Navier-Stokes equations in physical system as

$$\begin{cases} \partial_{t_p} \mathbf{u}_p + (\mathbf{u}_p \cdot \nabla_p) \mathbf{u}_p = -\nabla_p p_p / \rho_{0p} + \nu_p \nabla_p^2 \mathbf{u}_p, \\ \nabla_p \cdot \mathbf{u}_p = 0. \end{cases} \quad (2.6.8)$$

Provided the rescaling parameters - characteristic time t_0 and characteristic length l_0 , the dimensionless variables corresponding to their physical counterpart and their derivatives are given as

$$t_d = \frac{t_p}{t_0}, \quad l_d = \frac{l_p}{l_0}; \quad \mathbf{u}_p = \frac{l_0}{t_0} \mathbf{u}_d; \quad \partial_{t_p} = \frac{1}{t_0} \partial_{t_d}; \quad \nabla_p = \frac{1}{l_0} \nabla_d, \quad p_p = \rho_0 \frac{l_0^2}{t_0^2} p_d, \quad (2.6.9)$$

so that the dimensionless Navier-Stokes equations become

$$\begin{cases} \partial_{t_d} \mathbf{u}_d + (\mathbf{u}_d \cdot \nabla_d) \mathbf{u}_d = -\nabla_d p_d + \nabla_d^2 \mathbf{u}_d / Re, \\ \nabla_d \cdot \mathbf{u}_d = 0, \end{cases} \quad (2.6.10)$$

where $Re = l_0^2 / (t_0 \nu_p)$ is Reynolds number and the dimensionless viscosity $\nu_d = 1/Re$. Take space interval δx and time interval δt as the characteristic length and time for discretization. Similar to the renormalization from physical system to dimensionless system, we have for the discretization that

$$\mathbf{u}_d = \frac{\delta x}{\delta t} \mathbf{u}_{lb}, \quad \nu_d = \frac{\delta x^2}{\delta t} \nu_{lb} \Rightarrow \mathbf{u}_{lb} = \frac{\delta t}{\delta x} \mathbf{u}_d, \quad \nu_{lb} = \frac{\delta t}{\delta x^2} \nu_d = \frac{\delta t}{\delta x^2} \frac{1}{Re}. \quad (2.6.11)$$

As shown in (2.5.40), the relaxation time is related to the viscosity by

$$\tau = \frac{\nu_{lb}}{c_s^2} + \frac{1}{2} = 3\nu_{lb} + 0.5 \text{ for } D2Q9, D3Q15, D3Q19, D3Q27, \quad (2.6.12)$$

from which we can see that the relaxation time satisfy $\tau \in (0.5, \infty)$. The restriction for the discretization parameter $\delta t, \delta x$ is

$$\|\mathbf{u}_{lb}\|_2 = (\delta t / \delta x) \|\mathbf{u}_d\|_2 \ll c_s,$$

where $\|\mathbf{u}_d\|_2$ is always chosen to be 1 as l_0, t_0 are selected such that $\|\mathbf{u}_p\|_2 = l_0/t_0$. This restriction comes from that Mach number should be small enough, at least smaller than the lattice sound velocity [43].

2.6.4 Compressibility effect of lattice Boltzmann method

As shown in section 2.4, the system described by the lattice Boltzmann model is given by the simplified Navier-Stokes equations (2.4.37). The continuity equation (2.4.37)₁ introduces a compressibility effect which become critical for the simulation of the incompressible fluids. In order to reduce the compressibility effect for the time dependent flows, we need the following incompressible lattice Boltzmann method, a slight modification of the compressible one with respect to the density. Firstly, the equilibrium distribution function is modified as

$$f_i^{(eq)} = f_i^{(0)}(\mathbf{x}, t) = w_i \left(\rho + \rho_0 \left(\frac{\boldsymbol{\xi}_i \cdot \mathbf{u}}{c_s^2} + \frac{1}{2c_s^4} Q_i : \mathbf{u}\mathbf{u} \right) \right) \quad (2.6.13)$$

where ρ_0 is a constant corresponding to the density. For the non-equilibrium distribution ,we have also that

$$f_i^{(neq)} = -\frac{w_i \tau}{c_s^2} \left(Q_i : \rho_0 \nabla \mathbf{u} - \boldsymbol{\xi}_i \nabla : \rho_0 \mathbf{u}\mathbf{u} + \frac{1}{2c_s^2} (\boldsymbol{\xi}_i \cdot \nabla) (Q_i : \rho_0 \mathbf{u}\mathbf{u}) \right), \quad (2.6.14)$$

while to recover the velocity from distribution function, we have

$$\rho_0 \mathbf{u} = \sum_{i=0}^8 \boldsymbol{\xi}_i f_i. \quad (2.6.15)$$

Meanwhile, for all the boundary conditions that related to the item of $\rho g(\mathbf{u})$, they are given instead as $\rho_0 g(\mathbf{u})$, where $g(\mathbf{u})$ is the function of \mathbf{u} such as \mathbf{u} , \mathbf{u}^2 or $\mathbf{u}\mathbf{u}$. By the same Chapman-Enskog expansion procedure, we can obtain the incompressible Navier-Stokes equations accurate up to the second order of Mach number $O(Ma^2)$ for continuity equation and third order of Mach number $O(Ma^3)$ for momentum equations, as shown in (2.6.16). For more details, see [33].

$$\begin{cases} \partial_t \mathbf{u} - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mathbf{g} + O(Ma^3), \\ \nabla \cdot \mathbf{u} = 0 + O(Ma^2). \end{cases} \quad (2.6.16)$$

Notice that the above system is still biased from standard Navier-Stokes equations by the compressibility effect related to Mach number. In order to reduce this compressibility effect, small Mach number is preferable, which means that the time step δt has to be small with respect to the space step δx . From a practical point of view, time step and spatial step is usually chosen as $\delta t = O(\delta x^2)$ [42].

2.7 Conclusion

In this section, Euler equations and Navier-Stokes equations were derived from the kinetic Boltzmann - BGK equation by Grad's representation approach and Chapman-

Enskog expansion. Grad's representation approach of projecting the distribution function on Hermite polynomials and truncating it at certain order is relatively clear without loss of the accuracy up to the truncated order. Chapman-Enskog expansion and ansatz can also be used to analyze the accuracy of the approximation up to a certain order for the derivation from lattice Boltzmann - BGK equation to Navier-Stokes equation. What's more, by Chapman-Enskog ansatz we can also obtain independent bulk and shear viscosity for Navier-Stokes equations.

Gauss-Hermit quadrature formulas in 1D, 2D and 3D lead to the phase-space discretization of Boltzmann - BGK equation to lattice Boltzmann - BGK model, resulting in the lattice structures given in 1D, 2D and 3D. The full general algorithm for lattice Boltzmann method is given and compared to that of finite element method. In order to apply lattice Boltzmann method conveniently, we clarify the relation of variables and their derivatives as well as the equations between physical system, dimensionless system and lattice Boltzmann system. Meanwhile, we obtain the incompressible lattice Boltzmann method by reducing the compressibility effect.

Chapter 3

Boundary conditions for lattice Boltzmann method

3.1 Introduction

Boundary conditions play a very important role for conventional numerical methods, such as finite element method, to solve any differential system. In a similar way, imposing boundary conditions accurately is crucial for lattice Boltzmann method. In general, there are three distinct kinds of boundary conditions for macroscopic differential equations, namely Dirichlet boundary conditions which specify the value of the function on the boundary, Neumann boundary conditions that provide the normal derivative of the function on the boundary, and the combination of prescribed function and it's normal derivative with the name of Robin boundary conditions. When the boundary conditions are approximated on the boundary the same way as that within the domain by a certain conventional numerical method, the problem ends up with solving a linear or non linear algebraic system.

As for the conventional method, the macroscopic function imposed on the boundary is usually the same as the function directly treated by the method. Take velocity in Navier-Stokes equations for example, the prescribed velocity or pressure on the boundary is the unknown macroscopic variables to be solved by "macroscopic methods". For some aspects, lattice Boltzmann method is quite different from the conventional methods. More in detail, the leading role playing on the stage is not macroscopic variables but rather microscopic distribution function. Since there is no explicit distribution function (referred as unknown distribution function in the following) streaming from the boundary into the bulk, we have to construct the unknown distribution functions based on the macroscopic imposed boundary conditions, which is also one of the significant difficulties for lattice Boltzmann method. In this thesis, we will concentrate on how to impose Dirichlet boundary conditions by various boundary dynamics for lattice

Boltzmann method, while for Neumann and Robin boundary conditions, most of the same ideas can be applied but how to impose them in an appropriate way is still an interesting open problem.

Two distinct approaches are applied to address the difficulty of the boundary conditions for lattice Boltzmann method. One is the so called “Distribution Modification” while the other is “Distribution Reconstruction”. By distribution modification we mean that the unknown distribution functions are obtained by some physical rules such as bounce-back rule, mass and momentum conservation law or their combination (illustrated in detail later). The so called fullway bounce-back rule, halfway bounce-back rule [6, 47], as well as Inamuro boundary dynamics [49], Zou/He boundary dynamics [50] and so on belong to this school. On the other hand, distribution reconstruction stands for reconstructing all the distribution functions on the boundary by their explicit relation with the macroscopic variables - density, velocity and their derivatives, which can be approximated by the given boundary condition. Among the methods of this approach are regularized boundary dynamics [42], finite difference boundary dynamics [61] as well as the Guo et al. boundary dynamics [52]. Most of the aforementioned methods for imposing boundary conditions are developed originally for straight boundary - the boundary with straight edges.

While for curved boundary of the computational domain, which is more realistic from the physical point of view, interpolation or extrapolation is needed to approximate those variables sitting on the boundary nodes away from the boundary wall in the Cartesian grid. This differs from finite element method since the nodes in the mesh of finite element can be put on the curved boundary directly. Belonging to the “Distribution Modification” approach are the interpolating methods suggested by Fillipova et al. (FH dynamics) [53] and its improved version by Mei et al. (Mei dynamics) [54]. Two more simple fitting scheme were proposed later on by Bouzidi et al. (BFL dynamics) [55] and Yu et al. (uniform dynamics) [57] respectively. While in the family of “Distribution Reconstruction” comes the recently developed regularized curved boundary dynamics by Verschaeve et al. [58].

In section 3.2 the basic ideas of bounce-back rule, conservation law as well reconstruction for straight boundary are investigated for the bounce-back dynamics, Zou-He dynamics, regularized dynamics as well as finite difference dynamics. Poiseuille flow, lid driven cavity flow and Womersley flow are taken as benchmarks for the verification, error and efficiency analysis of these straight boundary dynamics in section 3.3. In section 3.4, differences and difficulties for imposing boundary conditions on curved boundary are first illustrated. As for the difficulties, we will examine different boundary dynamics including FH and Mei dynamics, BFL dynamics, uniform dynamics with interpolation and extrapolation as well as the stair-wise halfway bounce back dynam-

ics without any interpolation or extrapolation. Meanwhile, the modified regularized dynamics will also be applied for curved boundary. Finally, the Taylor-Couette flow is tested for the verification and error analysis of these boundary dynamics for curved boundary, in section 3.5.

3.2 Straight boundary conditions

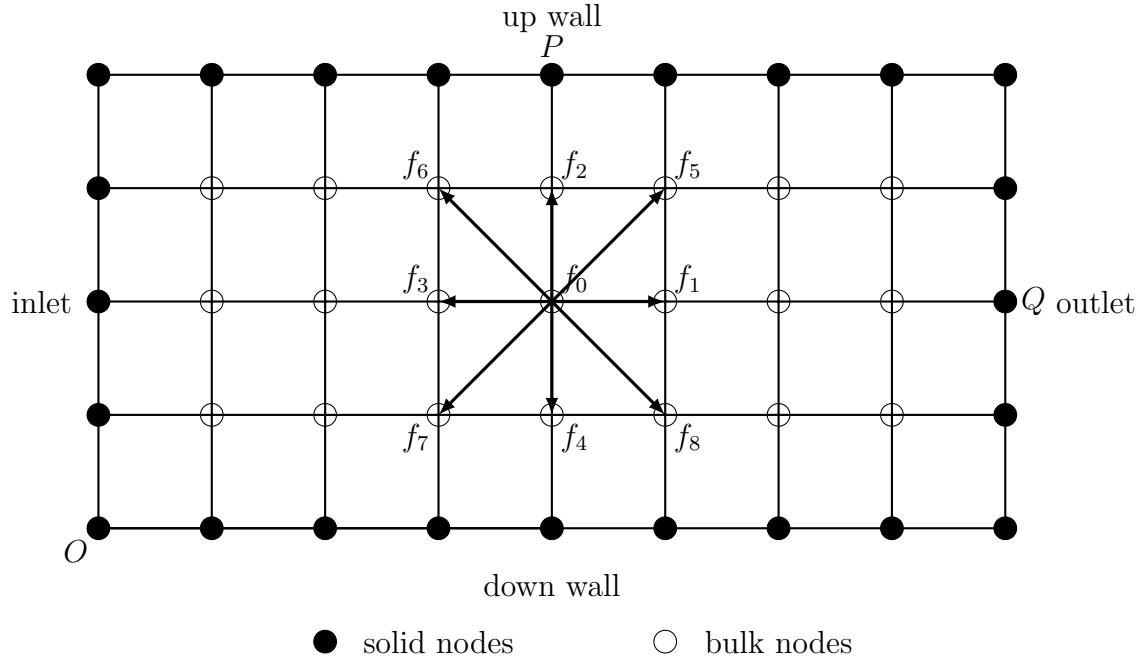


Figure 3.2.1: sketch for straight boundary condition with indexed distribution function as well as solid nodes and bulk nodes indexed with solid points and hollow points

In order to illustrate the idea of different kinds of straight boundary condition, the D2Q9 (corresponding to $E_{2,5}^9$) lattice structure is taken as example for simplicity.

Consider the boundary node P in Figure 3.2.1. The distribution function f_4 , f_7 and f_8 coming from the solid are unknown since there is no particle streaming from the solid to the bulk. Unfortunately, we have only the prescribed macroscopic variables such as velocity and pressure for boundary conditions instead of the unknown distribution functions. Noticing that the density is related to pressure as $p = \rho c_s^2$ by the result for ideal gas dynamics where the lattice Boltzmann method are developed from, we see that once the pressure is provided, the density is determined and vice versa.

The theoretical analysis in section 2 for lattice Boltzmann - BGK model sheds light on its accuracy as a Navier-Stokes solver by the result of mass and momentum conservation law, corresponding to the collision invariance. Hence, the principle for

tackling boundary conditions is clear, which comes in twofolds. The one is that mass and momentum or even energy have to be conserved so as to be consistent on the boundary with that in the bulk region. Correspondingly, the other is that all the distribution functions on the boundary, which can be separated as equilibrium part related to density and velocity and non-equilibrium part represented also by their derivatives, are required to satisfy the conservation law.

According to the formula between mass, momentum and the distribution functions (2.5.2) and (2.5.3), we have for the boundary node P with unknowns f_4, f_7 and f_8 that

$$\begin{cases} \rho = f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8, \\ \rho u_x = f_1 - f_3 + f_5 - f_6 - f_7 + f_8, \\ \rho u_y = f_2 - f_4 + f_5 + f_6 - f_7 - f_8. \end{cases} \quad (3.2.1)$$

Given the Dirichlet boundary conditions - the prescribed velocity explicitly, we obtain from (3.2.1)₁ + (3.2.1)₃ that

$$\rho = \frac{1}{1 + u_y} (f_0 + f_1 + f_3 + 2(f_2 + f_5 + f_6)), \quad (3.2.2)$$

which leads to the density exactly determined by the boundary velocity and known distribution functions $f_0, f_1, f_2, f_3, f_5, f_6$. With proper modification we can also get the pressure for other boundaries corresponding to (3.2.2) from (3.2.1).

More generally, given on a straight boundary two of the three unknowns ρ, u_x, u_y , it is always possible to retrieve the third one by (3.2.1). As a last example, consider the node Q in Figure 3.2.1, and assume that ρ, u_y are given. Then the velocity u_x can be computed as

$$u_x = -1 + \frac{1}{\rho} (f_0 + f_2 + f_4 + 2(f_1 + f_5 + f_8)). \quad (3.2.3)$$

Remark 3.2.1 As for Dirichlet boundary conditions, either velocity or pressure can be given as known variables. In 2D, if the velocity $\mathbf{u} = (u_x, u_y)^T$ are prescribed, the density could be obtained from (3.2.1). While if the pressure is prescribed, ρ is known, but we still need one component of the velocity, u_x for the boundary node P or u_y for Q .

Remark 3.2.2 Special attention should be paid to the nodes at the corners since the number of the unknown distribution functions increases to five instead of three, for which neither the density or the velocity can be obtained from (3.2.2) or (3.2.3). Two different approaches are developed to get the density and velocity. The one from the macroscopic point of view is to use interpolation or extrapolation of the density or velocity on the boundary nodes from the values of their neighbors up to a certain order

³. Whereas the other is played in the kinetic world, interpolating or extrapolating the unknown distribution functions directly from known ones sitting on their neighbors. For sake of clarity, the corners will be considered in detail in benchmarks later on.

3.2.1 Bounce-back boundary dynamics

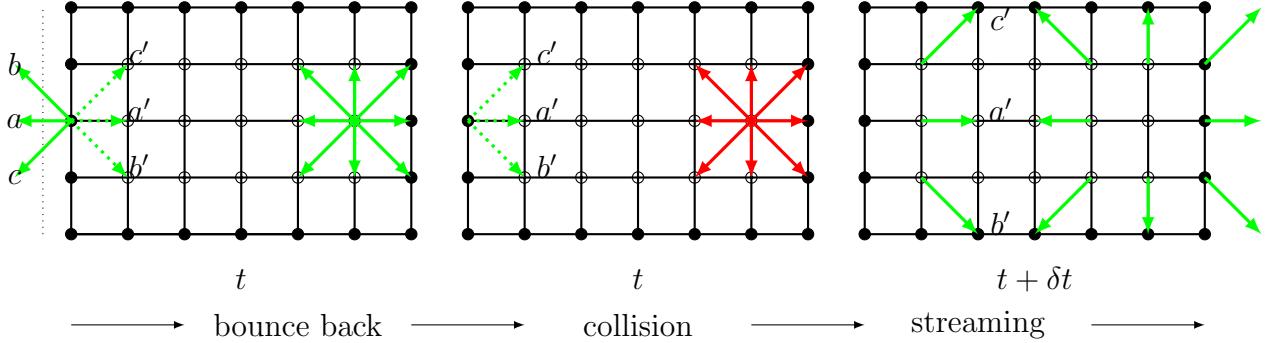


Figure 3.2.2: sketch for fullway bounce back for straight boundary conditions. On the left boundary, the arrows indicate bounce back of the distribution functions, while on the right part in the bulk region, the arrows indicate how distribution functions evolve in the bulk; red arrows stands for post collision distribution functions while green ones represents distribution functions after streaming step

The first boundary dynamics to impose boundary conditions we illustrate is the simplest one with name of fullway bounce back for non-slip boundary. As shown in Figure 3.2.2, the boundary nodes send back what streams into the solid in the opposite direction to the bulk during the collision step at the time t , and then pass this information to the bulk during the streaming step, after which the time is increased by δt . Meanwhile, the distribution functions near the boundary is streamed into the solid. Formally speaking, we have for the non-slip fullway bounce back boundary condition that

$$\begin{pmatrix} f_{a'} \\ f_{b'} \\ f_{c'} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_a \\ f_b \\ f_c \end{pmatrix} \quad (3.2.4)$$

As shown in Figure 3.2.2, the standard collision step is executed for distribution functions within the bulk region, but it is replaced by bounce back for distribution functions on the boundary. After which, the streaming step follows immediately. From (3.2.2) and (3.2.3), we can see that both mass and momentum are conserved by the bounce

³usually second order is enough since the lattice Boltzmann method is only accurate up to second order

back rule. As for the energy $2\rho\epsilon = \sum_{i=0}^8 \mathbf{c}_i^2 f_i$ and stress tensor $\Pi = \sum_{i=0}^8 \mathbf{c}_i \mathbf{c}_i f_i$, they are related to the second order moment of lattice velocity and can not be conserved since the unknown distribution functions are modified directly to those in the opposite or specular direction without taking the conservation of energy into account.

The fullway bounce back rule is simple but it is claimed in [50] that it achieves the first order spatial accuracy, limiting the advantage of lattice Boltzmann method, which essentially performs with the second order spatial accuracy. In order to achieve the second order accuracy, the halfway bounce back rule is introduced as follows. The difference between halfway bounce back rule and the full way bounce back is that the solid wall is placed in the middle of the solid boundary nodes and bulk nodes, where the name of halfway comes from. No node in the solid wall involves in the evolution of distribution function. Hence it will take $2\delta t$ for the distribution on the bulk nodes next to the boundary to bounce back by the fullway bounce back rule while δt by the halfway bounce back rule.

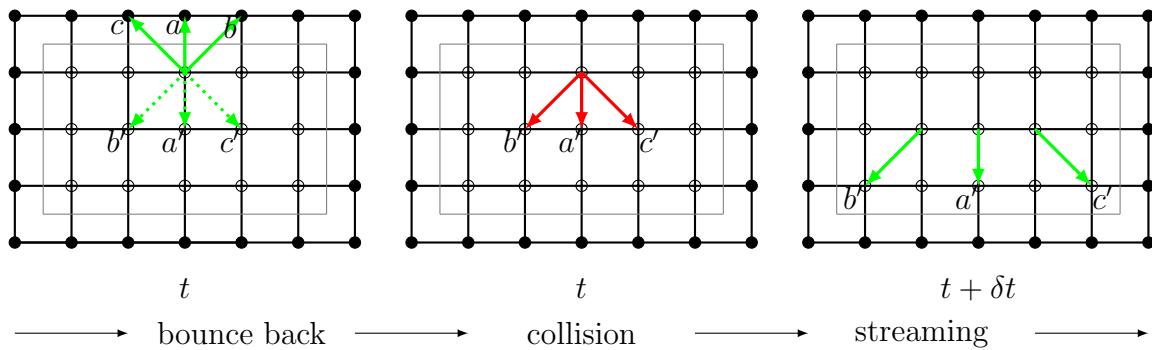


Figure 3.2.3: sketch for halfway bounce back for non-slip straight boundary

Figure 3.2.3 shows how halfway bounce back rule is applied for the non-slip boundary. For corners in 2D, all the unknown distribution functions can be obtained by bounce back or reflect rule.

Remark 3.2.3 While for lattice Boltzmann method in 3D, nothing changed in principle but more unknown distribution functions on boundary faces, boundary edges as well as at corners are to be obtained by bounce back or reflect rule.

3.2.2 Zou-He boundary dynamics

Recall that the full distribution function can be separated into the equilibrium part and non-equilibrium part as

$$f_i(\mathbf{x}, t) = f_i^{(eq)}(\rho, \mathbf{u}) + f_i^{(neq)}(\rho, \mathbf{u}, \nabla \mathbf{u}) \quad i = 0, \dots, 8$$

where $f_i^{(eq)}(\rho, \mathbf{u})$ and $f_i^{(neq)}(\rho, \mathbf{u}, \nabla \mathbf{u})$ are given as (2.4.52) and (2.5.33).

The basic idea of Zou-He boundary dynamics for boundary conditions is based on bounce back for the non-equilibrium part of the distribution function in the normal direction while obtaining the other two known distribution from the mass and momentum conservation as in (3.2.1). Let's consider again the boundary node P in Figure 3.2.1. Firstly, we apply the bounce back rule for the unknown distribution function f_4 ,

$$\begin{aligned} f_4(\mathbf{x}, t) &= f_4^{(eq)}(\rho, \mathbf{u}) + f_4^{(neq)}(\rho, \mathbf{u}, \nabla \mathbf{u}) \\ &= f_4^{(eq)}(\rho, \mathbf{u}) + f_2^{(neq)}(\rho, \mathbf{u}, \nabla \mathbf{u}) \\ &= f_4^{(eq)}(\rho, \mathbf{u}) + f_2(\mathbf{x}, t) - f_2^{(eq)}(\rho, \mathbf{u}) \\ &= f_2(\mathbf{x}, t) - \frac{2}{3}\rho u_y(\mathbf{x}, t) \end{aligned}$$

where f_2 is known and $f_2^{(eq)}$ is also known since the density and velocity are given by the boundary conditions and the conservation equations. There are still f_7 and f_8 to be determined. From (3.2.1) together with (3.2.2) we have

$$\left\{ \begin{array}{l} f_4 = f_2 - \frac{2}{3}\rho u_y, \\ f_7 = f_5 + \frac{1}{2}(f_1 - f_3) - \frac{1}{2}\rho u_x - \frac{1}{2}\rho u_y, \\ f_8 = f_6 - \frac{1}{2}(f_1 - f_3) + \frac{1}{2}\rho u_x - \frac{1}{2}\rho u_y. \end{array} \right. \quad (3.2.5)$$

Similar results can be obtained for other boundaries. As mentioned before, Zou-He boundary dynamics is derived from mass and momentum conservation, so that the mass and momentum are conserved on the boundary. Moreover, two facts, if not taken carefully, will degrade the accuracy of Zou-He boundary dynamics rule. The first is the corner nodes, and the second is the compressibility effect. For the former one, take for example in Figure 3.2.1 the node O lying at the corner of the intersection of inlet and up wall, we have by the bounce back rule for non-equilibrium part of the distribution function that

$$f_1 = f_1^{(eq)} + f_3 - f_3^{(eq)}; \quad f_4 = f_4^{(eq)} + f_2 - f_2^{(eq)}; \quad f_8 = f_8^{(eq)} + f_6 - f_6^{(eq)},$$

while

$$f_5 = f_7 = \frac{1}{2}(\rho - (f_0 + f_1 + f_2 + f_3 + f_4 + f_6 + f_8)).$$

This is slightly different from the original paper [50] by setting f_8 as $f_8^{(eq)} + f_6 - f_6^{(eq)}$ instead of f_6 directly. According to our experiment, it is also a good alternative way to impose boundary conditions at the corners. For the compressibility effect, the recipe goes along with adjusting the equilibrium distribution by modification of density from the item $\rho \mathbf{u}$ to $\rho_0 \mathbf{u}$ where the ρ_0 is a fixed value and only allow the term with single

density ρ to vary, see section 2.6.4. Formally, we change all the variable ρu_x and ρu_y to a single value $v_x = \rho_0 u_x$ and $v_y = \rho_0 u_y$, such that

$$\rho = -v_x + (f_0 + f_1 + f_3 + 2(f_2 + f_5 + f_6)),$$

$$u_x = -\rho + (f_0 + f_2 + f_4 + 2(f_1 + f_5 + f_8)),$$

so that the density (3.2.2) or velocity (3.2.3) are modified correspondingly as well as in (3.2.5).

Lastly, for 2D Zou-He boundary dynamics, the bounce back rule could also be chosen as fullway bounce back or halfway bounce back. As halfway bounce back produce more accurate result, it is preferred in practice.

When applied in 3D, the Zou-He boundary condition rule is not the same as that in 2D, because the number of unknown distribution functions are more than the number of equations (1 equation for mass conservation and 3 equations for momentum conservation) to close the system. In detail, there are 5, 5 and 9 unknown distribution functions for the lattice structure $E_{3,5}^{15}$, $E_{3,5}^{19}$ and $E_{3,5}^{27}$ respectively. One way to deal with this insufficiency of information is to employ bounce back rule for non-equilibrium part of more distribution functions up to close the system with the number of unknown distribution functions equal to the number of equations [59].

However, there is no general way to choose which unknown distribution function should be applied the bounced back rule. A more accurate method is as follows. Take the up wall in the left one of Figure A.0.5 for example, first bounce back the non-equilibrium part of all the unknown distribution functions in direction $e_4, e_9, e_{10}, e_{14}, e_{18}$, then keep the non-equilibrium distribution functions unchanged in the normal direction α (e_4 here), while redistribute the excess part δ_β of the non-equilibrium distribution functions in other directions β (here $e_9, e_{10}, e_{14}, e_{18}$) over all the unknown distribution functions before bounce back. Explicitly, we have

$$\delta_\beta = \sum_i f_i^{(neq)} c_{i\beta} \text{ for } \beta \neq \alpha (\alpha \text{ denotes the normal direction}). \quad (3.2.6)$$

$$f_i^{(neq)} = f_i^{(neq)} - \sum_{\beta \neq \alpha} \frac{1}{n_\beta} c_{i\beta} \delta_\beta \text{ for all unknown } f_i, \quad (3.2.7)$$

where n_β is the number of unknown distribution function for which $c_{i\beta}$ is nonzero.

3.2.3 Finite difference boundary dynamics

The finite difference method is developed by Skordos [61] based on the idea of the separation of the distribution function into equilibrium part and non-equilibrium part.

Firstly, recall that the equilibrium distribution is taken as the zeroth level of Chapman-Enskog expansion of the distribution function, which is up to second order accuracy of Mach number given by

$$f_i^{(eq)} = f_i^{(0)}(\mathbf{x}, t) = w_i \rho \left(1 + \frac{\boldsymbol{\xi}_i \cdot \mathbf{u}}{c_s^2} + \frac{1}{2c_s^4} Q_i : \mathbf{u}\mathbf{u} \right) + O(Ma^3), \quad (3.2.8)$$

where $Q_i = \boldsymbol{\xi}_i \boldsymbol{\xi}_i - c_s^2 I$, while the non-equilibrium part of the distribution function $f^{(neq)}$ is approximated by the first level of Chapman-Enskog expansion of the distribution function $f^{(1)}$. Without considering the force term and the correction term in (2.5.35), we have

$$f_i^{(neq)} \approx \varepsilon f_i^{(1)} = -\frac{w_i \tau}{c_s^2} \left(\underbrace{Q_i : \rho \nabla \mathbf{u}}_{O(\varepsilon Ma)} - \underbrace{\boldsymbol{\xi}_i \nabla : \rho \mathbf{u}\mathbf{u}}_{O(\varepsilon Ma^2)} + \frac{1}{2c_s^2} \underbrace{(\boldsymbol{\xi}_i \cdot \nabla)(Q_i : \rho \mathbf{u}\mathbf{u})}_{O(\varepsilon Ma^2)} \right). \quad (3.2.9)$$

Moreover, since the non-equilibrium distribution function lies on the scale of first order of Knudsen number, which is expected to be very small, it allows us to take for the non-equilibrium distribution function in (3.2.9) the terms only up to the first order of Mach number, which is

$$f_i^{(neq)} \approx -\frac{w_i \tau}{c_s^2} Q_i : \rho \nabla \mathbf{u} = -\frac{\rho w_i \tau}{c_s^2} Q_i : S, \quad (3.2.10)$$

where $S = (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)/2$ since Q is symmetric. (3.2.10) shows the linear dependence on $\nabla \mathbf{u}$ which can be approximated by the finite difference method. Take again the boundary node P for example in Figure 3.2.1. Suppose the index for the location of P is (i, j) , by first order difference we have

$$\begin{aligned} \frac{\partial u_x}{\partial x}(i, j) &= \frac{u_x(i+1, j) - u_x(i, j)}{\delta x} \text{ upwind ,} \\ \frac{\partial u_x}{\partial y}(i, j) &= \frac{u_x(i, j) - u_x(i, j-1)}{\delta y} \text{ downwind,} \end{aligned} \quad (3.2.11)$$

where $\delta x = \delta y$ is the lattice length for one spatial step. This is applicable everywhere on the boundary except at the corners, where the upwind or downwind is chosen correspondingly. For velocity in the vertical direction, the finite difference scheme (3.2.11) can also be applied.

Remark 3.2.4 *The second order approximation of finite difference method can also be applied*

$$\begin{aligned} \frac{\partial u_x}{\partial x}(i, j) &= \frac{u_x(i+1, j) - u_x(i-1, j)}{2\delta x}, \\ \frac{\partial u_x}{\partial y}(i, j) &= \frac{-3u_x(i, j) + 4u_x(i, j-1) - u_x(i, j-2)}{2\delta y}. \end{aligned} \quad (3.2.12)$$

Remark 3.2.5 Notice that the finite difference boundary dynamics does not only use the local information at (i, j) but also from its neighbors, which in a certain degree degrade the advantage of locality or parallelism of lattice Boltzmann method. We can also approximate the non-equilibrium distribution function by the so called nonlinear finite difference method, simply by applying (3.2.9) instead of it's simplified linear form (3.2.10) [52].

Remark 3.2.6 Since lattice Boltzmann method is essentially finite difference approximation for the Boltzmann equation (see [70]), thus the finite difference boundary dynamics performs in the most consistent way with the lattice Boltzmann method. And that's why finite difference boundary dynamics leads to the most robust lattice Boltzmann method, in the sense that Reynolds number can be pushed to very high for the lattice Boltzmann method to be still stable, as we will see in the benchmark of Poiseuille flow in section 3.3.

3.2.4 Regularized boundary dynamics

In order to take the advantage of the parallel merit of the lattice Boltzmann method and achieve a more stable (or robust) way to impose boundary conditions, the regularized boundary dynamics was developed by Lätt and Chopard [60]. The key idea of this boundary dynamics depends on the non-equilibrium tensor

$$\Pi^{(neq)} = \sum_{i=0}^8 Q_i f_i^{(neq)} = \sum_{i=0}^8 Q_i (f_i - f_i^{(eq)}). \quad (3.2.13)$$

From (3.2.10) we have

$$\Pi^{(neq)} = -\frac{\rho\tau}{c_s^2} \sum_{i=0}^8 w_i Q_i Q_i : S = -2\rho\tau c_s^2 S. \quad (3.2.14)$$

Therefore, by (3.2.10) and (3.2.14), the non-equilibrium distribution function can be obtained as

$$f_i^{(neq)} = \frac{w_i}{2c_s^4} Q_i : \Pi^{(neq)}, \quad (3.2.15)$$

where $\Pi^{(neq)}$ is given by (3.2.13). The advantage for recovering $f_i^{(neq)}$ by $\Pi^{(neq)}$ is that once f_i and $f_i^{(eq)}$ are known, it's trivial to compute the non-equilibrium distribution function locally without bothering their neighbors. However, it's not finished because the unknowns f_i in (3.2.13) should be determined firstly in order to get $\Pi^{(neq)}$. According to the original paper [60], the non-equilibrium part of all the unknown distribution functions are bounced back to have a first explicit distribution, then replace all the distribution functions by the regularized procedure.

Remark 3.2.7 Beside bounce back the non-equilibrium part of all the unknown distribution functions, there are other options. They could be first fullway bounce back for the distribution or even halfway bounce back without loss of the advantage of conserving mass and momentum. Moreover, Zou-He boundary condition could also be applied before the regularized procedure.

Remark 3.2.8 Notice that both the finite difference boundary dynamics and the regularized boundary dynamics apply the similar approach to construct the equilibrium distribution function from the exact density and velocity, by the collision invariance we have that both the mass and momentum conservation are achieved.

3.3 Benchmarks with straight boundary

In this section, Poiseuille flow is taken to investigate numerically the stability, accuracy as well as efficiency of the four boundary dynamics. In the second benchmark - lid driven cavity flow, the numerical results between lattice Boltzmann method and finite element method are compared as a verification of lattice Boltzmann method with regularized boundary dynamics and finite difference dynamics. For the last benchmark - Womersley flow, halfway non-equilibrium bounce back dynamics and Zou-He boundary dynamics are applied, mainly to investigate the compressibility effect and compare the compressible and incompressible lattice Boltzmann method. All benchmarks are implemented by Matlab.

3.3.1 Poiseuille flow in 2D

Poiseuille flow in 2D is the steady state incompressible flow between two stationary parallel plates where the flow is driven by an imposed pressure gradient, as shown in Figure 3.3.1.

In dimensionless system, the length in x-direction of the computational domain Ω is $L_x = 4$ while in y-direction it is $L_y = 1$. The problem for Poiseuille flow is given as (2.6.10) with the following boundary conditions:

$$\left\{ \begin{array}{l} p^{out} = 0 \text{ on } \Gamma_{out}, \\ p^{in} = p^{out} + 8L_x/Re \text{ on } \Gamma_{in}, \\ \mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_{wall}, \\ u_y = 0 \text{ on } \Gamma_{in} \cup \Gamma_{out}. \end{array} \right. \quad (3.3.1)$$

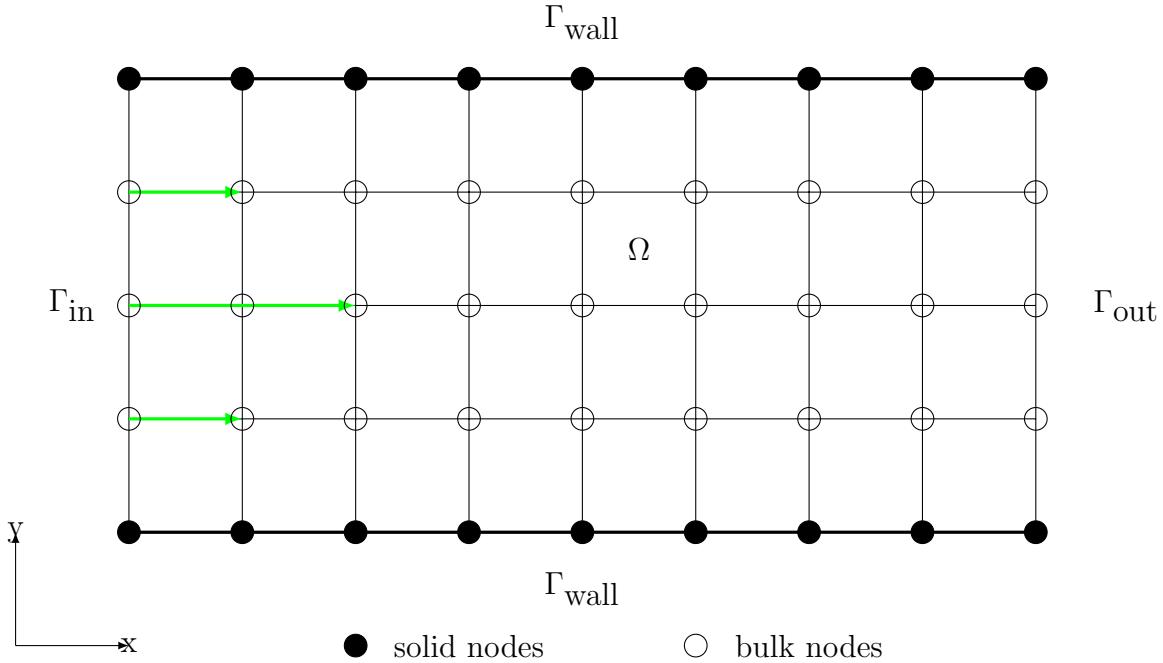


Figure 3.3.1: poiseuille flow driven by constant pressure gradient in 2D, with constant pressure $p = 0$ on the outlet boundary.

And the analytical solution is given as (p^a denotes analytical solution of pressure, the same for velocity)

$$\begin{cases} u_x^a(x, y) = 4(y - y^2) & \text{in } \Omega, \\ u_y^a = 0 & \text{in } \Omega, \\ p^a(x, y) = -8(x - 2)/Re & \text{in } \Omega. \end{cases} \quad (3.3.2)$$

In order to analyze the accuracy of the lattice Boltzmann method depending on spatial and time step as well as to compared the four different dynamics for imposing boundary conditions, we choose the following parameters:

- Vary the number of cells N for spatial discretization of vertical boundary (inlet and outlet) as $N = [8, 16, 32, 64]$,
- Vary δt to be the zeroth, first and second order of δx , e.g. $\delta t = 10^{-3}$, $\delta t = \delta x/100$ and $\delta t = \delta x^2$,
- Keep viscosity fixed as $\nu = 0.1$ or Reynolds number fixed to be $Re = 10$,
- Define the error in L^2 norm as

$$\text{Error} = \frac{\sqrt{\sum_{i=0}^{Nx} \sum_{j=0}^{Ny} ((u_x(i, j) - u_x^a(i, j))^2 + (u_y(i, j) - u_y^a(i, j))^2}}{Nx \times Ny}. \quad (3.3.3)$$

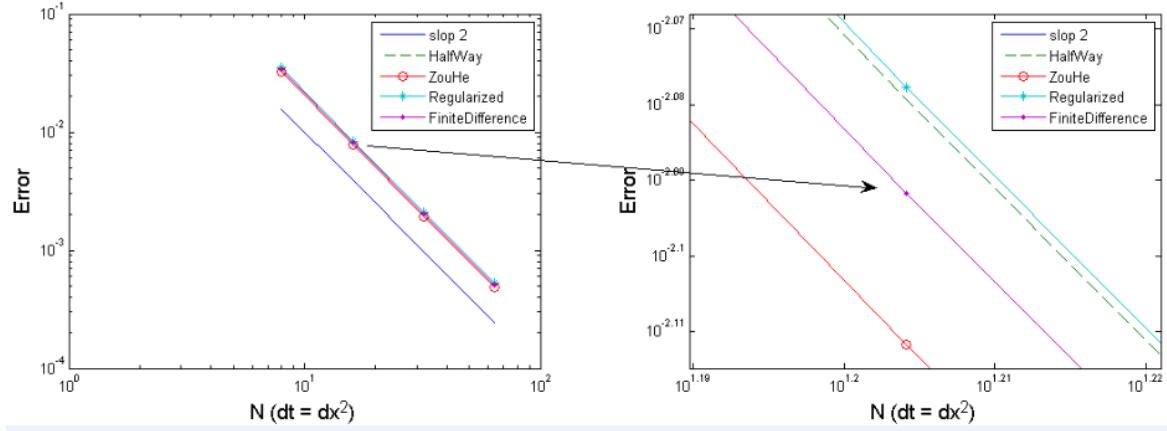


Figure 3.3.2: Error depending on N for Poiseuille flow with $\delta t = \delta x^2$

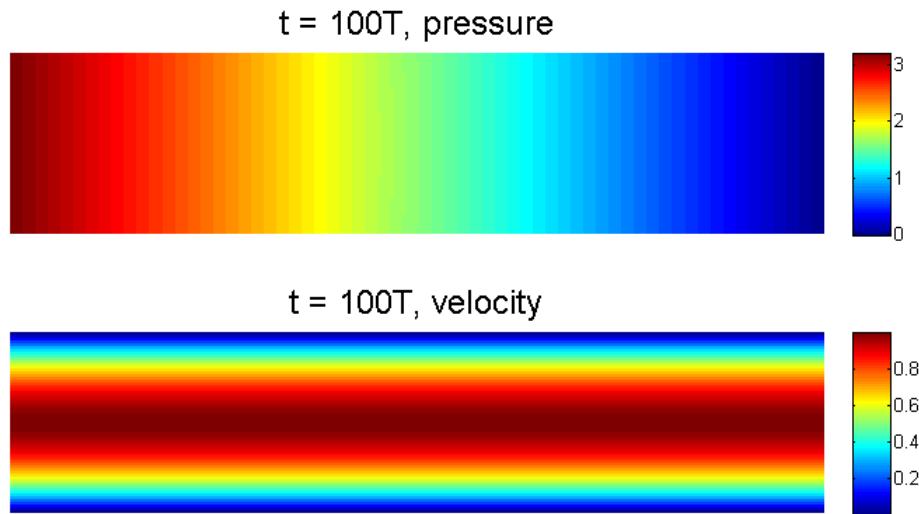


Figure 3.3.3: Poiseuille flow with initial velocity as 0 and constant pressure; pressure and velocity tend to be steady when the time goes long enough, e.g 100T

In Figure 3.3.3 the numerical simulation is shown to reach the steady state of Poiseuille when the time is at 100th period (by period T we mean $T = Lx/(\delta x/\delta t)$). Figure 3.3.2 shows the convergence rate of the error between the numerical simulation results obtained by lattice Boltzmann method and the analytical solution with four different boundary dynamics to impose boundary conditions as we have examined in the last section. Obviously, all the four methods produce almost undistinguished error and make the error converge with second order of accuracy, as it tells from the Figure 3.3.2. To compare more precisely the four methods, we enlarge a small region of the left one of Figure 3.3.2 to be visible of the difference among the four methods in the

right one, and it shows that the Zou-He boundary dynamics leads to the smallest error, followed by finite difference boundary dynamics and the boundary dynamics of halfway bounce back for non-equilibrium part of distribution function, bottomed by regularized boundary dynamics.

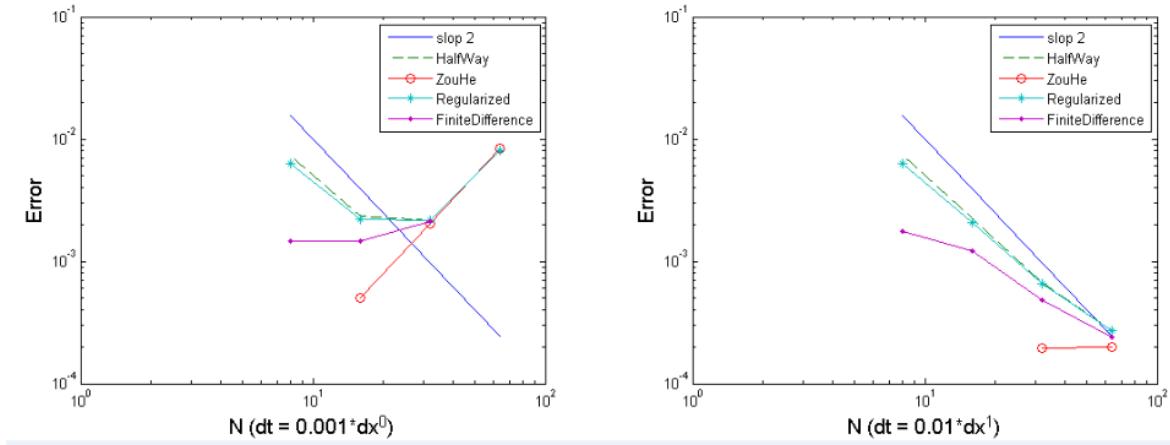


Figure 3.3.4: Poiseuille flow with $\delta t = 1/1000$ Left and Right $\delta t = \delta x/100$

Figure 3.3.4 displays the convergence rate when choosing δt to the zeroth order and first order dependent upon δx . From this figure we can see by the left one that the most precise method - Zou-He boundary dynamics reaches the lower bound of the error at beginning (the first point at $N = 8$ is missing because of stability issue of this method), while finite difference boundary dynamics keeps the error flat before increasing it, and the other two boundary dynamics decrease the error with second order of accuracy at first and start to increase the error later on. The right one of Figure 3.3.4, where $\delta t = O(\delta x)$, tends to keep the error flat when the grid is fine enough.

Remark 3.3.1 Analysis of Figure 3.3.2 and Figure 3.3.4 stands in favor of the heuristic error dependence on spatial and time step as well as the compressibility effect (arising due to large Mach number) as

$$E_{total} = E_{\delta x} + E_{\delta t} + E_{Ma} = O(\delta x^2) + O(\delta t^2) + O((\delta t/\delta x)^2), \quad (3.3.4)$$

where $Ma = \delta t/\delta x$. Clearly, when $\delta t = O(\delta x^2)$ the dominating factor for the error is the spatial resolution up to the power of second order as $O(\delta x^2)$, while $\delta t = O(\delta x)$, the error is dominated by spatial resolution as δx^2 when δx is big and switches to the compressibility effect as the main contributing factor. Finally, when the time step is fixed to a constant despite how δx changes, the dominating factor will be merely the compressibility effect and increasing as δx is decreasing. (see [42] for more details).

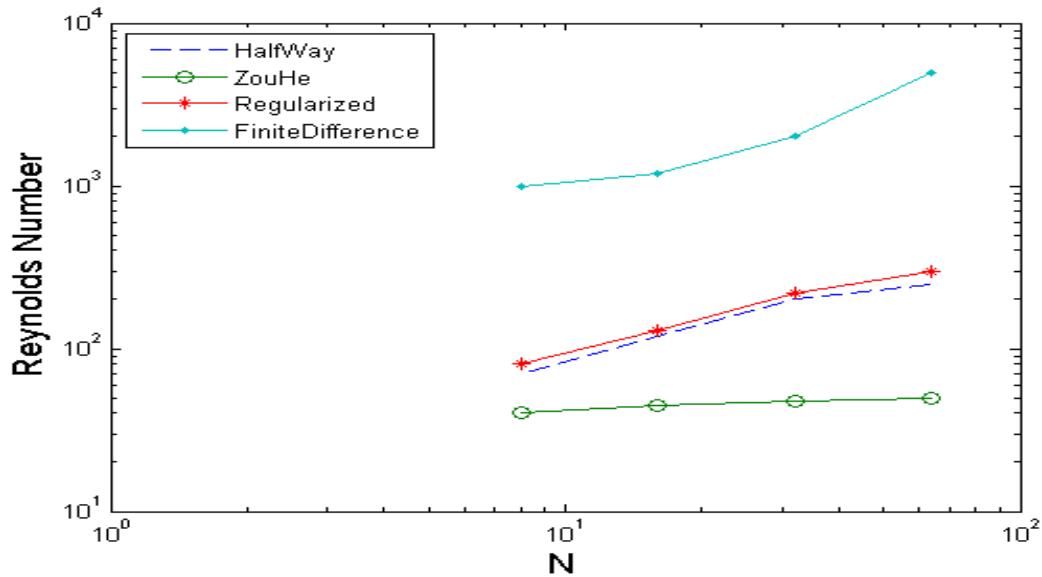


Figure 3.3.5: testing the stability (measured by how high the Reynolds number can climb) of the four different methods for imposing boundary conditions, Poiseuille flow with $\delta t = \delta x^2$

As can be seen from Figure 3.3.5, maybe there is no free lunch when stability and accuracy (though slightly different) are taken into account as trade-off. Zou-He boundary dynamics is identified to produce relatively the smallest error with sacrifice of stability; the dynamics with the biggest error - finite difference boundary dynamics allows Reynolds number to increase much higher than all the other methods, while halfway bounce back rule for non-equilibrium distribution function and the regularized dynamics hold hands close to grow the Reynolds number as the mesh is refined. It's worth to mention also that finite different dynamics is more time consuming and degrades the outstanding merit of lattice Boltzmann method - locality or parallelism.

Remark 3.3.2 *The result about stability is different from that reported in [59] when it comes to regularized boundary dynamics. In that paper, it shows that the regularized boundary dynamics pushes the Reynolds number as high as that by finite difference boundary dynamics. However, the same testing example fails to arrive at the same conclusion. One possible explanation could be that regularized boundary dynamics borrows the idea of halfway bounce back rule for non-equilibrium distribution function to give the unknown distribution functions before reconstructing the distribution function. It is here that the error mainly comes from. And it makes sense that the stability of non-equilibrium halfway bounce back dynamics and regularized boundary dynamics shares the same property. For another point, though finite difference boundary dynamics also*

belongs to the family of reconstruction, it gives the unknown distribution functions from the difference for velocity on the boundary, which is consistent with lattice Boltzmann method as a finite difference approximation for the Boltzmann - BGK equation.

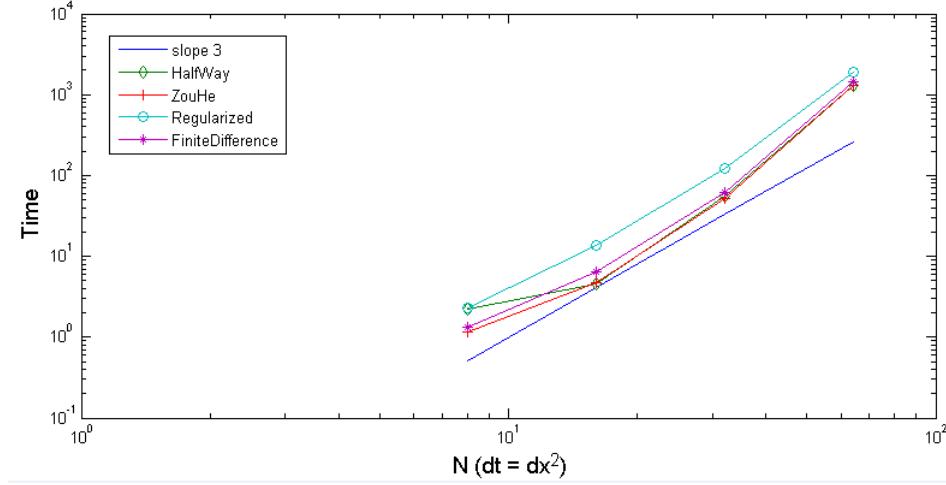


Figure 3.3.6: CPU time comparison between different boundary conditions

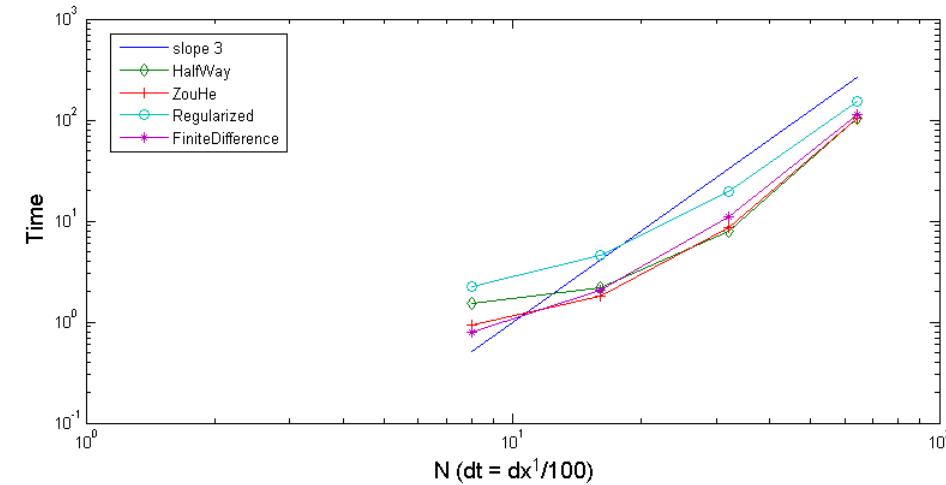


Figure 3.3.7: CPU time comparison between different boundary conditions

From (3.3.8), (3.3.7) and (3.3.6) we can see that in general, regularized boundary dynamics is the most time consuming method, followed by the finite difference boundary dynamics, and then comes Zou-He boundary dynamics, while Halfway bounce back rule for non-equilibrium distribution is slightly faster than Zou-He boundary dynamics. According to this basic benchmark implemented by Matlab, the computational time, CPU_t , for $\delta t = O(\delta x^2)$ tends to be proportional to N^4 , while it tends to N^2 for $\delta t = O(1)$, and N^3 for $\delta t = O(\delta x)$. Summarizing, we have

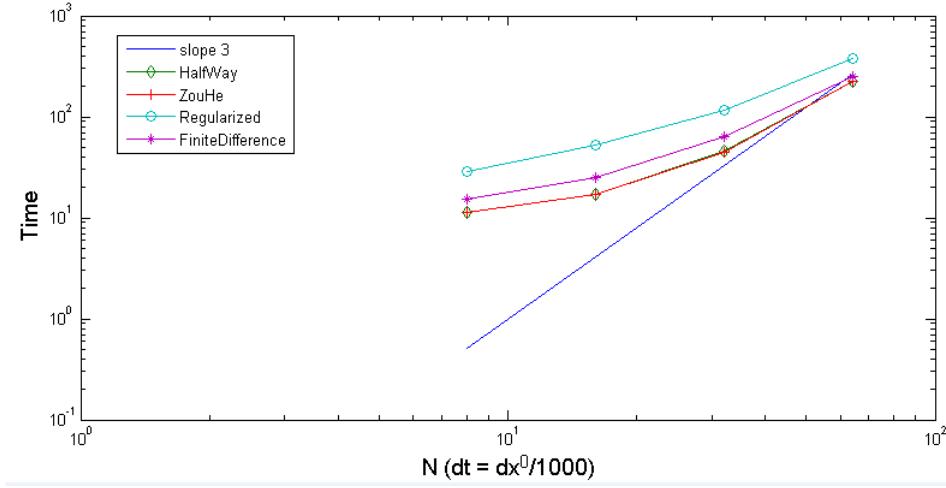


Figure 3.3.8: CPU time comparison between different boundary conditions

- $\delta t = O(1) \rightarrow CPU_t = O(N^2)$,
- $\delta t = O(\delta x) \rightarrow CPU_t = O(N^3)$,
- $\delta t = O(\delta x^2) \rightarrow CPU_t = O(N^4)$.

In order to illustrate the compressibility effect of lattice Boltzmann methods, we start from the initial condition where the velocity vanishes everywhere and the pressure is constant. Take $T = L_x/(\delta x/\delta t)$ as one period in our simulation, take $N = 64$ and $\delta x = 1/N$; $\delta t = \delta x^2$, for a laminar flow with Reynolds number $Re = 10$, we have the simulation in the first ten periods as shown in figure (3.3.9).

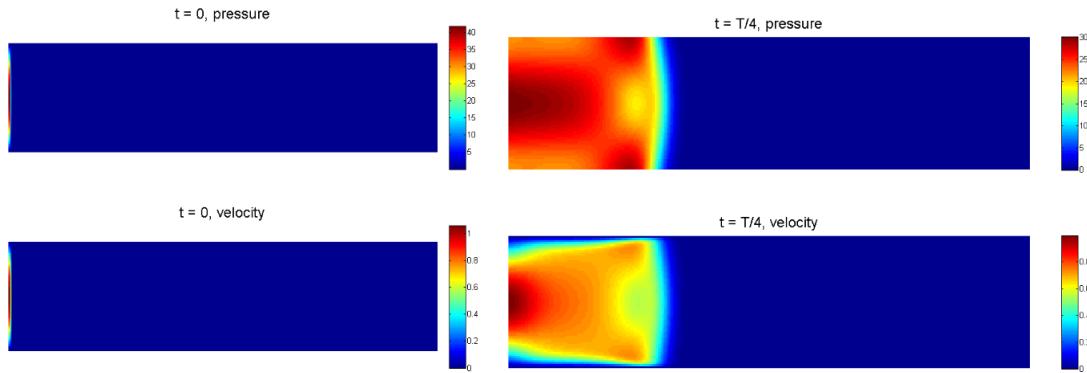


Figure 3.3.9: Poiseuille flow with initial velocity as 0 and constant pressure; Left is the pressure and velocity at the initial time while Right shows at a quarter period $t = T/4$

As can be seen from time at 0 and time at $T/4$, while the pressure and velocity are evolving to the left quarter of the domain in the first quarter of period, while on the

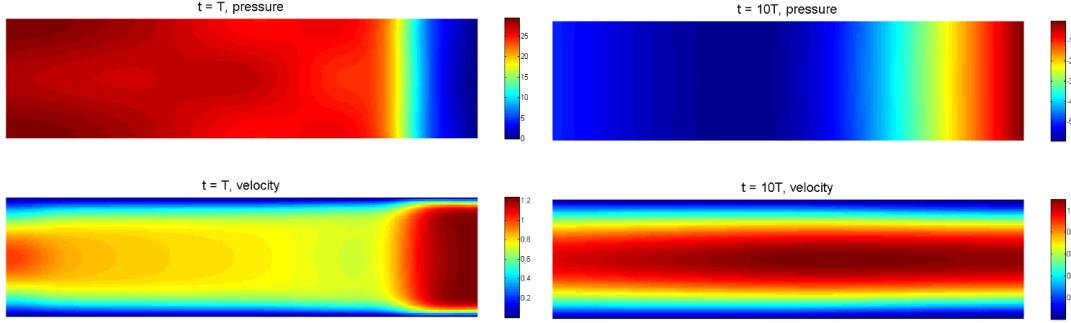


Figure 3.3.10: Poiseuille flow with initial velocity as 0 and constant pressure; Left is the pressure and velocity at one period while Right shows at $t = 10T$

right boundary they remain unchanged until the time arrives at $t = T$. Therefore, we can conclude firstly that the lattice Boltzmann method is a weak compressible method, even for the simulation of incompressible flow. Further more, the compressibility effect last as long as at least 10 periods to $10T$, compared to the steady flow in Figure [3.3.3].

Remark 3.3.3 From Figure [3.3.9], [3.3.10] and [3.3.3] we can see that despite lattice Boltzmann method can achieve the precise result in the long run for steady flow, the unideal initial conditions degrade the lattice Boltzmann method significantly (in the sense of time consuming) to simulate the steady flow, let alone the unsteady flow that depends also on the time. Therefore, how to impose precise initial conditions including both velocity and pressure becomes as important as boundary conditions for lattice Boltzmann method. This problem is dealt within this thesis by multigrid lattice Boltzmann method in section [4.5.2].

3.3.2 Lid driven cavity flow in 2D

Lid driven cavity flow in 2D is one of the most used and popular testing examples. The flow is situated within a square box in 2D, driven by imposed flow with constant velocity on one interface, and zero velocity on the other three interfaces. As shown in Figure [3.3.11], the boundary is divided into three walls and one interface where the flow comes across with constant velocity in x -direction. The problem for lid driven cavity flow is given as (2.6.10) with the following boundary conditions,

$$\begin{cases} \mathbf{u} = \mathbf{0} & \text{on } \Gamma_{\text{wall}}, \\ u_x = 1, \quad u_y = 0 & \text{on } \Gamma_{\text{surface}}. \end{cases} \quad (3.3.5)$$

There is no analytical solution for the above problem. Alternatively, we can compare the numerical results by lattice Boltzmann method with the results by finite element

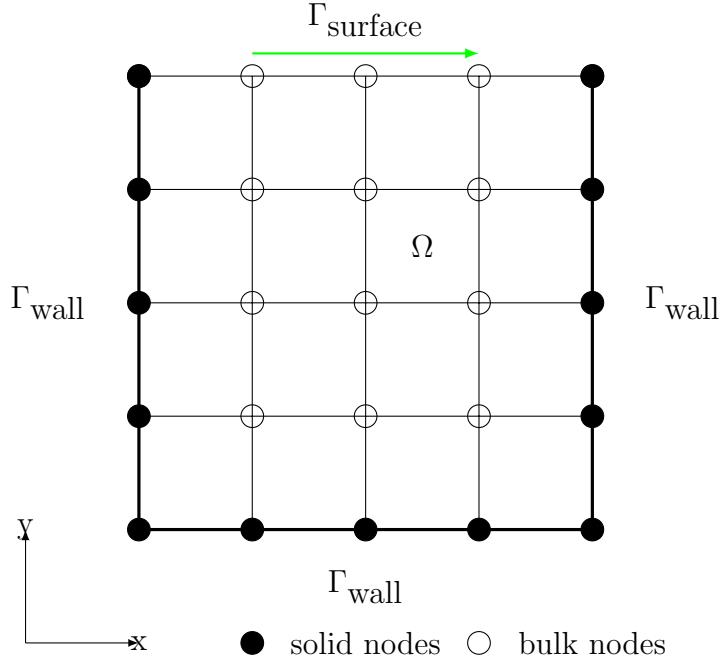


Figure 3.3.11: lid driven cavity flow in the region of a square box in 2D, with prescribed constant velocity in x -direction on the up boundary and zero velocity on the wall

method in [74]. Numerical results for both low and high Reynolds number will be compared.

As a verification of lattice Boltzmann method, we choose the following parameters:

- Viscosity are chosen to be $\nu = 0.01$ and $\nu = 0.001$ respectively,
- The length of the computational domain in x -direction and y -direction are both set to be 1 in dimensionless system, e.g. $L_x = L_y = 1$,
- The domain is discretized with 129×129 lattice nodes,
- The time step is specified as $\delta t = 0.2\delta x$,
- Regularized boundary dynamics is our option for imposing boundary conditions for $\nu = 0.01$ or $Re = 100$,
- finite difference boundary dynamics is our option for imposing boundary conditions for $\nu = 0.001$ or $Re = 1000$.
- The initial velocity is set to be 0 everywhere except on the up boundary $u_x = 1$, while the pressure is set to be constant $p = 0$ everywhere in the domain.

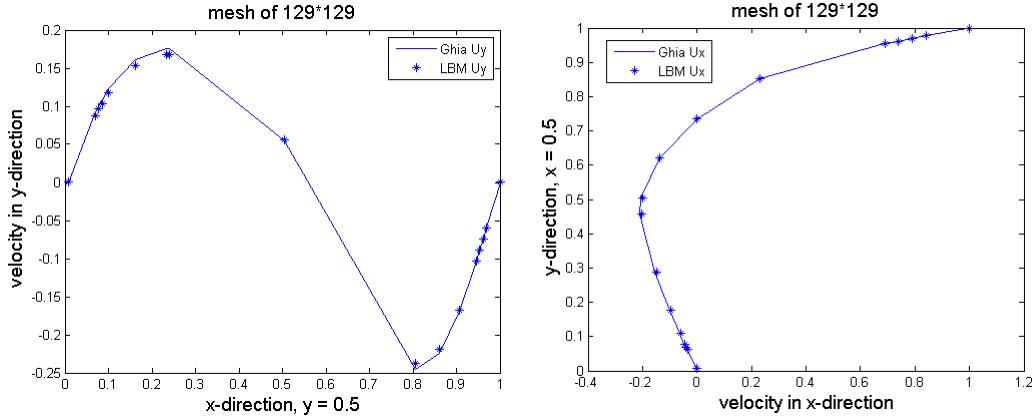


Figure 3.3.12: comparison of velocity by lattice Boltzmann method and finite element method when $Re = 100$. Left, $u_x, y = 0.5$ Right, $u_y, x = 0.5$

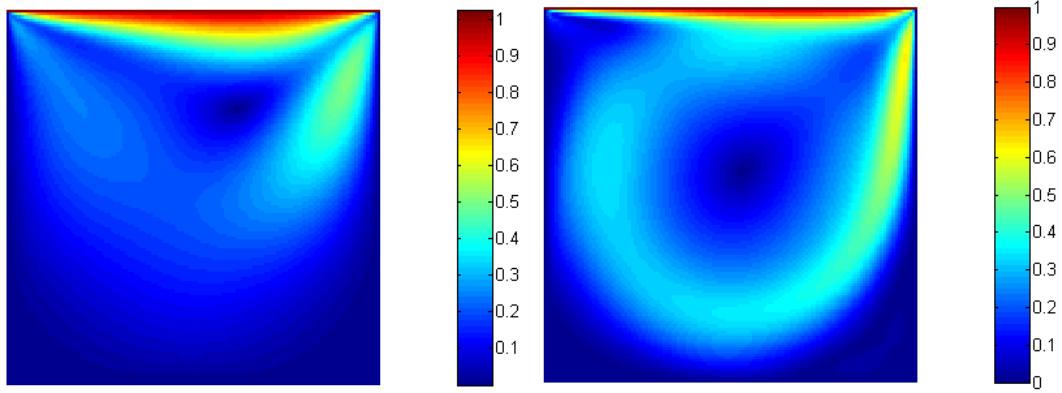


Figure 3.3.13: velocity in L^2 norm, Left: $Re = 100$, Right: $Re = 1000$

At first, we run the simulation with $Re = 100$ until the L^2 error between two successive step is smaller than 10^{-6} . The Figure 3.3.12 for cavity flow are the results of velocity in both directions compared with that obtained with finite element method by Ghia in [74], which displays good agreement.

When the Reynolds number is increased to 1000 for cavity flow, we have the steady flow when the simulation time goes up to 1000 periods as shown in Figure 3.3.13. Figure 3.3.14 shows the velocity along y -axis in the middle of x -axis, and along x -axis in the middle of y -axis, compared to that given by Ghia [74]. The last two figures in Figure 3.3.15 display the streamline of cavity flow in steady state. Both of the comparison for low Reynolds number $Re = 100$ and the high one $Re = 1000$ lead to the conclusion that lattice Boltzmann method is a good alternative solver for Navier-Stokes equations. Meanwhile, regularized boundary dynamics and finite difference boundary dynamics are well qualified for imposing boundary conditions.

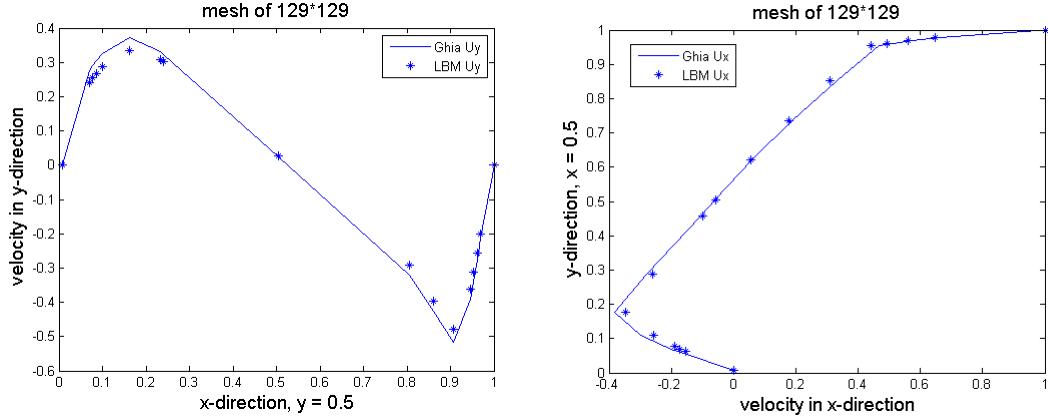


Figure 3.3.14: comparison of velocity by lattice Boltzmann method and finite element method when $Re = 1000$. Left, $u_x, y = 0.5$ Right, $u_y, x = 0.5$

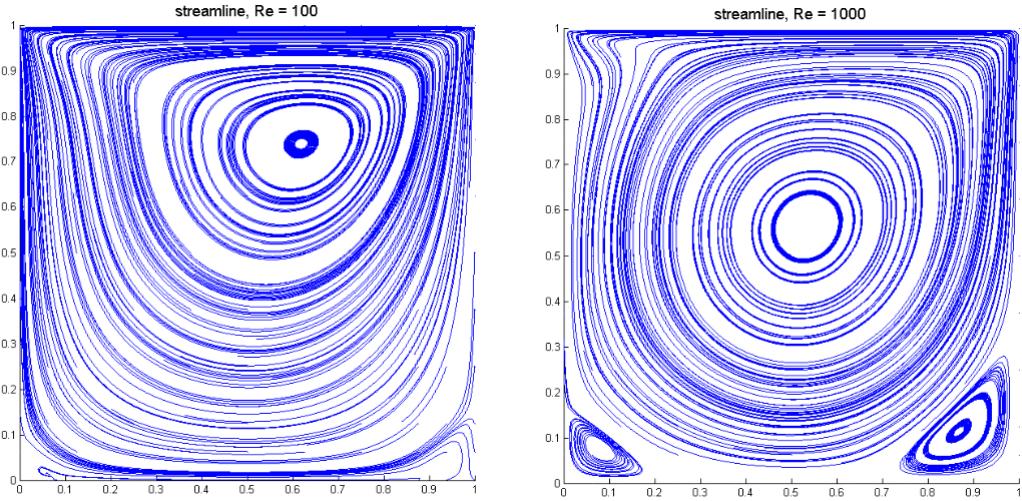


Figure 3.3.15: streamline of velocity, Left: $Re = 100$, Right: $Re = 1000$

3.3.3 Womersley flow in 2D

Womersley flow (see [59]) is a laminar flow similar to Poiseuille flow, as shown in Figure 3.3.1. The difference is that the pressure gradient along x -axis is oscillating in time according to a periodic function, in dimensionless system we have

$$\left\{ \begin{array}{l} p^{out} = 0 \text{ on } \Gamma_{out}, \\ p^{in} = p^{out} + AL_x \cos(\omega t) \text{ on } \Gamma_{in}, \\ \mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_{wall}, \\ u_y = 0 \text{ on } \Gamma_{in} \cup \Gamma_{out}. \end{array} \right. \quad (3.3.6)$$

where A is the amplitude of the oscillation and ω is the frequency. Define the Womersley number as

$$\alpha = \frac{L}{2} \sqrt{\frac{\omega}{\nu}}. \quad (3.3.7)$$

The analytical velocity for Womersley flow is given as [75]

$$u_x(x, y, t) = -\text{real} \left[\frac{Aie^{iat}}{\alpha\rho} \left(1 - \frac{\cos[\sqrt{-i\alpha^2}(2y/l_y - 1)]}{\cos(\sqrt{-i\alpha^2})} \right) \right], \quad u_y(x, y, t) = 0, \quad (3.3.8)$$

where *real* denote the real part of the complex number.

From Poiseuille flow we can see that the compressibility effect is obvious despite the Mach number is small. The compromised solution for this problem is by letting the simulation run for a long time until steady state.

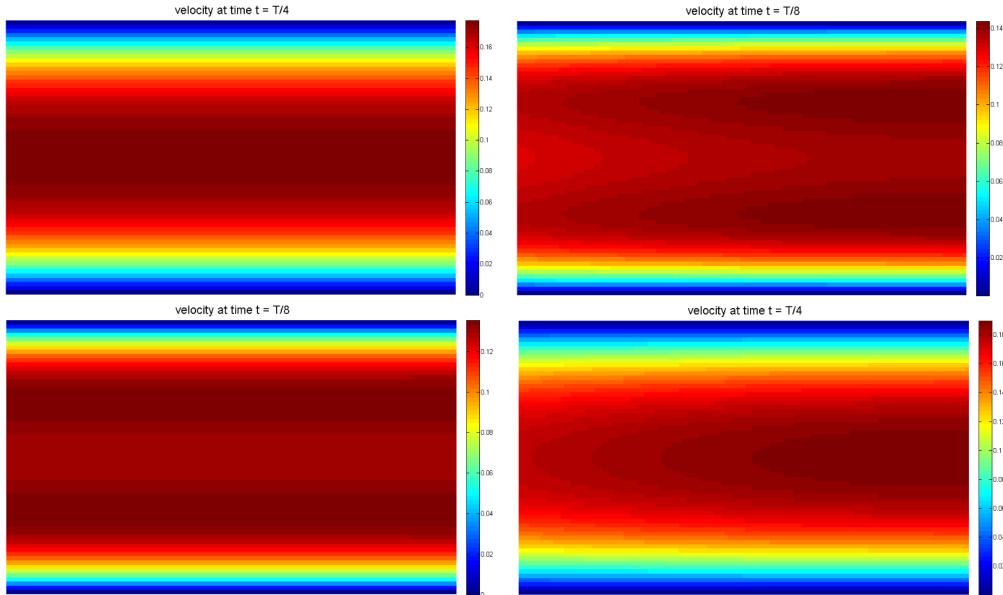


Figure 3.3.16: $u_x, x = 1$ at the first quarter period with Womersley number 3.9633. Left: analytical solution. Right: numerical solution by compressible lattice Boltzmann method. Up: at time $t = T/8$. Down: at time $t = T/4$

However, for time dependent flow, we have to use the incompressible lattice Boltzmann method as introduced in section 2.6.4. Otherwise the compressibility effect will keep the flow inside the domain persisting for a while before the information imposed on the boundary passing all the way to the center, so that the compressible lattice Boltzmann method is not suitable for the time dependent simulation [76].

The parameters are taken as follows:

- Take computational domain as $L_x = 2, L_y = 1$,

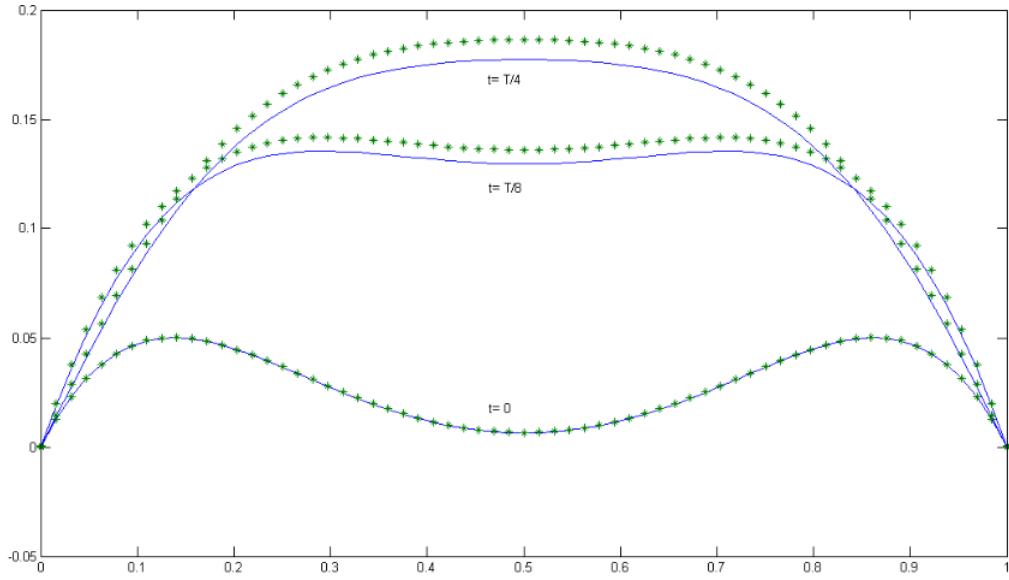


Figure 3.3.17: comparison between analytical velocity and numerical velocity $u_x, x = 1$ by compressible LBM within a quarter period with Womersley number 3.9633

- Viscosity is always given as $\nu = 0.1$ in the following three test cases,
- Choose **halfway non-equilibrium bounce back rule** to impose boundary conditions on up wall and down wall, **Zou-He Boundary dynamics** for inlet and outlet boundary,
- In the first test case by **compressible** lattice Boltzmann method, take $\omega = 2\pi$ so that the period is $T = 2\pi/\omega = 1$, and Womersley number is given by (3.3.7) as $\sqrt{2\pi/0.1}/2 = 3.9633$, spatial step is $\delta x = 1/64$ and time step is $\delta t = \delta x^2$,
- In the second test case by **incompressible** lattice Boltzmann method, take Womersley number the same as in the first test case, spatial step is $\delta x = 1/128$ and time step is $\delta t = \delta x^2$,
- In the third test case by **incompressible** lattice Boltzmann method, increasing the frequency of Womersley flow up to ten times of the first case, $\omega = 20\pi$, we have the Womersley number as $\sqrt{20\pi/0.1}/2 = 12.5331$, spatial step and time step is the same as in the second test case.

Figure 3.3.16 and Figure 3.3.17 display the obvious compressibility effect by the first test case. Figure 3.3.18 and Figure 3.3.19 show the second test case, from which we can see that the incompressible lattice Boltzmann method is pretty good at catching the

time-dependent flow. As for the boundary conditions, halfway non-equilibrium bounce back dynamics and the Zou-He boundary dynamics work well.

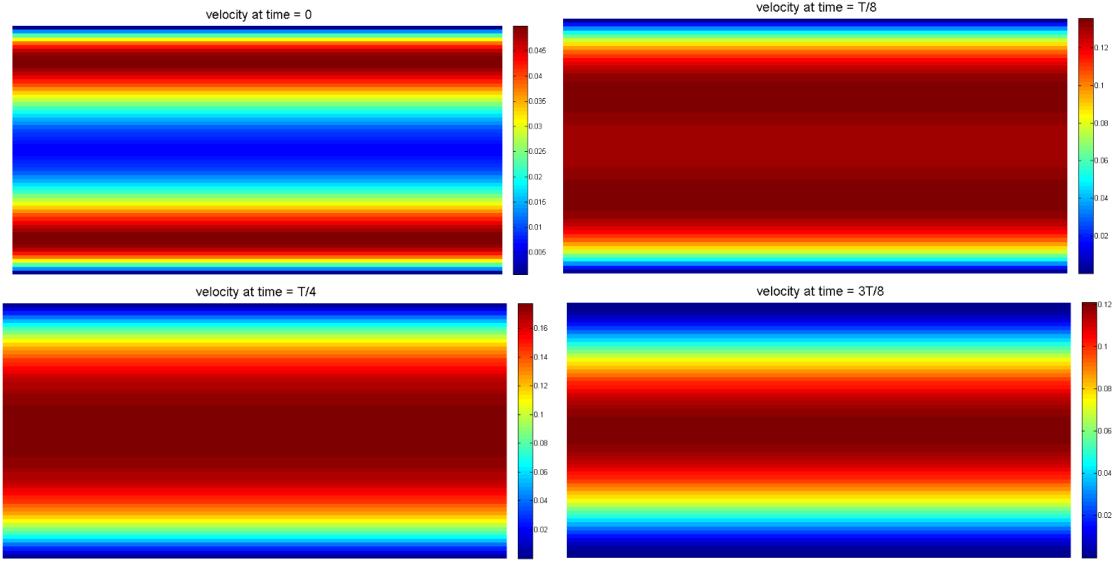


Figure 3.3.18: $u_x, x = 1$ at the first half period with Womersley number 3.9633, by incompressible LBM

Figure [3.3.20] and [3.3.21] display very consistent result between the numerical and analytical solution. However, simulation at time = $3T/8$ in Figure [3.3.20] shows still a little bit compressibility effect, which can be reduced by taking smaller time step or increase the spatial refinement (see section [2.6.4]). However, the compressibility effect can not be eliminated totally even by the so called incompressible lattice Boltzmann method [42].

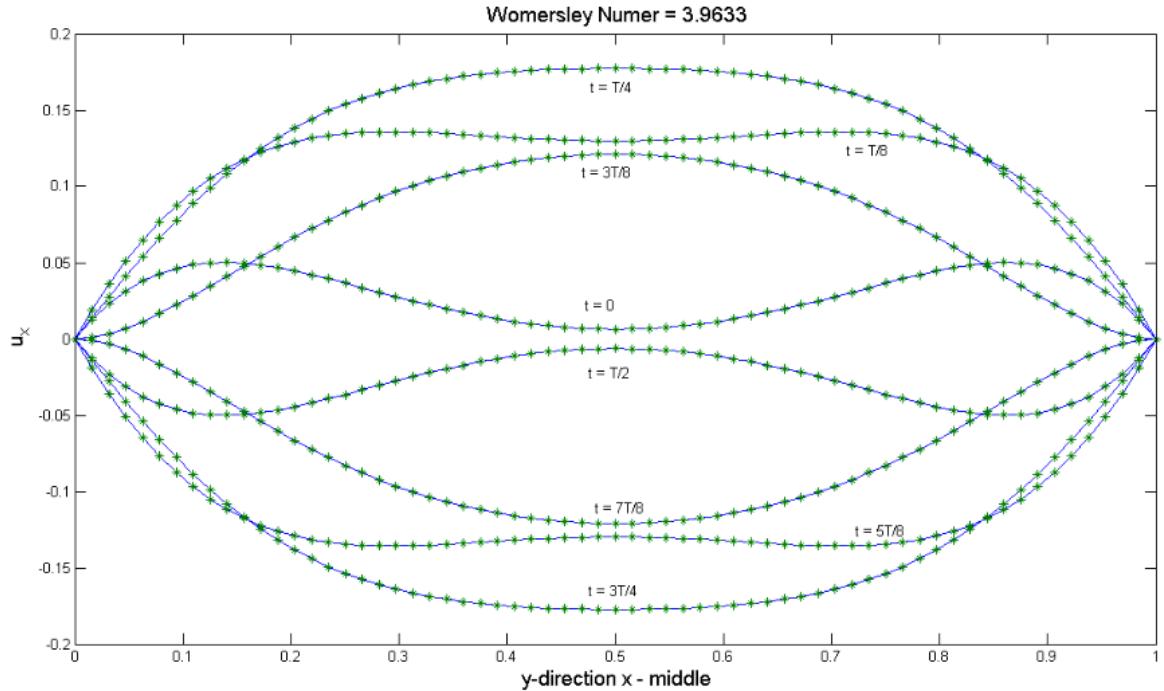


Figure 3.3.19: comparison between analytical velocity and numerical velocity u_x , $x = 1$ within one period with Womersley number 3.9633 by incompressible LBM

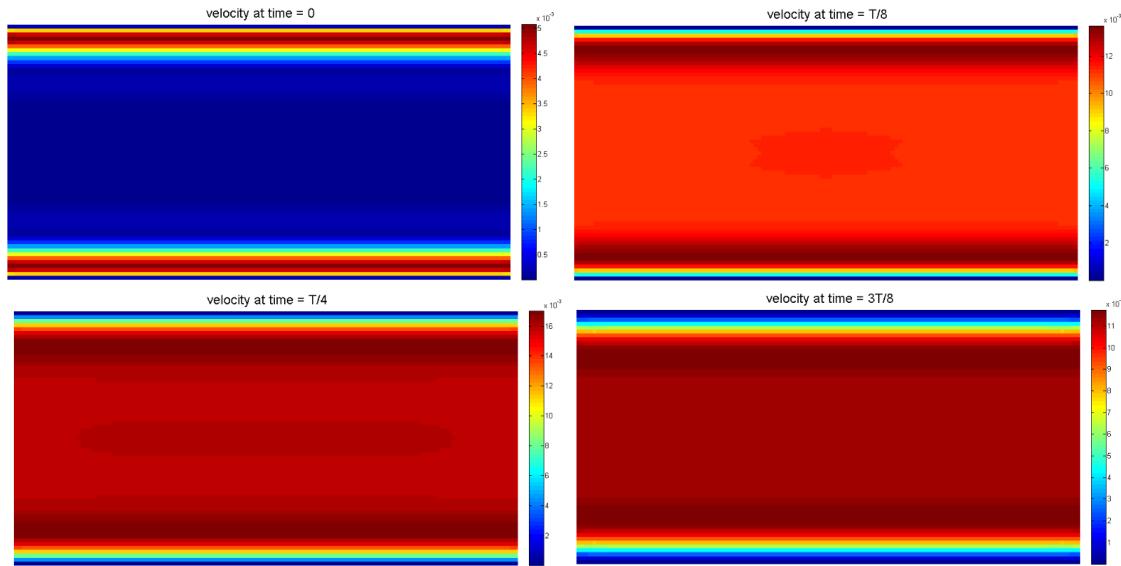


Figure 3.3.20: u_x , $x = 1$ at the first half period with Womersley number 12.5331 by incompressible LBM

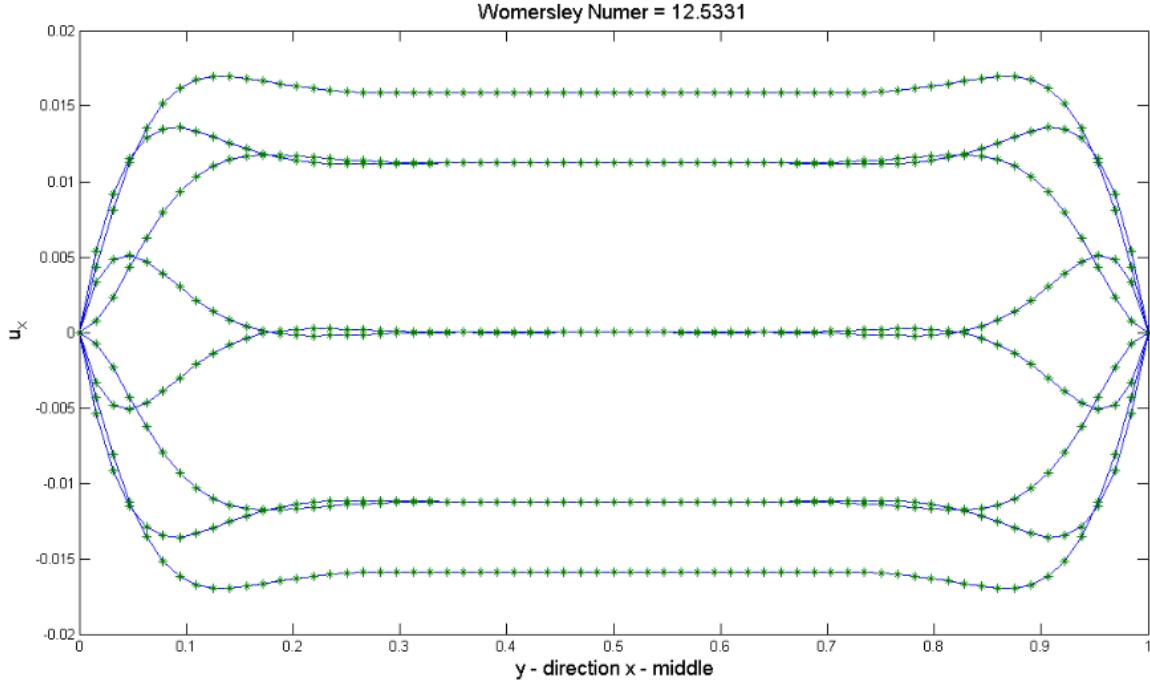


Figure 3.3.21: comparison between analytical velocity and numerical velocity $u_x, x = 1$ within one period with Womersley number 12.5331 by incompressible LBM

3.4 Curved boundary conditions

Though various methods aforementioned for dealing with straight boundary conditions are appreciated for their high order of accuracy and concision for implementation, yet, they are not applicable for more realistic geometries - blood flow in arteries, channel flow with cylinder obstacle inside, flows across porous medium, to name a few. Therefore, we have to take care of the curved boundary in general. The difficulty for curved boundary, compared to that in straight boundary, are mainly twofolds. On the one hand, the boundary is not exactly located on the lattice site or in the middle, but rather arbitrary, just having a look at a cylinder sitting on the lattice. On the other hand, the density or velocity can not be obtained by conservation law as for the straight boundary (see [3.2.1]) since the unknown distribution functions are too many (see right of Figure 3.4.1) that the known distribution functions cannot provide enough information. As shown in the left and right of Figure 3.4.1, the curved boundary wall sit away from the lattice nodes and the unknown distribution functions is more or less than those of the straight boundary (Figure 3.4.1 Middle).

Similar to the straight boundary condition, for curved boundary conditions we identify two different approaches. The first one belongs to the family of bounce back scheme, such as the generalization of the bounce back rule proposed by Lallemand et al.

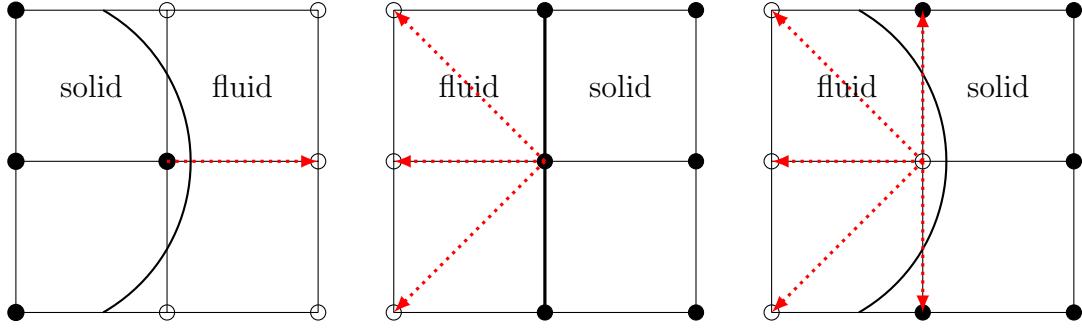


Figure 3.4.1: sketch for curved boundary between solid and fluid (Left and Right), compared to straight boundary (Middle). The red arrow denotes the unknown information streaming from solid to fluid. Left: 1 unknown, Middle: 3 unknowns, Right: 5 unknowns

[55] [63], the so called one point boundary condition, which is essentially a modification for bounce back by Junk et al. [67] and the reflection back scheme with modification of velocity gradient to the non-equilibrium distribution by Chen et al. [64]. The second approach is based on the fitting of the distribution function at the boundary nodes of the curved boundary. The pioneers in this approach are back to Filippova et al. [53], whose method was improved shortly after that by Mei et al. [54] for stability and applicable range. Later on, Yu et al. [57] took a uniform fitting method to simplify the interpolation. Beside these two approaches, another method proposed recently by Dupuis et al. [68] considered the effects of the boundary wall as a force term in the lattice Boltzmann equation, which also applied the idea of interpolating the velocity and density on the boundary nodes from their neighbors. Based on the idea of interpolating and extrapolating the velocity and density, Verschaeve et al. [58] borrowed Lätt et al.'s [42] regularized boundary dynamics for straight boundary condition to the curved one, and obtained the second order accuracy too.

In section 3.4.1, the Filippova-Hänel boundary dynamics (FH) for curved boundary condition as well as the improved version by Mei et al. (FHetMei) are illustrated firstly. Then it comes to the Bouzidi-Firdauss-Lallemand boundary dynamics (BFL) in section 3.4.2. The uniform fitting model (Uniform) developed by the Yu et al. is explained in section 3.4.3, which is followed by the halfway bounce back scheme for its sake of simplicity in section 3.4.4. The method of regularized boundary dynamics in the family of Distribution Reconstruction is illustrated in the last section 3.4.5. Finally, Taylor-Couette flow is taken as a benchmark for the verification and analysis of accuracy and efficiency of these boundary dynamics for curved boundary. And the simulation of blood flow in the left arm with a more realistic shape is carried out.

3.4.1 Filippova-Hänel boundary dynamics

The illustrative geometry for the method of direct fitting by Filippova and Hänel is taken the same as that in Mei's [54] paper as shown in Figure 3.4.2. The left up triangle is occupied by fluid while in the opposite direction locates the solid. The direction from fluid to solid is denoted as $\bar{\alpha}$ and its opposite as α , the interest nodes are indexed as $\mathbf{x}_{ff}, \mathbf{x}_f, \mathbf{x}_w, \mathbf{x}_b$ while the velocity on these nodes are denoted as $\mathbf{u}_{ff}, \mathbf{u}_f, \mathbf{u}_w, \mathbf{u}_b$, notice that \mathbf{u}_w is the given Dirichlet boundary conditions. The distance between the first fluid node \mathbf{x}_f and the wall nodes \mathbf{x}_w in the vertical direction is $\Delta\delta x$, where the ratio $0 \leq \Delta \leq 1$ is defined as

$$\Delta = \frac{|\mathbf{x}_f - \mathbf{x}_w|}{|\mathbf{x}_f - \mathbf{x}_b|}. \quad (3.4.1)$$

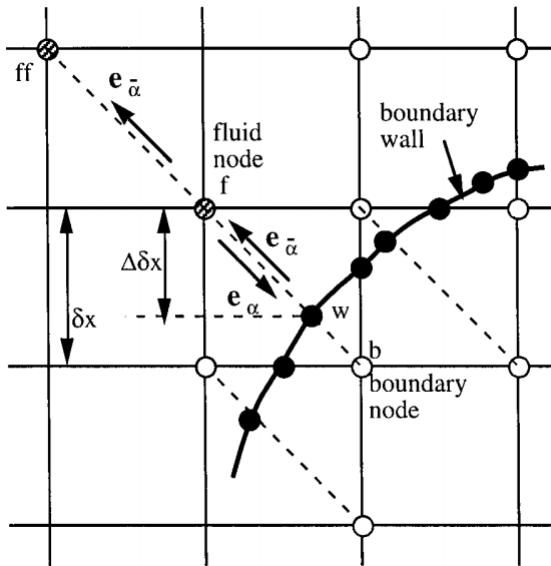


Figure 3.4.2: sketch for curved boundary condition, from [54]

Recall the discrete lattice Boltzmann - BGK model with δt ,

$$\text{collision: } \tilde{f}_\alpha(t, \mathbf{x}) = f_\alpha(t, \mathbf{x}) - \frac{1}{\tau} (f_\alpha(t, \mathbf{x}) - f_\alpha^{(eq)}(t, \mathbf{x})) \quad (3.4.2)$$

$$\text{streaming: } f_\alpha(t + \delta t, \mathbf{x} + \mathbf{e}_\alpha \delta t) = \tilde{f}_\alpha(t, \mathbf{x}) \quad (3.4.3)$$

where $f_\alpha^{(eq)}(t, \mathbf{x})$ is the equilibrium distribution function given as (2.4.52).

After the collision step, the distribution function in the direction from solid to fluid at the boundary node (see Figure 3.4.2) is interpolated from the post collision distribution function in the opposite direction at the first fluid node and a fictitious distribution function similar to equilibrium distribution function as

$$\tilde{f}_{\bar{\alpha}}(t, \mathbf{x}_b) = (1 - \chi) \tilde{f}_\alpha(t, \mathbf{x}_f) + \chi f_\alpha^*(t, \mathbf{x}_b), \quad (3.4.4)$$

where we assume that the boundary wall is fixed and the fictitious equilibrium distribution function $f_\alpha^*(t, \mathbf{x}_b)$ is given as

$$f_\alpha^*(t, \mathbf{x}_b) = w_\alpha \rho(t, \mathbf{x}_f) \left(1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}(t, \mathbf{x}_b)}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u}(t, \mathbf{x}_f))^2}{2c_s^4} - \frac{\mathbf{u}^2(t, \mathbf{x}_f)}{2c_s^2} \right). \quad (3.4.5)$$

$\mathbf{u}(t, \mathbf{x}_b)$ could be chosen as $\mathbf{u}(t, \mathbf{x}_f)$ when $\Delta < 1/2$ or the linear extrapolated value from the velocity on the fluid node and that on the wall when $\Delta \geq 1/2$, reading as

$$\begin{cases} \mathbf{u}(t, \mathbf{x}_b) = \mathbf{u}(t, \mathbf{x}_f) & \Delta < 1/2, \\ \mathbf{u}(t, \mathbf{x}_b) = \frac{\Delta - 1}{\Delta} \mathbf{u}(t, \mathbf{x}_f) + \frac{1}{\Delta} \mathbf{u}(t, \mathbf{x}_w) & \Delta \geq 1/2. \end{cases} \quad (3.4.6)$$

Taking the velocity $\mathbf{u}(t, \mathbf{x}_w)$ of a moving wall into consideration for the fitting formula (3.4.4), we have the modified distribution function by halfway non-equilibrium bounce back rule as

$$\tilde{f}_{\bar{\alpha}}(t, \mathbf{x}_b) = (1 - \chi) \tilde{f}_{\bar{\alpha}}(t, \mathbf{x}_f) + \chi f_\alpha^*(t, \mathbf{x}_b) + 2w_\alpha \rho(t, \mathbf{x}_f) \frac{\mathbf{u}(t, \mathbf{x}_w) \cdot \mathbf{e}_{\bar{\alpha}}}{c_s^2}, \quad (3.4.7)$$

The fitting weight χ in (3.4.4) and (3.4.7) is left to be determined by the following assumptions according to [53] [54]:

1. The time step is comparable to the first order of Knudsen number, e.g. $\delta t = O(\varepsilon)$,
2. The ratio between the space step and the characteristic length is comparable to the first order of Knudsen number, e.g. $\delta x/L = O(\varepsilon)$,
3. The ratio between characteristic velocity $U = L/T$ and the lattice velocity $c_l = \delta x/\delta t$ is comparable to the first order of Knudsen number, e.g. $U/c_l = O(\varepsilon)$,
4. The temporal variation of distribution function is comparable to the ratio between distribution function and the characteristic time, e.g. $O(\partial f_{\bar{\alpha}}/\partial t) = O(f_{\bar{\alpha}}/T)$.

By Taylor expansion, we have

$$f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_f) = f_{\bar{\alpha}}(t, \mathbf{x}_f) + \frac{\partial f_{\bar{\alpha}}(t, \mathbf{x}_f)}{\partial t} \delta t + O(\delta t^2). \quad (3.4.8)$$

Provided assumption 4, by noticing $U = L/T$, $c_l = \delta x/\delta t$ we have that

$$O\left(\frac{\partial f_{\bar{\alpha}}(t, \mathbf{x}_f)}{\partial t} \delta t\right) = f_{\bar{\alpha}}(t, \mathbf{x}_f) O\left(\frac{\delta t}{T}\right) = f_{\bar{\alpha}}(t, \mathbf{x}_f) O\left(\frac{\delta x}{L} \frac{U}{c_l}\right) = O(\varepsilon^2). \quad (3.4.9)$$

The last step is based assumptions 2 and 3. Therefore, we get the following approximation for (3.4.8) by assumption 1 and (3.4.9)

$$f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_f) = f_{\bar{\alpha}}(t, \mathbf{x}_f) + O(\varepsilon^2). \quad (3.4.10)$$

In fact, in order to obtain (3.4.10), assumption 1 and the assumption of $\partial_t f_{\bar{\alpha}}(t, \mathbf{x}_f) \leq O(\varepsilon)$ are enough, as can be seen from (3.4.8). Substituting streaming step to collision step in (3.4.2) and dividing both sides by δt , we obtain the simplified Boltzmann - BGK equation approximately as

$$(\partial_t + \mathbf{e}_\alpha \cdot \nabla) f_\alpha = -(f_\alpha - f_\alpha^{eq})/\tau\delta t, \quad (3.4.11)$$

Equation (3.4.11) together with the multiscale expansion of the distribution function

$$f_\alpha(t, \mathbf{x}) = f_\alpha^{eq}(t, \mathbf{x}) + \varepsilon f_\alpha^{(1)}(t, \mathbf{x}) + O(\varepsilon^2), \quad (3.4.12)$$

lead to

$$\varepsilon f_\alpha^{(1)}(t, \mathbf{x}) = -\tau\delta t(\partial_t + \mathbf{e}_\alpha \cdot \nabla) f_\alpha^{eq}(t, \mathbf{x}) + O(\varepsilon^2). \quad (3.4.13)$$

As a consequence, the approximation for the post streaming distribution function can be computed as

$$\begin{aligned} f_{\bar{\alpha}}(t, \mathbf{x}_f) &= f_{\bar{\alpha}}^{eq}(t, \mathbf{x}_f) - \tau\delta t(\partial_t + \mathbf{e}_{\bar{\alpha}} \cdot \nabla) f_{\bar{\alpha}}^{eq}(t, \mathbf{x}_f) + O(\varepsilon^2) \\ &= f_{\bar{\alpha}}^{eq}(t, \mathbf{x}_f) - \tau\delta t \mathbf{e}_{\bar{\alpha}} \cdot \nabla f_{\bar{\alpha}}^{eq}(t, \mathbf{x}_f) + O(\varepsilon^2) \quad (\text{by } \delta t \partial_t f_{\bar{\alpha}}^{eq}(t, \mathbf{x}_f) \leq O(\varepsilon^2)) \\ \text{by (2.4.52)} &= f_{\bar{\alpha}}^{eq}(t, \mathbf{x}_f) - \tau\delta t \rho w_{\bar{\alpha}}(\mathbf{e}_{\bar{\alpha}} \cdot \nabla \mathbf{u}(t, \mathbf{x}_f)) \cdot \mathbf{e}_{\bar{\alpha}}/c_s^2 + O(\varepsilon^2) + O(Ma^2) \\ \text{by (2.4.52)} &= f_\alpha^{eq}(t, \mathbf{x}_f) - 2\rho w_\alpha \mathbf{e}_\alpha \cdot \mathbf{u}(t, \mathbf{x}_f)/c_s^2 \quad (\bar{\alpha} \rightarrow \alpha) \\ &\quad - \tau\delta t \rho w_\alpha(\mathbf{e}_\alpha \cdot \nabla \mathbf{u}(t, \mathbf{x}_f) \cdot \mathbf{e}_\alpha)/c_s^2 + O(\varepsilon^2) + O(Ma^2). \end{aligned} \quad (3.4.14)$$

While for the post collision distribution, we have the approximation

$$\begin{aligned} \tilde{f}_{\bar{\alpha}}(t, \mathbf{x}_b) &= (1 - \chi)\tilde{f}_\alpha(t, \mathbf{x}_f) + \chi f_\alpha^*(t, \mathbf{x}_b) + 2w_\alpha \rho(t, \mathbf{x}_f) \frac{\mathbf{u}(t, \mathbf{x}_w) \cdot \mathbf{e}_{\bar{\alpha}}}{c_s^2} \\ &= (1 - \chi)(f_\alpha^{eq}(t, \mathbf{x}_f) + (1 - 1/\tau)\varepsilon f_\alpha^{(1)}(t, \mathbf{x}_f)) \\ &\quad + \chi(f_\alpha^{eq}(t, \mathbf{x}_f) + \rho w_\alpha \mathbf{e}_\alpha \cdot (\mathbf{u}(t, \mathbf{x}_b) - \mathbf{u}(t, \mathbf{x}_f))/c_s^2) \\ &\quad - 2\rho w_\alpha \mathbf{e}_\alpha \cdot \mathbf{u}(t, \mathbf{x}_w)/c_s^2 + O(\varepsilon^2) \\ &= f_\alpha^{eq}(t, \mathbf{x}_f) - (1 - \chi)(\tau - 1)\delta t \rho w_\alpha(\mathbf{e}_\alpha \cdot \nabla \mathbf{u}(t, \mathbf{x}_f) \cdot \mathbf{e}_\alpha)/c_s^2 \\ &\quad + \rho w_\alpha \mathbf{e}_\alpha \cdot (\chi(\mathbf{u}(t, \mathbf{x}_b) - \chi \mathbf{u}(t, \mathbf{x}_f) - 2\mathbf{u}(t, \mathbf{x}_w))/c_s^2 + O(\varepsilon^2)). \end{aligned} \quad (3.4.15)$$

By Taylor expansion of velocity on the wall, we have

$$\mathbf{u}(t, \mathbf{x}_w) = \mathbf{u}(t, \mathbf{x}_f) + (\mathbf{x}_w - \mathbf{x}_f) \cdot \nabla \mathbf{u}(t, \mathbf{x}_f) + O(\varepsilon^2) = \mathbf{u}(t, \mathbf{x}_f) + \Delta \delta t \mathbf{e}_\alpha \cdot \nabla \mathbf{u}(t, \mathbf{x}_f) + O(\varepsilon^2).$$

Comparing (3.4.14) and (3.4.15) when (3.4.6) holds, we have

$$\begin{cases} \chi = \frac{2\Delta - 1}{\tau - 1} & \Delta < 1/2, \\ \chi = \frac{2\Delta - 1}{\tau} & \Delta \geq 1/2. \end{cases} \quad (3.4.16)$$

This is the basic result that Filippova and Hänel obtained for the curved boundary condition, and it is worth to mention that this method is accurate up to second order, as illustrated later by benchmarks. However, the shortcoming for this method is weak stability for τ going to 1. Mei et al. [54] suggested to take $\mathbf{u}(t, \mathbf{x}_b) = \mathbf{u}(t, \mathbf{x}_{ff})$ instead of $\mathbf{u}(t, \mathbf{x}_f)$ when $\Delta < 1/2$, which leads to a more stable method for curved boundary condition,

$$\begin{cases} \chi = \frac{2\Delta - 1}{\tau - 2} & \Delta < 1/2, \\ \chi = \frac{2\Delta - 1}{\tau} & \Delta \geq 1/2. \end{cases} \quad (3.4.17)$$

3.4.2 Bouzidi-Firdaouss-Lallemand boundary dynamics

The philosophy behind this method [55] is to apply the bounce back scheme plus the linear or quadratic interpolation for distribution function directly instead of reconstructing the distribution function by the interpolated velocity as that in FH boundary dynamics.

Take Figure 3.4.2 for illustration, for non-slip boundary where the velocity on the wall vanishes, direct bounce back scheme for distribution function (zeroth order of interpolation) leads to

$$f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_f) = \tilde{f}_{\alpha}(t, \mathbf{x}_f), \quad (3.4.18)$$

and by linear interpolation we have

$$\begin{aligned} f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_f) &= 2\Delta \tilde{f}_{\alpha}(t, \mathbf{x}_f) + (1 - 2\Delta) \tilde{f}_{\alpha}(t, \mathbf{x}_{ff}) \text{ for } \Delta < 1/2, \\ f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_f) &= \frac{1}{2\Delta} \tilde{f}_{\alpha}(t, \mathbf{x}_f) + \frac{2\Delta - 1}{2\Delta} \tilde{f}_{\bar{\alpha}}(t, \mathbf{x}_f) \text{ for } \Delta \geq 1/2, \end{aligned} \quad (3.4.19)$$

For moving boundary where the velocity \mathbf{u}_w is not zero we have by Taylor expansion up to the first order that

$$\mathbf{u}(t, \mathbf{x}_f) = \mathbf{u}(t, \mathbf{x}_w) + (\mathbf{x}_f - \mathbf{x}_w) \cdot \nabla \mathbf{u}(t, \mathbf{x}_w). \quad (3.4.20)$$

An additional term added to the interpolated result for distribution function (3.4.19)

due to the present velocity, is given as

$$\begin{aligned}\delta f_{\bar{\alpha}}^{(1)}(t, \mathbf{x}_f) &= -2\rho(t, \mathbf{x}_f) w_\alpha \mathbf{e}_\alpha \cdot \mathbf{u}(t, \mathbf{x}_w) / c_s^2 \text{ for } \Delta < 1/2, \\ \delta f_{\bar{\alpha}}^{(1)}(t, \mathbf{x}_f) &= -\frac{1}{\Delta} \rho(t, \mathbf{x}_f) w_\alpha \mathbf{e}_\alpha \cdot \mathbf{u}(t, \mathbf{x}_w) / c_s^2 \text{ for } \Delta \geq 1/2.\end{aligned}\quad (3.4.21)$$

where the density at the first fluid node $\rho(t, \mathbf{x}_f)$ is obtained from interpolation from its neighbors, up to second order corresponding to the interpolation rule for velocity.

Remark 3.4.1 *By quadratic interpolation we obtain more accurate distribution function as*

$$\begin{aligned}f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_f) &= \Delta(2\Delta + 1)\tilde{f}_\alpha(t, \mathbf{x}_f) + (2\Delta + 1)(1 - 2\Delta)\tilde{f}_\alpha(t, \mathbf{x}_{ff}) \\ &\quad - \Delta(1 - 2\Delta)\tilde{f}_\alpha(t, \mathbf{x}_{fff}) \text{ where } (\mathbf{x}_{fff} = \mathbf{x}_f - 2\mathbf{e}_\alpha) \text{ for } \Delta < 1/2, \\ f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_f) &= \frac{1}{\Delta(2\Delta + 1)}\tilde{f}_\alpha(t, \mathbf{x}_f) + \frac{2\Delta - 1}{\Delta}\tilde{f}_{\bar{\alpha}}(t, \mathbf{x}_f) \\ &\quad + \frac{1 - 2\Delta}{1 + 2\Delta}\tilde{f}_{\bar{\alpha}}(t, \mathbf{x}_{ff}) \text{ for } \Delta \geq 1/2.\end{aligned}\quad (3.4.22)$$

And the corresponding additional term for moving boundary is

$$\begin{aligned}\delta f_{\bar{\alpha}}^{(2)}(t, \mathbf{x}_f) &= -2\rho(t, \mathbf{x}_f) w_\alpha \mathbf{e}_\alpha \cdot \mathbf{u}(t, \mathbf{x}_w) / c_s^2 \text{ for } \Delta < 1/2, \\ \delta f_{\bar{\alpha}}^{(2)}(t, \mathbf{x}_f) &= -\frac{2}{\Delta(2\Delta + 1)}\rho(t, \mathbf{x}_f) w_\alpha \mathbf{e}_\alpha \cdot \mathbf{u}(t, \mathbf{x}_w) / c_s^2 \text{ for } \Delta \geq 1/2.\end{aligned}\quad (3.4.23)$$

3.4.3 Uniform interpolation boundary dynamics

As FH and Mei's boundary dynamics apply the idea of fitting the distribution right after collision with different formulas according to the ratio $\Delta \geq 1/2$ or $\Delta < 1/2$ given in (3.4.1), Yu et al. [57] proposed a uniform fitting method with the same formula for different ratio. This method is based on the distribution function after streaming step instead of collision step. Since $f_\alpha(t + \delta t, \mathbf{x}_b)$ and $f_\alpha(t + \delta t, \mathbf{x}_f)$ are known after the streaming step, the unknown post streaming distribution function $f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_f)$ is interpolated as follows.

Firstly, the fictitious distribution function on the wall is interpolated as

$$f_\alpha(t + \delta t, \mathbf{x}_w) = f_\alpha(t + \delta t, \mathbf{x}_f) + \Delta(f_\alpha(t + \delta t, \mathbf{x}_b) - f_\alpha(t + \delta t, \mathbf{x}_f)). \quad (3.4.24)$$

Suppose that the boundary is fixed s.t. $\mathbf{u}(t, \mathbf{x}_w) = 0$, we have

$$f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_w) = f_\alpha(t + \delta t, \mathbf{x}_w). \quad (3.4.25)$$

After which the unknown distribution function $f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_f)$ is interpolated as

$$f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_f) = f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_w) + \frac{\Delta}{\Delta + 1} (f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_{ff}) - f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_w)). \quad (3.4.26)$$

The uniform interpolation method provides also the second order accuracy despite its simplicity. However, when the boundary wall is moving, where $\mathbf{u}(t, \mathbf{x}_w) \neq 0$, the distribution function on the wall is modified as

$$f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_w) = f_{\alpha}(t + \delta t, \mathbf{x}_w) - 2\rho(t, \mathbf{x}_w) w_{\alpha} \mathbf{e}_{\alpha} \cdot \mathbf{u}(t + \delta t, \mathbf{x}_w) / c_s^2. \quad (3.4.27)$$

Unfortunately, the density on the wall $\rho(t, \mathbf{x}_w)$ is not known but has to be interpolated or extrapolated from their neighbors. To avoid large computation, Yu suggested to use $\rho(t, \mathbf{x}_f)$ as $\rho(t, \mathbf{x}_w)$, which is valid for incompressible flows.

3.4.4 Halfway bounceback boundary bynamic

Beyond all the methods listed above for curved boundary in the sense of simplicity, we borrow the idea of halfway bounceback rule from straight boundary for our curved boundary. This yields the stair wise boundary dynamics for curved boundary as can be seen from Figure 3.4.2. Formally, we have, for all the unknown distributions in the direction α

$$f_{\bar{\alpha}}(t + \delta t, \mathbf{x}_f) = f_{\alpha}(t + \delta t, \mathbf{x}_f) - 2\rho(t, \mathbf{x}_w) w_{\alpha} \mathbf{e}_{\alpha} \cdot \mathbf{u}(t + \delta t, \mathbf{x}_w) / c_s^2, \quad (3.4.28)$$

where we assume that $\rho(t, \mathbf{x}_w) = \rho(t, \mathbf{x}_f)$, which is valid for incompressible flow.

Remark 3.4.2 *It's the easiest to implement and achieve the first order of accuracy. And it is also one of the most popular methods for imposing curved boundary conditions, taking the particle suspension [77] and porous media [78] for example.*

3.4.5 Regularized interpolation/extrapolation boundary dynamics

A different method based on Distribution Reconstruction for curved boundary was recently investigated by Verschaeve et al. [58]. The idea behind this strategy is to accurately interpolate or extrapolate velocity and density as well as the derivative of the velocity at first, then reconstruct the distribution function by the regularized method developed by Lätt et al. [59] for the straight boundary conditions.

In order to identify the interpolation and extrapolation nodes, three different configuration are suggested according to the location of the boundary nodes as shown

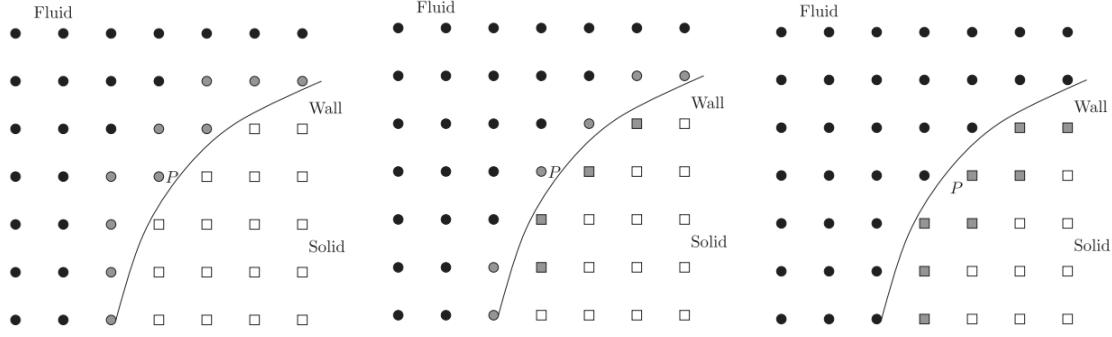


Figure 3.4.3: three different configurations, black for fluid nodes, white for solid nodes, gray for boundary nodes. Left: boundary nodes locate in the fluid domain; Middle: boundary nodes sit on both the solid domain and the fluid domain, according to the distance to the boundary wall; Right: boundary nodes lie on the solid domain. P is one node taken for example. The illustration figures are taken from [58]

in Figure 3.4.3. Take the distance from the boundary nodes to the boundary wall into consideration, we may draw the conclusion that the Middle one of Figure 3.4.3 would give more accurate result since the interpolation or extrapolation for boundary condition is more precise with smaller distance.

To reconstruct the distribution function on the boundary nodes, density ρ , velocity \mathbf{u} as well as strain rate tensor S are needed but remain unknown. So we have to interpolate or extrapolate them by the known quantities of their neighbors. Without loss of generality, let's take the Left configuration of Figure 3.4.3 for example to interpolate the velocity and compute the corresponding derivative of the velocity, as shown in Figure 3.4.4.

Step 1 : \mathbf{u} . First of all, the direction to interpolate the velocity is decided upon the smallest angle between the link of the nodes ($P - H_0$) and the normal direction of the boundary wall, see left of Figure 3.4.4. Velocity along this direction on nodes H_1, H_2 as well as the given boundary condition (Dirichlet boundary condition) on H_0 are known before collision step as $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_0$. The interpolation is given by quadratic Lagrange interpolation polynomials as follows. Suppose the distance from H_2, H_1, P to H_0 are denoted as d_{H_2}, d_{H_1}, d_P , so the velocity on node P is given as

$$\mathbf{u}(d_P) = \mathbf{u}_0 l_0(d_P) + \mathbf{u}_1 l_1(d_P) + \mathbf{u}_2 l_2(d_P), \quad (3.4.29)$$

where $l_0(d_P), l_1(d_P), l_2(d_P)$ are the quadratic Lagrange polynomials evaluated at d_P , explicitly reading as

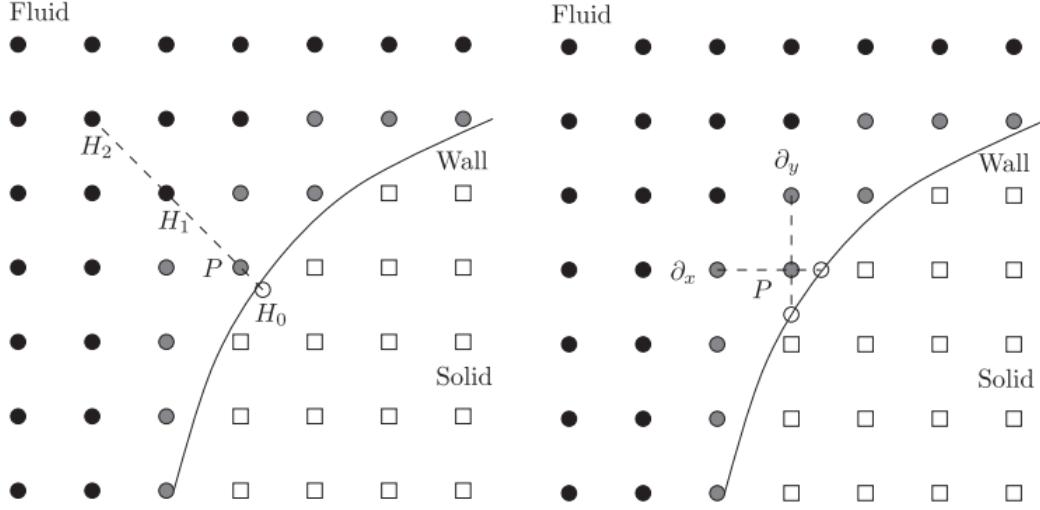


Figure 3.4.4: Left: interpolate the velocity on the boundary node P by its neighbors on the fluid domain and the boundary wall; Right: compute the velocity derivative by finite difference method for node P . The illustration figures are taken from [58]

$$\begin{cases} l_0(d_P) = \frac{(d_P - d_{H_1})(d_P - d_{H_2})}{d_{H_1}d_{H_2}}, \\ l_1(d_P) = \frac{d_P(d_P - d_{H_2})}{d_{H_1}(d_{H_1} - d_{H_2})}, \\ l_2(d_P) = \frac{d_P(d_P - d_{H_1})}{d_{H_2}(d_{H_2} - d_{H_1})}, \end{cases} \quad (3.4.30)$$

which can be computed once and for all at the beginning for a fixed boundary.

Step 2: S . For the second step, the strain rate tensor S , given as the derivative of velocity is computed by weighted finite difference method either to first order (3.2.11) or second order (3.2.12) with the known velocity in the fluid domain, on the boundary or the velocity we obtained by the interpolation, as shown in Figure 3.4.4.

Step 3: ρ . Lastly, the interpolation or extrapolation for density is a bit more complicated since the unknown direction for the curved boundary is not uniform as that for the straight boundary but vary according to the specific curved boundary. Hence, we can use interpolation or extrapolation for density as in other methods aforementioned, just like what we did for the velocity. Otherwise, the unknown distribution functions need to be taken into account. Denote by \mathcal{F} , \mathcal{B} and \mathcal{S} as the index for the direction from P to the fluid nodes, other boundary nodes and the solid nodes respectively. Take P in Figure 3.4.4 for example, we have for the D2Q9 lattice structure in Figure A.0.3

that

$$\mathcal{F} = \{6\}; \quad \mathcal{B} = \{2, 3, 5, 7\}; \quad \mathcal{S} = \{1, 4, 8\}.$$

Recall the distribution function separated into the equilibrium part and non-equilibrium part,

$$\begin{aligned} f_i(t, \mathbf{x}) &= f_i^{(eq)}(\rho, \mathbf{u}) + f_i^{(neq)}(\rho, S) \\ &= w_i \rho \left(1 + \frac{\boldsymbol{\xi}_i \cdot \mathbf{u}}{c_s^2} + \frac{1}{2c_s^4} Q_i : \mathbf{u}\mathbf{u} \right) - \frac{\rho w_i \tau}{c_s^2} Q_i : S \\ &\triangleq \rho(f_i^{(eq)}(1, \mathbf{u}) + f_i^{(neq)}(1, S)) \\ &\triangleq \rho(g_i^{(eq)}(1, \mathbf{u}) + g_i^{(neq)}(1, S)) \triangleq \rho g_i(t, \mathbf{x}). \end{aligned} \quad (3.4.31)$$

Therefore, by the known distribution function f_i and the interpolated velocity and strain rate tensor, the density could be solved from the above equation. In order to avoid the instability as stated in [11], we use the direction ensemble as \mathcal{K} , which is of directions opposite to those of \mathcal{F} . In our example above, it is 8. Since there are one or more elements in \mathcal{K} , we have correspondingly the same number of equations as (3.4.31). Therefore, we have the following different approximations for the density as:

$$1. \text{ Least Squares Solution: } \rho = \frac{\sum_{i \in \mathcal{K}} f_i g_i}{\sum_{i \in \mathcal{K}} g_i^2}. \quad (3.4.32)$$

Least square method is the most efficient method in the sense of using the known information, similar to the maximum likelihood method.

$$2. \text{ Weighted Average Solution: } \rho = \frac{1}{|\mathcal{K}|} \sum_{i \in \mathcal{K}} \frac{f_i}{g_i}. \quad (3.4.33)$$

Weighted average method is an naive but efficient method in the sense of using plenty of known information, analogy to law of large numbers.

$$3. \text{ Global Average Solution: } \rho = \frac{\sum_{i \in \mathcal{K}} f_i}{\sum_{i \in \mathcal{K}} g_i}. \quad (3.4.34)$$

Global average method coincides with the definition of density, achieve the same density when the boundary is straight according to Chopard et al. (see[69]).

$$4. \text{ Mass Conservation Solution: } \rho = \frac{\sum_{i \in \mathcal{K}} f_i}{\sum_{i \in \mathcal{K}} (1 - 1/\tau) g_i^{(neq)} + g_i^{(eq)}}. \quad (3.4.35)$$

Mass conservation method tries to conserve the in and out streaming mass, leading to the homogeneous Neumann boundary conditions for straight boundary, according to Verschaeve. However, it's not always valid for the curved boundary because the

unknown distribution functions are not always the same as those in straight boundary (see [71]).

With the density $\rho(t, \mathbf{x})$, velocity $\mathbf{u}(t, \mathbf{x})$ and the strain rate tensor $S(\mathbf{u})$ at hand, we can construct the distribution functions in all direction on the boundary nodes by the formula given in (3.4.31). It differs from the approaches mentioned above in that this approach not only approximates the unknown distribution function but also replace all the known distribution functions by the reconstruction.

Remark 3.4.3 As the reader might wonder, this so called regularized interpolation or extrapolation boundary dynamics actually apply the idea of finite difference scheme to obtain the velocity gradient in the right figure of Figure 3.4.4. Concerning this point, the boundary dynamics should be called finite difference interpolation or extrapolation boundary dynamics instead of regularized method in the original paper [58]. While for regularized interpolation or extrapolation boundary dynamics we can apply halfway bounceback rule for non-equilibrium part of the unknown distribution functions or other methods such as uniform interpolation boundary dynamics firstly, and then reconstruct the full distribution functions the same as for straight boundary by (3.2.13).

3.5 Benchmarks with curved boundary

Taylor Couette flow is taken as the benchmark for investigating the numerical properties such as accuracy, efficiency of the different boundary dynamics introduced in the last section for curved boundary. And a more practical application - the blood flow in the left arm based on curved boundary is tested in this section.

3.5.1 Laminar Taylor Couette flow in 2D

Laminar Taylor Couette flow [58] is in between two cylinders, where the interior cylinder with radius r_1 is rotating at a constant angular speed W while the exterior cylinder with radius r_2 is at rest and the pressure is null, as shown in Figure 3.5.1. More specifically, the problem for Taylor Couette flow is given as (2.6.10) with the following boundary conditions:

$$\begin{cases} \mathbf{u}(r_1) = \mathbf{u}_0 & \text{on interior boundary,} \\ \mathbf{u}(r_2) = 0, \quad p(r_2) = 0, & \text{on exterior boundary.} \end{cases} \quad (3.5.1)$$

where the velocity on the interior boundary is $\mathbf{u}_0 = u_0 \mathbf{e}_\theta = r_1 W$, define $\beta = r_1/r_2$.

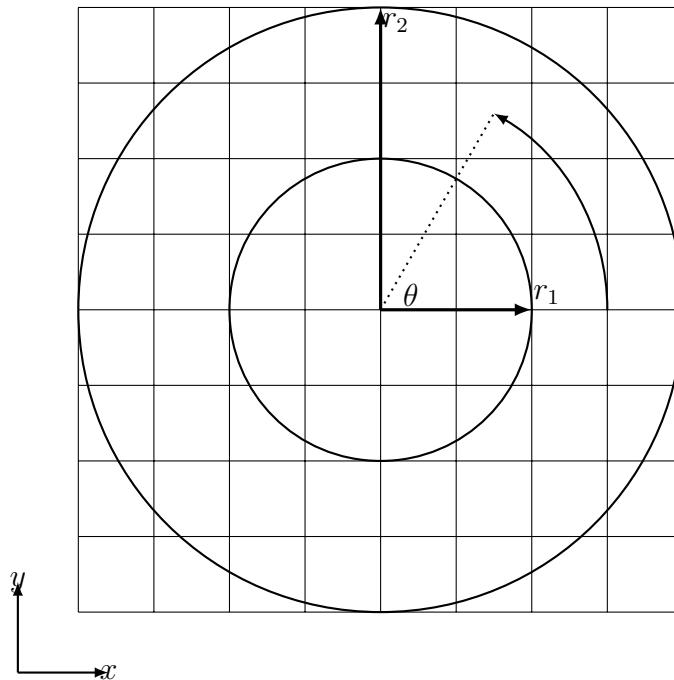


Figure 3.5.1: sketch for Taylor Couette flow in between two cylinders, with radius of interior cylinder r_1 and r_2 for the exterior cylinder, steady flow counter-clockwise

The analytical solution of velocity for Taylor Couette flow is given as

$$\mathbf{u}(r) = \frac{u_0\beta}{1-\beta^2} \left(\frac{r_2}{r} - \frac{r}{r_2} \right) \mathbf{e}_\theta, \quad (3.5.2)$$

and the pressure is

$$p(r) = \frac{\rho u_0^2}{2} \left(\frac{\beta}{1-\beta^2} \right)^2 \left(\frac{r^2}{r_2^2} - \frac{r_2^2}{r^2} - 4 \ln \left(\frac{r}{r_2} \right) \right). \quad (3.5.3)$$

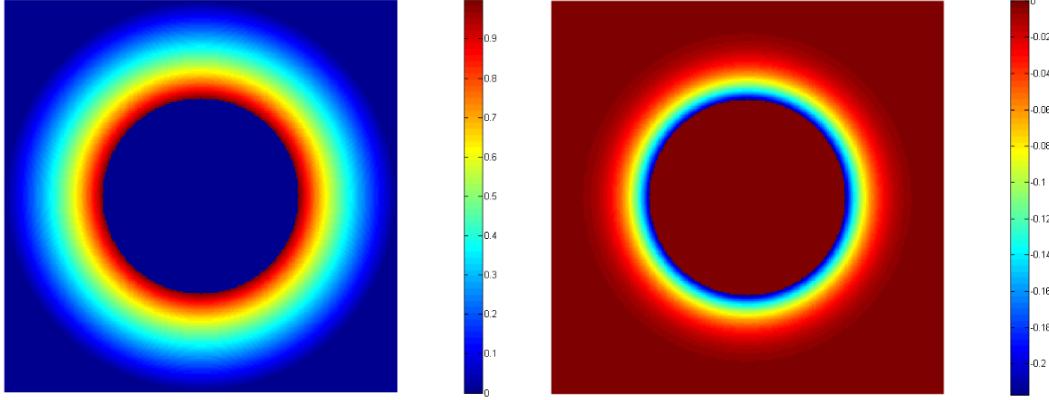


Figure 3.5.2: numerical solution for Taylor Couette flow, viscosity $\nu = 0.01$

The parameters for the simulation are chosen as follows:

- The computational domain is set as $Lx = Ly = 2$ and $r_1 = 0.5, r_2 = 1$ so that $\beta = r_1/r_2 = 0.5$, spatial step is $\delta x = 1/64$, time step is $\delta t = \delta x^2$,
- The boundary conditions are prescribed by velocity on the exterior boundary 0 and on the interior boundary $u_0 \mathbf{e}_\theta$ and $u_0 = 1$,
- Viscosity is 0.01, equivalently, the Reynolds number is given as $(r_2 - r_1)u_0/\nu = 50$,
- Initialize the velocity and pressure by the analytical solution given in (3.5.2) and (3.5.3),
- As for the curved interior and exterior boundary, Halfway non-equilibrium bounce back(HalfWay), uniform interpolation/extrapolation (Uniform), Bouzidi-Firdaouss-Lallemand (BFL), FH et Mei (FHetMei) as well as the regularized interpolation/extrapolation (Regularized) boundary dynamics are applied.

With the error defined in (3.3.3), the comparison for the five methods is as shown in Figure 3.5.3 while the CPU time cost by the five methods is shown in Figure 3.5.4.

From Figure 3.5.3 we can tell that both the halfway non-equilibrium bounceback boundary dynamics and the regularized boundary dynamics which also apply the

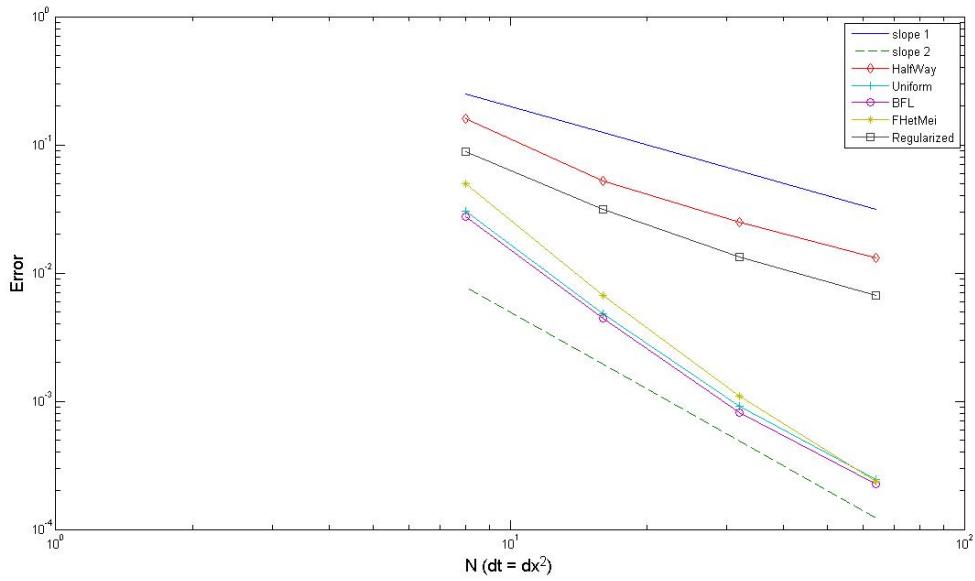


Figure 3.5.3: comparison of numerical error by different boundary dynamics

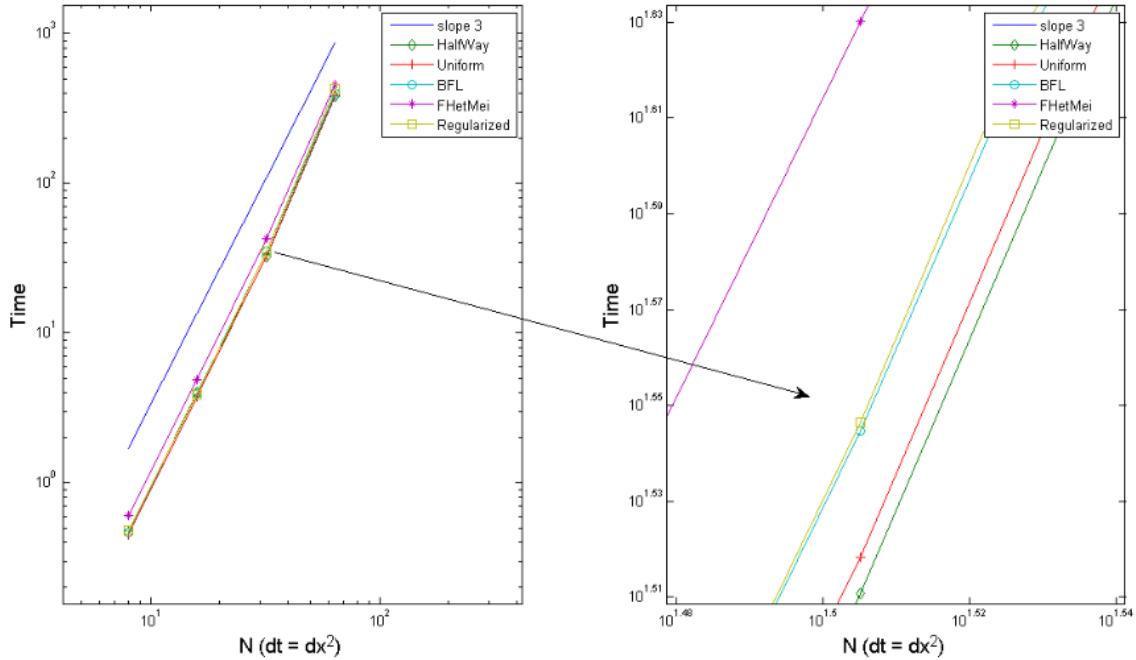


Figure 3.5.4: comparison of CPU time by the five methods for curved boundary

halfway bounceback rule tend to the first order of accuracy, while the uniform method, BFL method as well as FH et Mei method converge to the second order of accuracy when it comes to refine time step and spatial step by the relation $\delta t = \delta x^2$, and the

best method to tackle the curved boundary problem is the BFL method (slightly better than uniform method), which is not only the most accurate, especially when the mesh is very fine, but also the most efficient with respect to the convergence order.

Meanwhile, from Figure 3.5.4 about the CPU time cost for different methods, we see that FH et Mei method is more time consuming than the other four methods, and halfway non-equilibrium bounceback method is the best one with regard to computational time as expected for its simplicity.

3.5.2 An example of blood flow in 2D

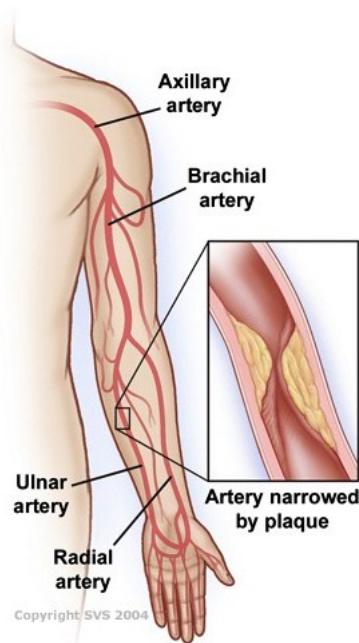


Figure 3.5.5: sketch for blood flow in the artery of the left arm with plaque enlarged

Lattice Boltzmann method has been widely applied in many fields, including the complicated medical simulation for cardiovascular flow. In order to have a precise simulation, high spatial resolution is needed. Consequently, large scale computation is inevitable and prohibitive for most of the conventional numerical methods. Lattice Boltzmann method thereby stands out as an attractive algorithm because of its essential merit of parallelism for computation, easy to implement as well as its advantage of providing pressure directly instead of solving further Poisson problem for every time step. An interesting multiscale simulation of heart-circulation system by lattice Boltzmann method on the platform of IBM Blue Gene/P was achieved recently [84]. It's worth to mention that in this paper, the authors used Zou and He boundary dynamics

for inlet and outlet boundary, where a prescribed velocity was imposed on the inlet boundary and a zero pressure difference from the inlet boundary for the outlet boundary is maintained. While for the non-slip boundary wall of the blood vessel, a halfway bounce back rule was applied.

For a more realistic application of lattice Boltzmann method, we take the blood flow in the left arm for example, as shown in Figure 3.5.5. First of all, a rough sketch of the geometry of the blood vessel in 2D is drawn according to the shape of a planar photo of the blood vessel of the left arm, see Figure 3.5.6. In the second step, we keep part of the artery with big radius in sketch while neglecting the artery with small radius as well as capillary. The artery narrowed down due to plaque is taken into consideration by narrowing the vessel. The mathematical problem is described by the incompressible Navier-Stokes equations as

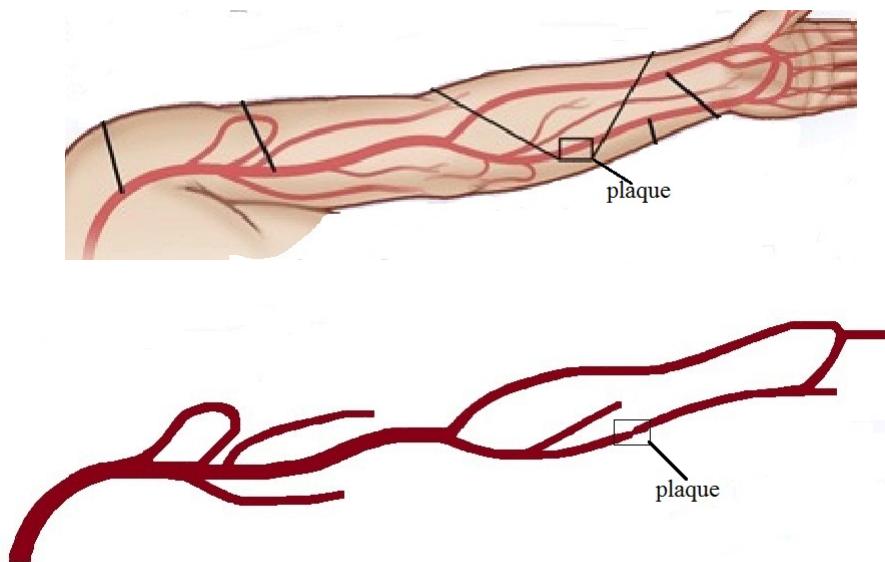


Figure 3.5.6: Computational domain is taken based on the rough sketch for blood vessel of the left arm; the part of vessel within the rectangle is narrowed down due to plaque

$$\left\{ \begin{array}{ll} \nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} = \mathbf{0} & \text{on the outlet boundaries,} \\ \mathbf{u}(t, \mathbf{x}) = \phi(t, \mathbf{x}) & \text{on the inlet boundary,} \\ \mathbf{u}(t, \mathbf{x}) = \mathbf{0} & \text{on the wall of blood vessels,} \\ \mathbf{u}(0, \mathbf{x}) = \mathbf{0} \quad p(0, \mathbf{x}) = 0 & \text{in } \Omega \end{array} \right. \quad (3.5.4)$$

The computational domain is taken from pixels of the top picture in 3.5.6 with rough measure $17.9\text{cm} \times 46.2\text{cm}$ and discretized into a mesh of 358×924 . Spatial step

and time step are given as $\delta x = 0.1$ and $\delta t = \delta x^2/10 = 0.001$ respectively. Blood viscosity is set as $0.035 g/(cm \cdot s)$ [87]. Null velocity and pressure are set as the initial condition.

As for the boundary condition according to [85], homogeneous Neumann boundary conditions are applied to the five outlet boundaries, whereas Dirichlet boundary conditions with the prescribed periodic velocity $\phi(t, \mathbf{x})$ distributed in a parabolic way with peak in the middle and zero at the edge are imposed to the inlet boundary, as shown in Figure 3.5.7 and 3.5.8. The inlet boundary conditions is prescribed by given peak periodic velocity within one period of 1s as shown in Figure 3.5.9, while a homogeneous Neumann boundary conditions is imposed on the outlet boundary. Halfway bounce back rule for non-equilibrium distribution is applied for the curved vessel wall, while regularized reconstruction of distribution function is conducted on the inlet and outlet boundary.

Remark 3.5.1 *Halfway bounce back boundary dynamics is applied for the curved vessel wall because it can handle complicate curved boundary without much effort spent on interpolation and extrapolation. In this sense, it is especially useful for the moving boundaries and also widely used for large scale parallel computations.*

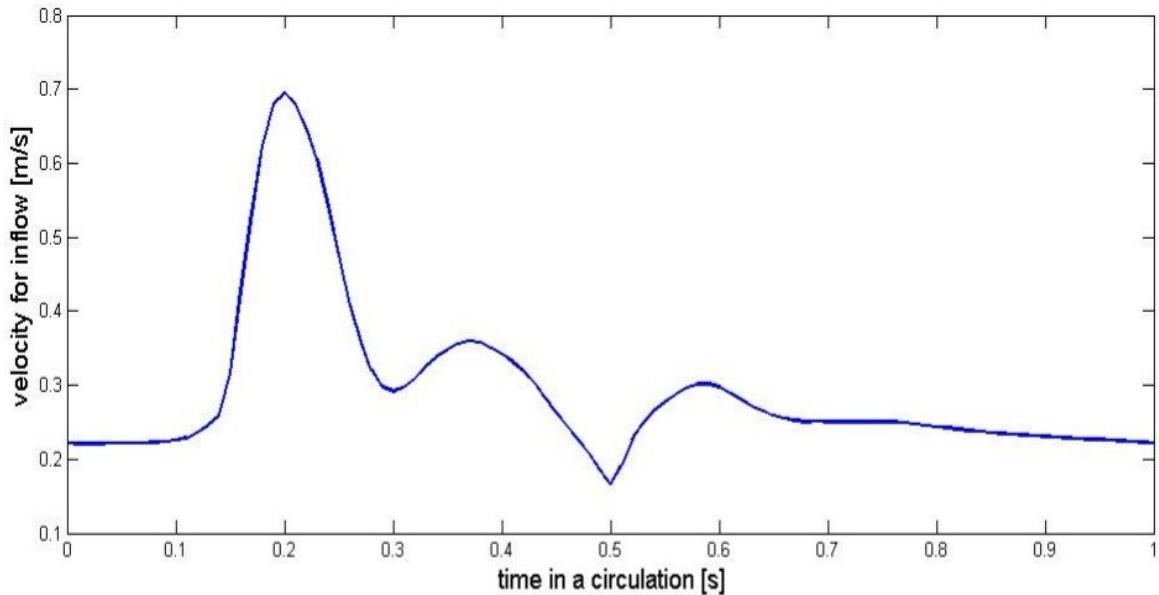


Figure 3.5.7: blood velocity within a period of one second

At the beginning of the simulation, the pressure and velocity is shown in Figure 3.5.9. After 100 periods at time 100s, the blood displays a periodic flow according to the prescribed velocity given in Figure 3.5.7 throughout the entire computational domain.

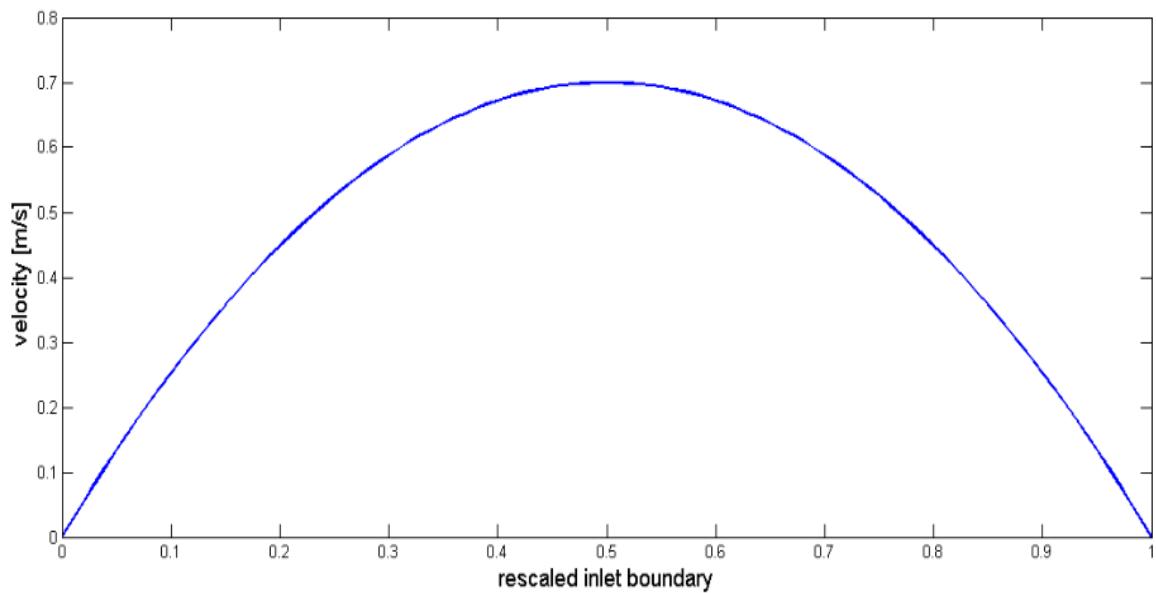


Figure 3.5.8: parabolic distribution of velocity on the inlet boundary at time $t = 0.2s$

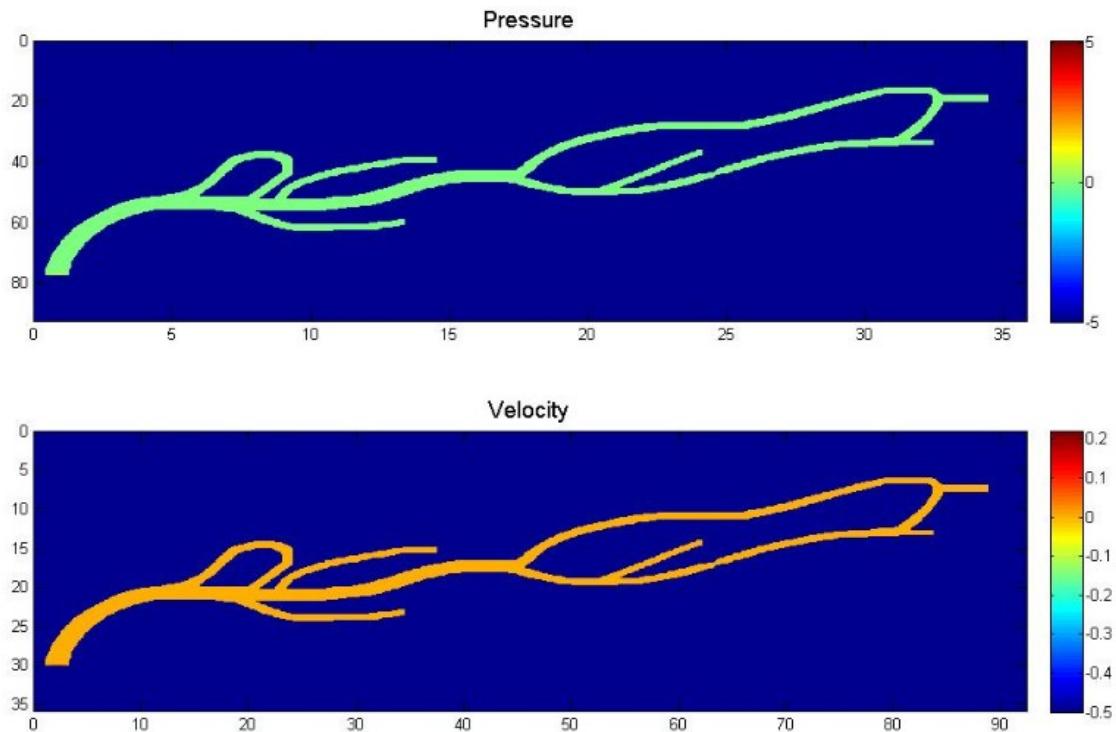


Figure 3.5.9: simulation for pressure and velocity in the blood vessel at time $t = 0$

Figure 3.5.10 to 3.5.13 show the pressure and velocity within a period when the blood flow becomes periodic with between the 10th period and the 11th period.

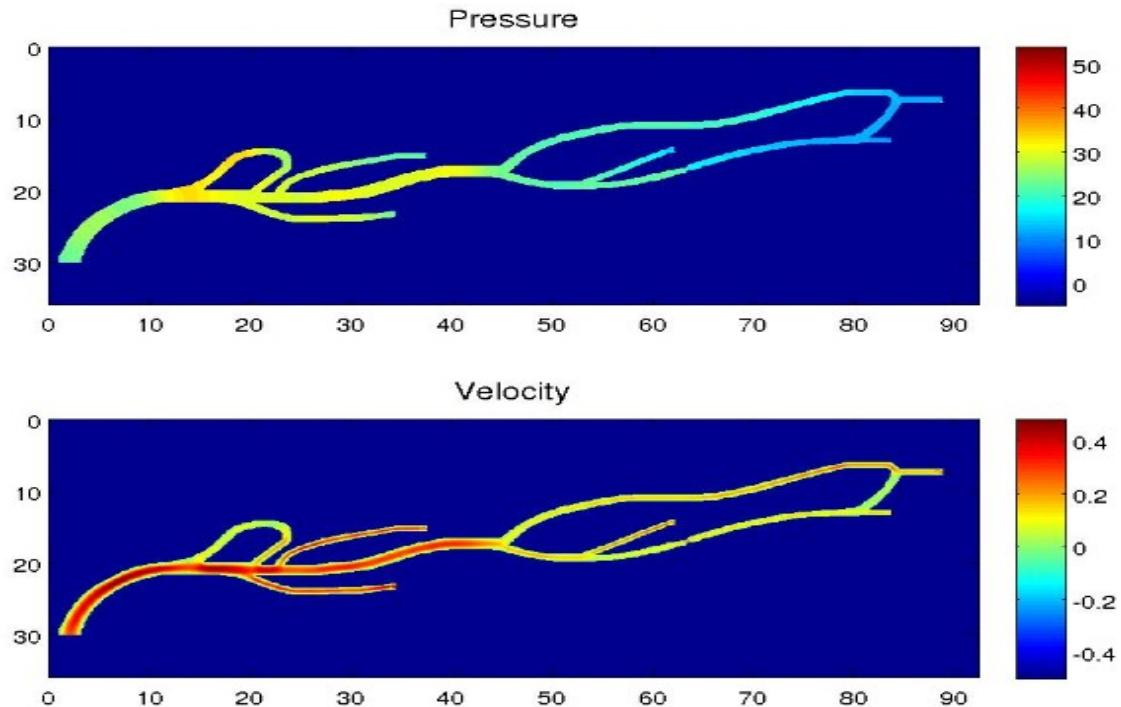


Figure 3.5.10: simulation for pressure and velocity in the blood vessel at time $t = 10s$

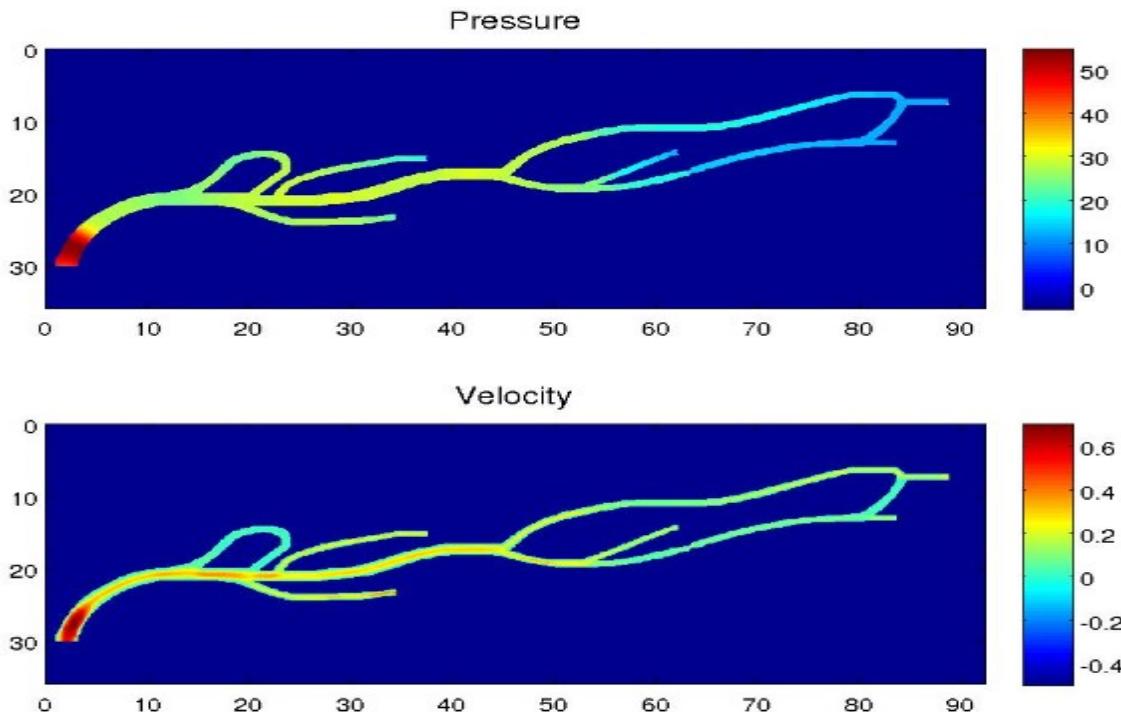


Figure 3.5.11: simulation for pressure and velocity in the blood vessel at time $t = 10.25s$

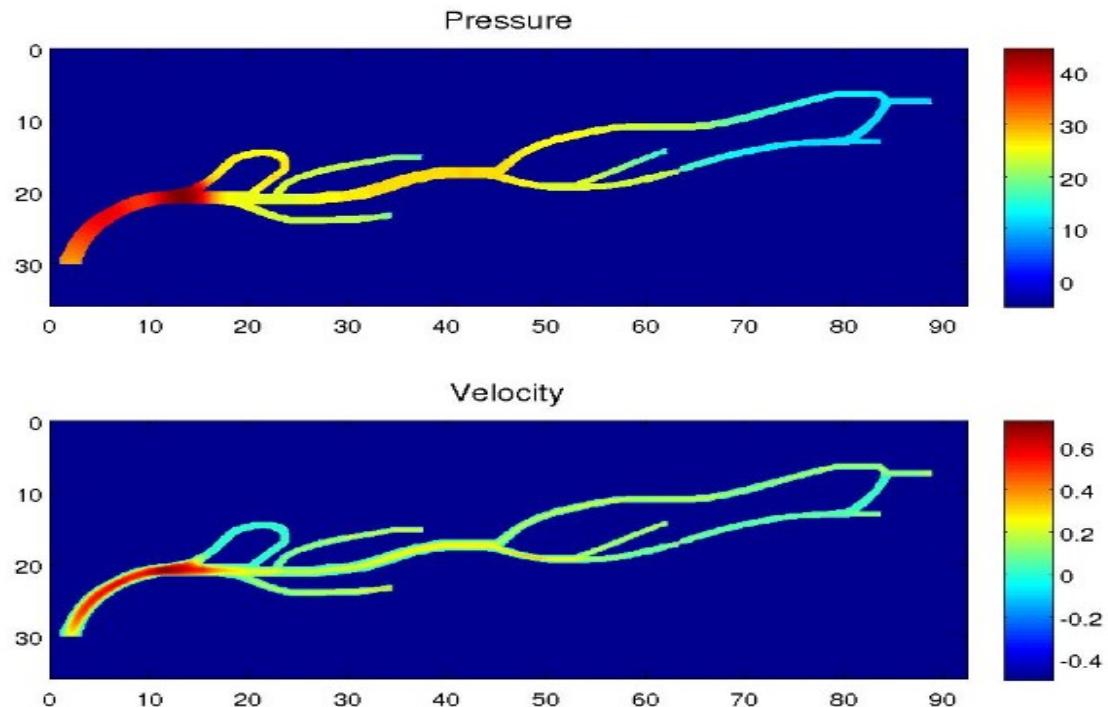


Figure 3.5.12: simulation for pressure and velocity in the blood vessel at time $t = 10.5s$

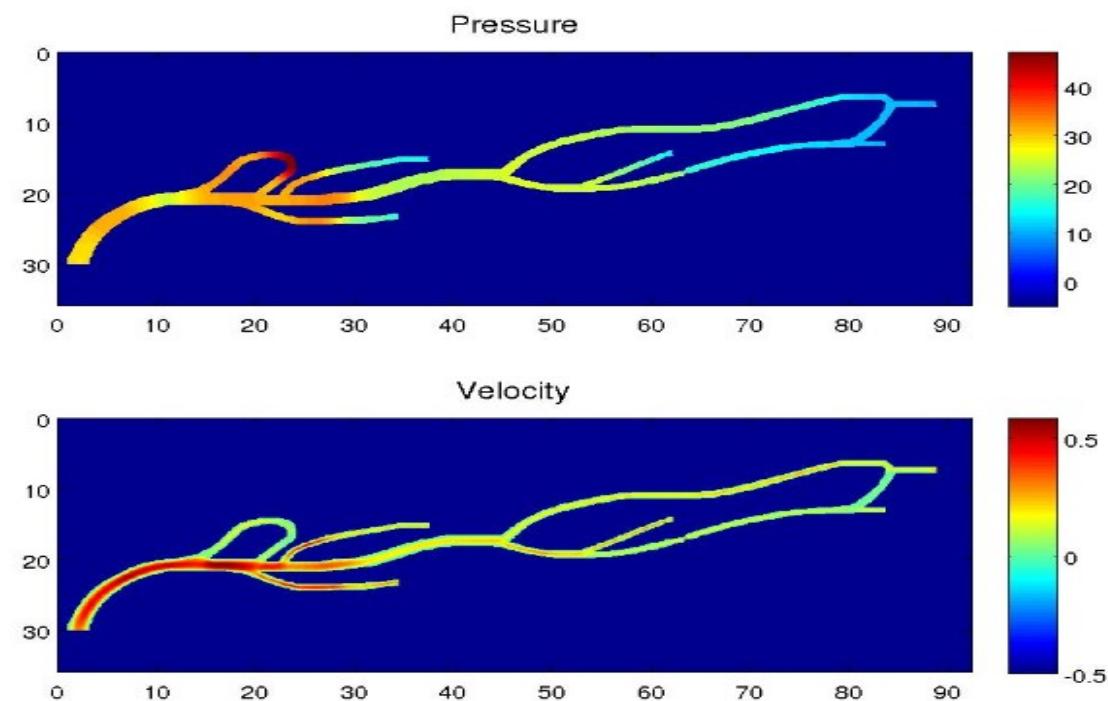


Figure 3.5.13: simulation for pressure and velocity in the blood vessel at time $t = 10.75s$

3.6 Conclusion

Imposing boundary conditions accurately and stably is crucial for lattice Boltzmann method. In this section, the difficulties for imposing boundary conditions on straight boundary and curved boundary were identified and analyzed at first. Then, different boundary dynamics for imposing Dirichlet boundary conditions on both straight boundary and curved boundary were investigated.

For straight boundaries, fullway bounce back rule and halfway bounce back rule for both equilibrium and non-equilibrium distribution functions as well as Zou-He boundary dynamics are examined in the family of “Distribution Modification”. While finite difference dynamics and regularized dynamics were considered in the family of “Distribution Reconstruction”. All of them except fullway bounce back rule produced second order of accuracy dependent on spatial discretization. Comparably, finite difference dynamics is the most stable approach.

For curved boundary, interpolation and extrapolation were needed to achieve second order of accuracy. Filippova-Hänel boundary dynamics and its improved version Mei boundary dynamics were derived and tested along with Bouzidi-Firdaouss-Lallemand boundary dynamics, uniform boundary dynamics. While the simplest non-equilibrium halfway bounce back dynamics and regularized boundary dynamics achieve first order of accuracy without any interpolation or extrapolation.

Chapter 4

Multiblock and multigrid lattice Boltzmann method

4.1 Introduction

By the previous benchmarks including Poiseuille flow with straight boundary as well as Taylor-Couette flow with curved boundary, we have shown that lattice Boltzmann method has a good performance on uniform lattice in a single block in Cartesian coordinate. However, in real application this kind of mesh does not always meet our requirement in an efficient way, either in the sense of multiscale physical problem, memory requirement or for the sake of parallel computation.

Consider for example blood flow in the artery of the left arm as shown in Figure 3.5.5, the physical geometry is much more complicated beyond the descriptive ability of uniform lattice in one single block. On the one hand, there are many branches of blood vessels sparsely distributed with relatively arbitrary geometry. Therefore, a uniform lattice covering the whole artery would cost considerable computer memory for the region without artery passing by. The recipe to deal with this problem is to design a certain number of blocks according to the shape of the artery and the efficiency with respect to parallel computation. On the other hand, the radius of axillary arteries or radial arteries are larger than that of capillaries or the region of arteries narrowed by plaque (see for example in Figure 3.5.5). Hence, a uniform lattice would be either too coarse for capillary to capture the right behavior of the flow or too fine for artery so that simulation will become considerable time consuming. This problem is tackled by grid refinement or multigrid method, where hierachic meshes are employed.

While from the computational point of view, mesh generated for the entire domain has to be refined accordingly in order to obtain more precise simulation, for which heavy computation can not be afforded by a single computer processor. It is also for this critical reason that the techniques of multiblock and multigrid have been developed

for lattice Boltzmann method. It's worth to mention that the multiblock and multigrid method can be regarded as the domain decomposition method for conventional methods such as finite element method. Therefore, some of the basic ideas from domain decomposition method, such as with or without overlapping on the interface of two adjacent subdomains, consistent condition on the interface such as mass conservation and momentum conservation and so on, could also be applied to the multiblock and multigrid lattice Boltzmann method.

The difficulty of multiblock and multigrid lattice Boltzmann method comes from the communication of information between the neighboring blocks and different grids. The information exchanged on the interface, similar to that for boundary condition, could either be distribution function in the kinetic level or macroscopic variables such as density and velocity in the macro level. Since a direct modification of distribution function is easier and faster than reconstruction of distribution function from macroscopic variables, most of the block connection and grid refinement techniques have been developed in the kinetic level. The first grid refinement model, so far as we know, is the one developed by Filippova and Hänel [53]. In this paper they proposed a scheme to modify the distribution functions on the interface in all directions according to the ratio of refinement as well as relaxation time immediately after collision step. In [79], Lin and Lai suggested a scheme living on composite grids where distribution functions are directly exchanged, using coarse to fine grid interpolation for passing information. This scheme was improved by Dupuis and Chopard [80] for its inaccuracy, and they provided a more stable and accurate scheme by not only taking proper interpolation into consideration but also touching the non-equilibrium part of distribution function.

In the following part of the section, we provide an overview of multiblock and multigrid methods with extensions. Specifically, in section 4.2 the general ideas and an illustrative layout for multiblock and multigrid techniques are introduced. In the following two sections 4.3 and 4.4, both the basic techniques are investigated in detail and problems for each method are pointed out, for which we suggest some improvements. While in the following section 4.5, Poiseuille flow is taken as a testing example for verification of the multiblock and multigrid methods. Moreover, we test the multigrid lattice Boltzmann method for initialization and demonstrate numerically that the time spent by this method for initialization is much less than the lattice Boltzmann itself without grid refinement. Finally, a more practical application for aneurysm with an idealized shape in 2D is carried out.

4.2 General ideas of multiblock and multigrid method

Let's take a simple geometry-three blocks sitting on two different grids-to illustrate the general ideas of multiblock and multigrid lattice Boltzmann method, as shown in Figure 4.2.1.

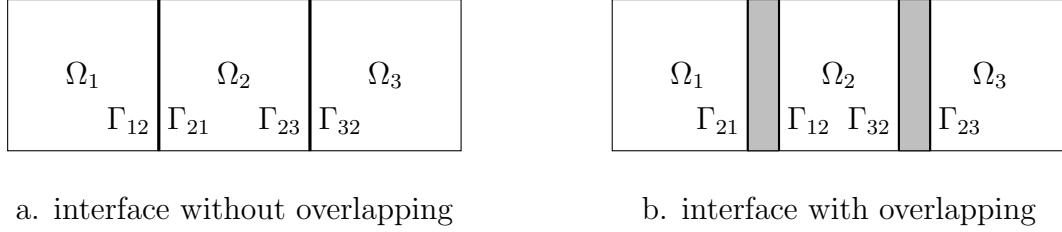


Figure 4.2.1: sketch for domain decomposition with two different layouts of interface.

The subdomains $\Omega_1, \Omega_2, \Omega_3$ are connected by the interfaces $\Gamma_{12} \subset \Omega_1, \Gamma_{21} \subset \Omega_2, \Gamma_{23} \subset \Omega_2, \Gamma_{32} \subset \Omega_3$. First of all, the interface could be an edge in 2 dimension as shown in Figure 4.2.1 a. $\Gamma_{12} = \Gamma_{21}, \Gamma_{23} = \Gamma_{32}$ or an overlapping region in Figure 4.2.1 b. $\Gamma_{12} \neq \Gamma_{21}, \Gamma_{23} \neq \Gamma_{32}$. The difference between the interface without overlapping and that with overlapping is, roughly speaking, that only the information on one layer of the grid or the information on several layers is shared by the neighboring subdomain during every iteration step. When it comes to lattice Boltzmann method, the overlapping scheme provides more convenient computation in the sense of parallelism, which is also the outstanding merit of lattice Boltzmann method(see Lätt et al. [82]).

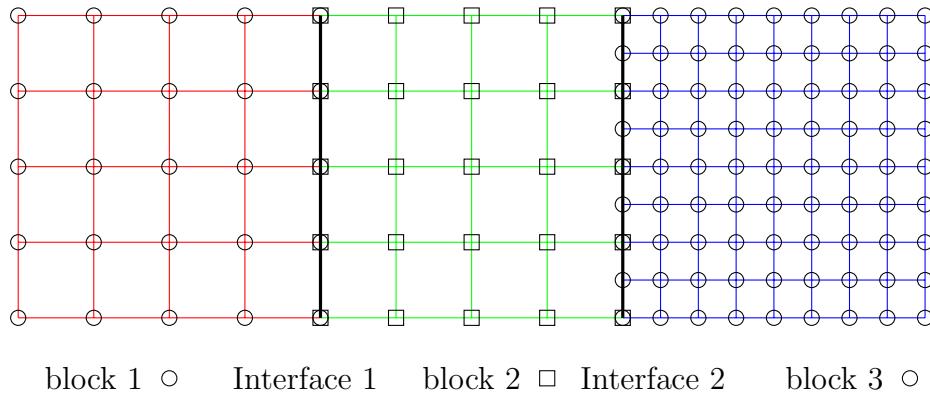


Figure 4.2.2: sketch for multiblock and multigrid without overlap on the interface; grids on interface 1 between block 1 and block 2 are kept the same while grids on interface 2 between block 2 and block 3 are refined by ratio 2

Figure 4.2.2 describes the multiblock and multigrid lattice without overlapping on

the interface, while Figure 4.2.3 displays the one with overlapping on the interface, corresponding to Figure (4.2.1).

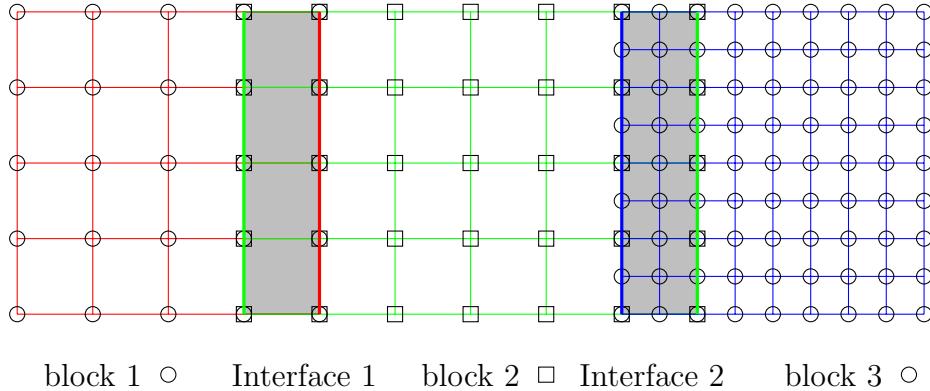


Figure 4.2.3: sketch for multiblock and multigrid with one layer of overlap on the interface; grids on interface 1 between block 1 and block 2 are kept the same while grids on interface 2 between block 2 and block 3 are refined by ratio 2

The ratio between fine grid and coarse grid can be any positive integral ratio n other than 2 as shown here. In practice, in order to pass information from very coarse grid to a much finer grid, multilevel hierarchical grids are applied as a compensation for high ratio.

4.3 Multiblock lattice Boltzmann method

Take block 1 and block 2 and their interface 1 in Figure 4.2.2 and 4.2.3 for illustration, we have two different schemes. The first is to set the multiblock without overlapping while the second is to connect the multiblock with overlapping on the interface. Either for the first scheme or the second one, the exchange of information on the interface could be carried out immediately after collision step or after streaming step. It does not make much difference because the lattice Boltzmann-BGK equations for the adjacent two blocks are the same with the same relaxation time. Let's take for example the second one - exchange of information after streaming step to examine multiblock lattice Boltzmann method without and with overlapping on the interface.

4.3.1 Multiblock without overlapping

All the steps (discretization, imposing boundary conditions, collision) are kept to be the same for both of the two blocks as those in one single block except for the steps of initialization and streaming:

Initialization step Set the time step δt and spatial step δx to be the same in both block 1 and block 2. Initialize density ρ and velocity \mathbf{u} as well as all the distribution function f_i for $i = 0, \dots, q$ as the equilibrium distribution function f_i^{eq} for $i = 0, \dots, q$ given by (2.4.52) in block 1 and block 2 respectively, and set them in block 2 to be the same as those in block 1 on the interface.

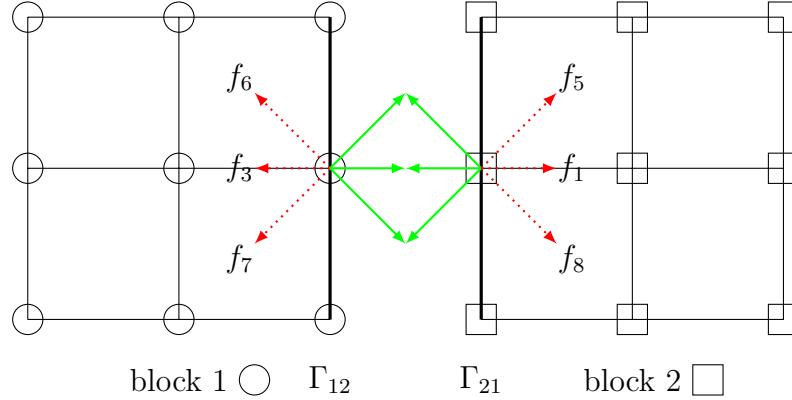


Figure 4.3.1: sketch for multiblock without overlapping on the interface

Streaming step After streaming step for each block, set the unknown distribution functions pointing from the node on the interface to the inner nodes of block 1 by the known distribution functions pointing in the same directions from this same node to the nodes outside of block 2, and do the same thing for the unknown distributions functions for block 2. Specifically, let $f_{3,6,7}|_{block1} = f_{3,6,7}|_{block2}$ on Γ_{12} and $f_{1,5,8}|_{block2} = f_{1,5,8}|_{block1}$ on Γ_{21} as shown in Figure 4.3.1. For the remaining distribution function, namely $f_{0,2,4}$, we don't touch them because they are always the same at the same node on the interface since the streaming and collision step in both of the two blocks keep $f_{0,2,4}$ to be the same from the initial step. For the node on the intersection of boundary and interface, there are three unknown distribution functions coming from outside the boundary. We can either apply the halfway bounceback rule for all of them or use regularized method or finite difference method to reconstruct the distribution function or even simply take the equilibrium distribution function as the distribution function after streaming step.

Remark 4.3.1 The scheme specified by the above two steps for multiblocks yields exactly the same simulation as by one single block with uniform grid everywhere in the computational domain. However, it requires specific directions (in Figure 4.3.1 the directions are 1, 5, 8 for block 1 and 3, 6, 7 for block 2) for information communication on the interface according to specific geometric outlay of blocks. A more practical scheme for multiblock method come from the way to exchange information on the interface,

which is swapping the distribution function in each direction for every node on the interface.

4.3.2 Multiblock with overlapping

All the steps are kept to be the same for both of the two blocks as those in one single block except the following step:

Initialization step Set the time step δt and spatial step δx to be the same in both block 1 and block 2. Initialize density ρ and velocity \mathbf{u} as well as all the distribution function f_i as the equilibrium distribution function f_i^{eq} given by (2.4.52) in block 1 and block 2 respectively, and replace all the distribution functions on the first layer of block 1 within the interface by the distribution functions on the second layer within the interface in block 2, and do the same thing for block 2 on the first layer, as shown in Figure 4.3.2. Then go to collision step and streaming step.

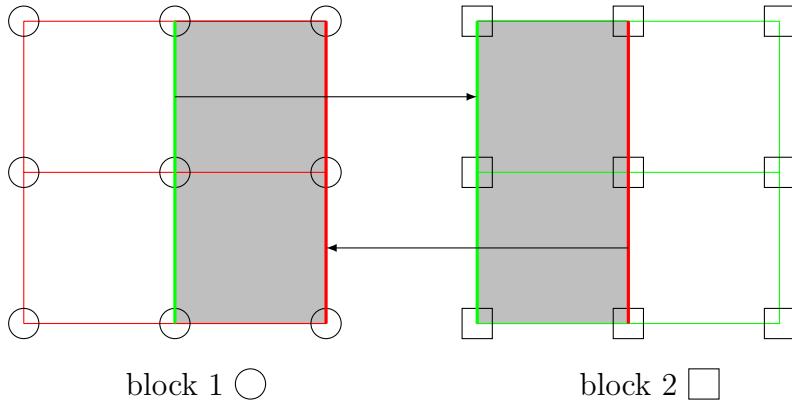


Figure 4.3.2: sketch for multiblock with overlapping on the interface

Streaming step After streaming step, replace all the distribution functions on the first layer of block 1 within the interface by the distribution functions on the second layer within the interface in block 2, and do the same thing for block 2 on the first layer, as shown in Figure 4.3.2. After this reset of the distribution functions on the interface, we continue with collision step in the next iteration process.

4.4 Multigrid lattice Boltzmann method

In this section, three different kinds of grid refinement techniques, namely the Filippova and Hänel, Dupuis and Chopard as well as Lin and Lai grid refinement, are investigated, compared and extended in detail.

4.4.1 Grid refinement - Filippova and Hänel

The first grid refinement technique we consider has been developed by Filippova and Hänel [53]. Two different scale of grids sitting on two adjacent blocks sharing an interface as an edge in dimension 2 or a face in dimension 3, as shown in Figure 4.4.1. The grid is refined by an integral ratio n ($n = 2$ in our illustration Figure 4.4.1), so that $\delta_x^c = n\delta_x^f$ and we also take $\delta_t^c = n\delta_t^f$ to have the same lattice velocity, where $\delta_x^c, \delta_t^c, \delta_x^f, \delta_t^f$ denote spatial and time step in coarse grid and fine grid respectively.

Remark 4.4.1 According to [86] for reaction-diffusion process, there are other options such as $\delta_t^c = C(n\delta_t^f)^2$ or $\delta_t = C$, C is a constant independent of δ_t^f .

In order to have a consistent physical flow, the Reynolds number in the two different grids should be kept the same. As a result, from (2.6.11) we have

$$\frac{(\delta_x^f)^2}{\delta_t^f} \nu_{lb}^f = \frac{1}{Re} = \frac{(\delta_x^c)^2}{\delta_t^c} \nu_{lb}^c, \quad (4.4.1)$$

together with the relation between relaxation time and viscosity in the lattice system $1/\omega = \tau = 1/2 + \nu_{lb}/c_s^2 = 1/2 + 3\nu_{lb}$, where ω is the relaxation frequency, we have

$$\omega_f = \frac{2}{1 + n(2/\omega_c - 1)}. \quad (4.4.2)$$

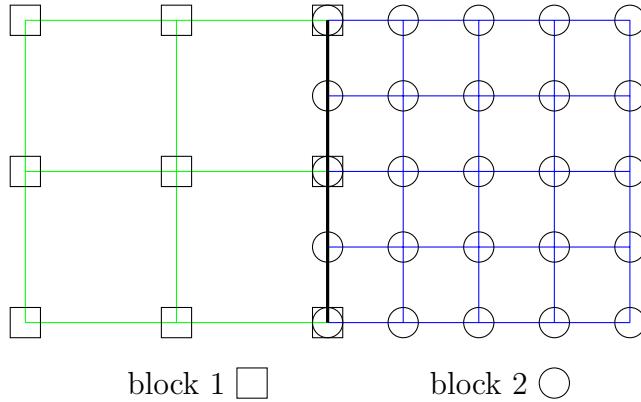


Figure 4.4.1: sketch for grid refinement without overlapping on the interface

Recall the collision step (2.4.68), and write it in another way for both coarse and fine grid as

$$f_i^{out,c} = f_i^{eq,c} + (f_i^c - f_i^{eq,c})(1 - \omega_c), \quad (4.4.3)$$

$$f_i^{out,f} = f_i^{eq,f} + (f_i^f - f_i^{eq,f})(1 - \omega_f). \quad (4.4.4)$$

Notice that equilibrium distribution functions $f_i^{eq,c}(\rho, \mathbf{u}) = f_i^{eq,f}(\rho, \mathbf{u})$ because they only depends on the density and velocity which are the same on the interface in both grids, as can be seen from the formula for equilibrium distribution function (2.4.52). From the lattice Boltzmann - BGK equation (2.4.16) without the term of force, the collision operator is given in the above discrete form as

$$(f_i - f_i^{eq}) = -\frac{1}{\omega}(\partial_t + \boldsymbol{\xi} \cdot \partial_{\mathbf{x}})f_i, \quad (4.4.5)$$

we obtain approximating the full distribution function by the equilibrium part in the derivative that

$$(f_i^c - f_i^{eq,c}) = -\frac{1}{\omega_c}(\partial_t^c + \boldsymbol{\xi} \cdot \partial_{\mathbf{x}}^c)f_i^c \approx -\frac{1}{\omega_c}(\partial_t^c + \boldsymbol{\xi} \cdot \partial_{\mathbf{x}}^c)f_i^{eq,c}. \quad (4.4.6)$$

Taking $f_i^{eq,c} = f_i^{eq,f}$, $\delta_x^c = n\delta_x^f$ and $\delta_t^c = n\delta_t^f$ into consideration, we obtain for (4.4.6) that

$$(f_i^c - f_i^{eq,c}) \approx -\frac{1}{\omega_c}(\partial_t^c + \boldsymbol{\xi} \cdot \partial_{\mathbf{x}}^c)f_i^{eq,c} = -\frac{n}{\omega_c}(\partial_t^f + \boldsymbol{\xi} \cdot \partial_{\mathbf{x}}^f)f_i^{eq,f}. \quad (4.4.7)$$

Because

$$(f_i^f - f_i^{eq,f}) = -\frac{1}{\omega_f}(\partial_t^f + \boldsymbol{\xi} \cdot \partial_{\mathbf{x}}^f)f_i^f \approx -\frac{1}{\omega_f}(\partial_t^f + \boldsymbol{\xi} \cdot \partial_{\mathbf{x}}^f)f_i^{eq,f}, \quad (4.4.8)$$

by comparing (4.4.7) and (4.4.8), we get the post collision distribution function on the interface in the coarse grid from (4.4.3) as

$$\begin{aligned} f_i^{out,c} &= f_i^{eq,c} + (f_i^c - f_i^{eq,c})(1 - \omega_c) \\ &= f_i^{eq,f} + \frac{n\omega_f(1 - \omega_c)}{\omega_c}(f_i^f - f_i^{eq,f}) \\ &= f_i^{eq,f} + \frac{n\omega_f(1 - \omega_c)}{\omega_c(1 - \omega_f)}(f_i^{out,f} - f_i^{eq,f}). \end{aligned} \quad (4.4.9)$$

Following the same analysis as for the post collision distribution function in the coarse grid, or immediately from (4.4.9), we have for the post collision distribution function in the fine grid

$$f_i^{out,f} = \tilde{f}_i^{eq,c} + \frac{\omega_c(1 - \omega_f)}{n\omega_f(1 - \omega_c)}(\tilde{f}_i^{out,c} - \tilde{f}_i^{eq,c}), \quad (4.4.10)$$

where \tilde{f} denote the spatial and temporal interpolation function from coarse grid to fine grid. There are several options for the interpolation, such as the first order or the second order interpolation via Lagrangian polynomials or spline interpolation.

Remark 4.4.2 While the grid refinement scheme proposed by Filippova and Hänel has no overlapping on the interface, we push this scheme forward for the interface

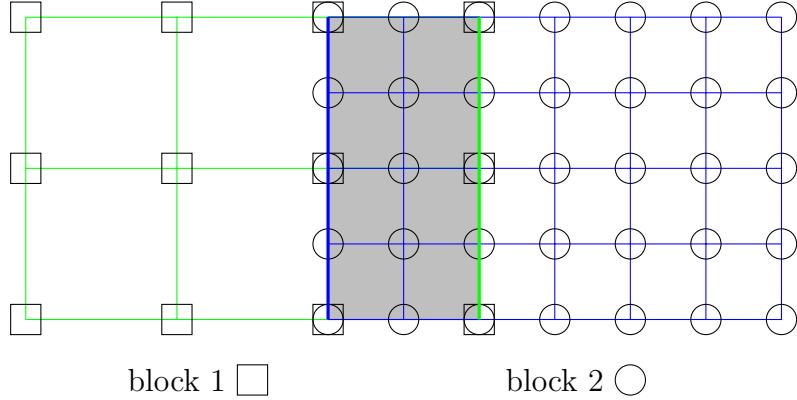


Figure 4.4.2: sketch for grid refinement with overlapping on the interface

with overlapping as shown in Figure 4.4.2. Basically, (4.4.9) is applied on the right boundary of the interface in Figure 4.4.2 to get the post collision distribution function in the coarse grid, while (4.4.10) is borrowed for the post collision distribution function in the fine grid on the left boundary of the interface. After this step, the streaming step is followed.

4.4.2 Grid refinement - Dupuis and Chopard

Dupuis and Chopard [80] pointed out that when the relaxation frequency was close to 1 in (4.4.10) or (4.4.9), the scheme developed by Filippova and Hänel displayed a bad behavior - instability for multigrid lattice Boltzmann method. This defect has been cured by a relatively simple scheme proposed by Dupuis and Chopard.

First of all, the exchange of information takes place after streaming step instead of collision step in the non overlapping interface as shown in Figure 4.4.1. The post streaming distribution function on the interface can be decomposed into equilibrium part and non-equilibrium part as $f_i^{in} = f_i^{eq} + f_i^{neq}$. As the equilibrium distribution function are the same for both the fine grid and the coarse grid since it depends merely on the same density and velocity, there is only the non-equilibrium part left to be taken care of.

Since the non-equilibrium distribution function can be approximated by (2.5.35) without the external force and the correction term, we have by $\delta_x^c = n\delta_x^f$ that

$$\begin{aligned}
f_i^{neq,c} &= -\frac{w_i \tau_c}{c_s^2} \left(Q_i : \rho \nabla_1^c \mathbf{u} - \boldsymbol{\xi}_i \nabla_1^c : \rho \mathbf{u} \mathbf{u} + \frac{1}{2c_s^2} (\boldsymbol{\xi}_i \cdot \nabla_1^c) (Q_i : \rho \mathbf{u} \mathbf{u}) \right) \\
&= -\frac{w_i \tau_c n}{c_s^2} \left(Q_i : \rho \nabla_1^f \mathbf{u} - \boldsymbol{\xi}_i \nabla_1^f : \rho \mathbf{u} \mathbf{u} + \frac{1}{2c_s^2} (\boldsymbol{\xi}_i \cdot \nabla_1^f) (Q_i : \rho \mathbf{u} \mathbf{u}) \right) \quad (4.4.11) \\
&= \frac{\tau_c n}{\tau_f} f_i^{neq,f}.
\end{aligned}$$

Therefore, we obtain the grid refinement scheme as

$$f_i^{in,c} = f_i^{eq,f} + \frac{\tau_c n}{\tau_f} (f_i^{in,f} - f_i^{eq,f}). \quad (4.4.12)$$

$$f_i^{in,f} = \tilde{f}_i^{eq,c} + \frac{\tau_f}{\tau_c n} (\tilde{f}_i^{in,c} - \tilde{f}_i^{eq,c}), \quad (4.4.13)$$

where \tilde{f} denote the interpolated distribution function from coarse grid to fine grid.

Remark 4.4.3 It's worth to mention that in the original paper [80] after streaming step, the distribution functions are not known in all directions. So (4.4.12) cannot be directly applied to all the directions the same as that for the post collision distribution function in the grid refinement scheme by Filippova and Hänel. To deal with this problem, one method is to exchange distribution functions only in the known directions, the same idea as multiblock without overlapping interface. However, this method can not conserve density and velocity on the interface for both of the grids due to interpolation for fine grid from coarse grid and of partial treatment for all the distribution functions by grid refinement. Different density and velocity lead to different equilibrium distributions so that the claim of the same equilibrium distribution function on the interface does not hold any more, though the difference of the equilibrium distribution function is small.

A better idea is to use the overlapping interface as shown in Figure 4.4.2. After streaming step, apply (4.4.12) on the right boundary of the interface to obtain the post streaming distribution function in the coarse grid by that in the fine grid in all the directions. And then apply (4.4.13) on the left boundary of the interface to calculate the post streaming distribution function in the fine grid by proper spatial and temporal interpolation.

4.4.3 Grid refinement - Lin and Lai

This scheme follows the idea of refining grid when needed [79], which means that we build firstly a background grid - the most coarse grid - everywhere, and refine it to one

finer grid or several nesting finer grids at the location where a more accurate simulation is needed. Figure 4.4.3 displays a two level composite grids in a certain location, where the ratio of spatial step between the coarse grid and the fine grid is 2 (it can be any positive integer n).

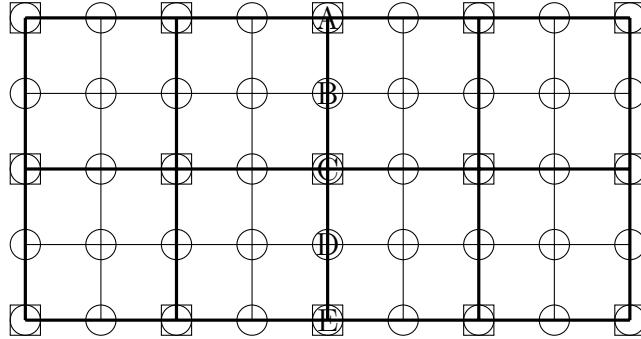


Figure 4.4.3: sketch for composite grid refinement

The process for the composite grid is run as shown in Figure 4.4.4

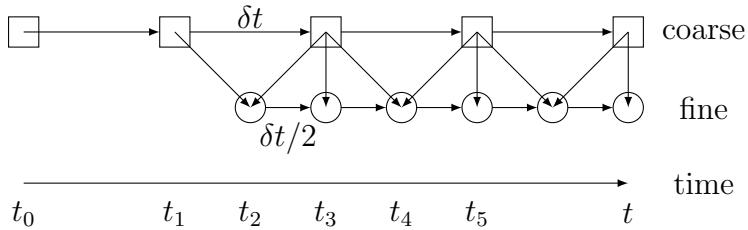


Figure 4.4.4: sketch for the process of composite grid refinement

Unlike the grid refinement process as we have seen before, this composite grid refinement only involves passing information from coarse grid to fine grid, as shown in Figure 4.4.4. The process for composite grid refinement is given in the following steps:

Step 1 Provide initial conditions for the coarse grid at time t_0 and run lattice Boltzmann algorithm until a certain time t_1 , which depends on a demanding state of the flow in the coarse grid.

Step 2 Run lattice Boltzmann algorithm a step further in the coarse grid to time t_3 and then take the temporal interpolation of the distribution functions in the coarse grid for the distribution functions in the fine grid at the same node. In the fine grid, take the spatial interpolation of the distribution functions interpolated before to obtain the distribution functions on the nodes sitting only in the fine grid.

Step 3 Impose boundary conditions for the fine grid: the distribution function f_i after streaming, instead of density and velocity, are taken from coarse grid at time t_3

for the fine grid at the same nodes on the boundary of the two grids. Interpolate from the imposed boundary conditions to get the boundary conditions at the nodes sitting only in the fine grid. Run lattice Boltzmann algorithm a step further in the fine grid to time t_3 .

Step 4 Impose boundary conditions for the coarse grid and run lattice Boltzmann algorithm a step further in the coarse grid to time t_5 .

Step 5 Impose boundary conditions for the fine grid: take the temporal interpolation of the distribution functions in the coarse grid for the distribution functions in the fine grid at the same node on the boundary instead of the whole domain. In the fine grid, take the spatial interpolation of the distribution functions interpolated before to obtain the distribution functions on the nodes sitting only in the fine grid on the boundary. Run lattice Boltzmann algorithm a step further in the fine grid to time t_4 with the boundary conditions imposed in the form of distribution functions obtained from Step 4.

Step 6 Impose boundary conditions for the fine grid as we did in Step 3. Run lattice Boltzmann algorithm a step further in the fine grid to time t_5 with the boundary conditions obtained from Step 5.

Step 7 Continue from time t_5 and repeat Step 4 to Step 6 until it meets stopping condition.

Remark 4.4.4 *Lin and Lai boundary dynamics are especially useful for initialization, for which we run Step 7 until arriving at a demanding state in the fine grid. If the state meets our requirement of accuracy before a given stopping time, then stop. Otherwise refine the fine grid to be finer and repeat Step 2 to Step 7 to run the lattice Boltzmann algorithm in a multilevel nesting grid.*

If the boundary node is on the wall and a bounce back rule is applied for the boundary condition in the coarse grid, take for example A in Figure 4.4.3, then there is distribution function unknown at node A after streaming so that we can't use interpolation for node B in the fine grid as we do for node D by interpolation with known distribution functions at node C, E . Two approaches could be applied to deal with this problem. The first introduced by Lin and Lai replaces the distribution function required at node B by equilibrium distribution function constructed by the interpolated density and velocity from node A and C . The other approach is to extrapolate the distribution function at node B from the distribution function at node C and E in the coarse grid.

Remark 4.4.5 *As pointed out by Dupuis and Chopard, this composite grid refinement is not as accurate as the grid refinement proposed either by Filippova and Hanel, or by*

Dupuis and Chopard because it does not take the non-equilibrium part of the distribution function into consideration. Therefore, we can borrow the idea from Filippova and Hanel or from Dupuis and Chopard for the composite grid refinement to improve the accuracy.

Remark 4.4.6 *It's worth to mention that the composite grid refinement lattice Boltzmann method could be well applied for initialization of pressure (or density), which provides an alternative method for solving Poisson problem to obtain the initial pressure. The example of Poiseuille flow is taken to show the better performance of the multigrid lattice Boltzmann method than the uniform grid lattice Boltzmann method in the benchmark later on.*

4.5 Numerical verification and application

In this section, different kinds of multiblock and multigrid lattice Boltzmann methods are tested and compared. Specifically, using the Poiseuille flow as a benchmark, we test in the first benchmark for the verification of multiblock with or without overlapping, multigrid by different grid refinements. The second benchmark, initialization by multigrid lattice Boltzmann method, verifies that multigrid technique provides a very efficient way to initialize the problem. As a further application, the blood flow in an aneurysm, is simulated.

4.5.1 Verification of multiblock and multigrid method

Poiseuille flow is the first example to illustrate multiblock and multigrid methods. Three blocks sitting on two grids according to Figure 4.2.2 without overlapping and Figure 4.2.3 with overlapping are taken as the simulating domain respectively. All the blocks are rectangles with width 2 and height 1 in dimensionless system, while the left two blocks are discretized as 65×33 cells and the right one is refined to be a mesh of 129×65 . The time step for the coarse grid is given as $\delta t_c = \delta x_c^2$, while for fine grid it is refined accordingly. The physical parameters are taken the same as those in the first benchmark - Poiseuille flow - for straight boundary condition in section 3.3.

Between block 1 and block 2, we apply the multiblock method introduced in section 4.3 on the interface. While for block 2 and block 3, the multigrid methods of Filippova and Hanel as well as Dupuis and Chopard with formulas 4.4.12 and 4.4.13 are applied for illustration.

Starting from a null initial condition for velocity and constant value for pressure, we reach the steady state for Poiseuille flow after 100 periods by both of the two methods, as shown in Figure 4.5.1 and Figure 4.5.2 for pressure at $y = 0.5$ along x axis.

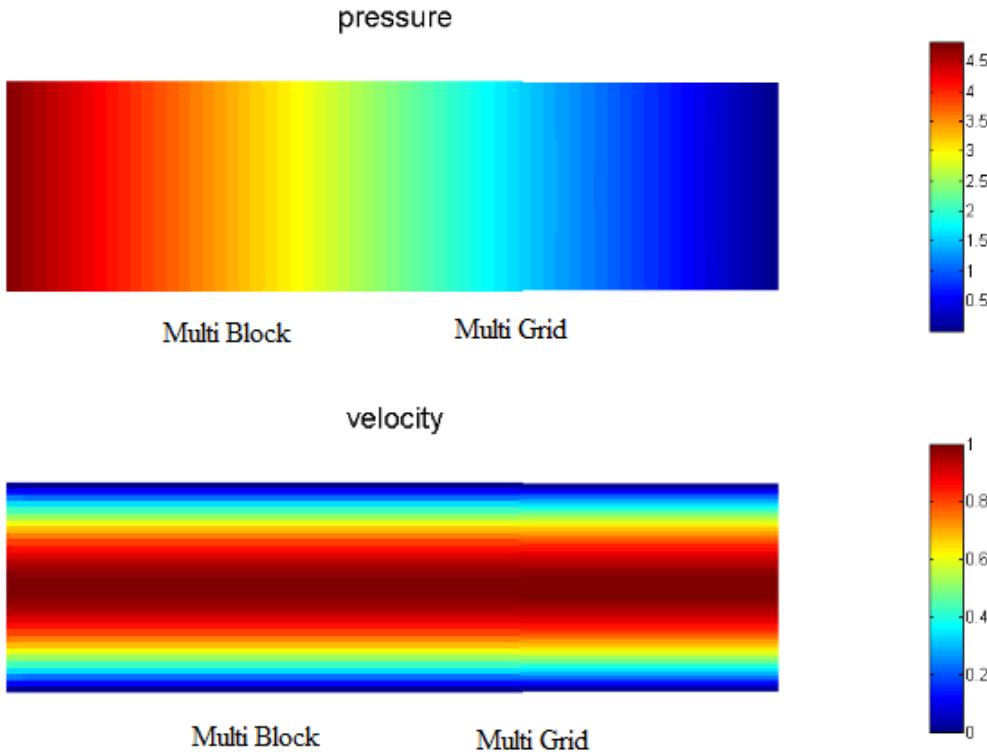


Figure 4.5.1: sketch for steady numerical results of pressure and velocity for Poiseuille flow in three blocks sitting on two grids, by the scheme without overlapping

Table 4.5.1: Error comparison between different multigrid method

Error in L^2 norm	Non overlapping	Overlapping
Filippova and Hänel	0.0044	0.0042
Dupuis and Chopard	0.0028	0.0026

Remark 4.5.1 *By steady state, it means that when the overall error in L^2 norm between numerical solution and analytical solution for the three blocks becomes a constant depending on multiblock and multigrid method as well as the discretization and specific boundary conditions.*

The Errors between analytical solution and numerical solution by Filippova and Hänel's multigrid method as well as by Dupuis and Chopard's multigrid method are shown in Table 4.5.1. Meanwhile, steady state is reached a bit earlier for the scheme with overlapping than that without for both of the two methods.

Remark 4.5.2 *From this benchmark we can see firstly that both the scheme without overlapping interface and the one with overlapping interface work pretty well for multi-*

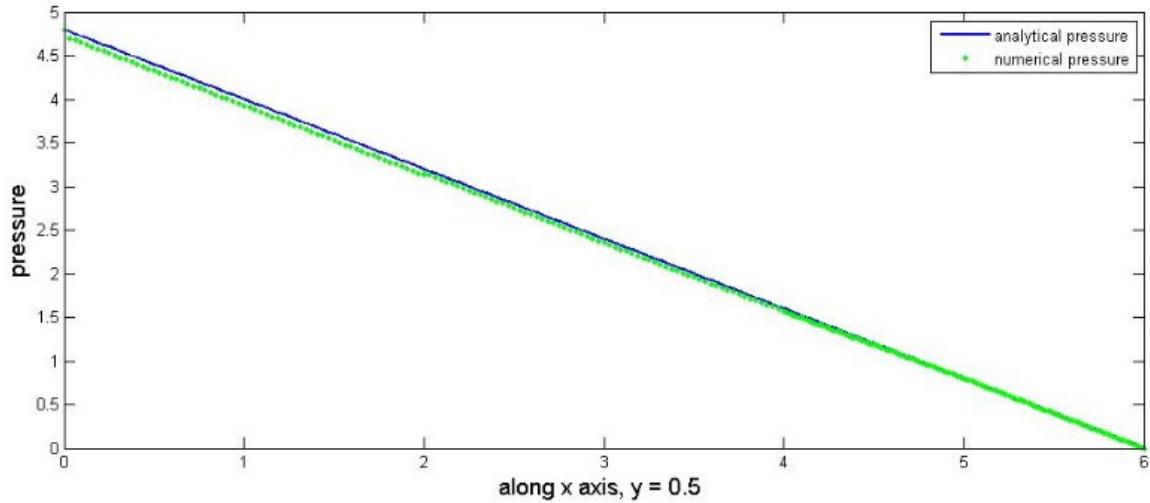


Figure 4.5.2: comparison between analytical pressure and numerical pressure by multi-block with overlapping and Dupuis and Chopard's multigrid with overlapping along x axis in the middle of y direction, e.g. $y = 0.5$

block and multigrid method proposed by Filippova and Hänel as well as by Dupuis and Chopard. Comparatively, the scheme with overlapping interface is slightly better than that without overlapping interface, and Dupuis and Chopard's method is more accurate than Filippova and Hänel's.

4.5.2 Initialization by multigrid lattice Boltzmann method

As mentioned before, lattice Boltzmann method is a weak-compressible Navier-Stokes solver. Therefore, providing precise initial conditions not only for velocity but also for pressure is as important as good boundary conditions, especially for unsteady flows. In order to have a good initial pressure, we are offered one choice to solve Poisson problem by conventional method such as finite difference method or finite element method. An alternative option is to apply lattice Boltzmann method to obtain the precise initial pressure by keeping the initial velocity unchanged. However, the problem for the latter approach is that it will take a long time because of its compressibility effect. Two improvements have been made to deal with this problem. One is the so called multiple relaxation time (MRT) lattice Boltzmann method proposed by Mei et al. [62], where the number of relaxation time is extended to be nine instead of the single one in lattice Boltzmann - BGK model. The acceleration comes from the additional relaxation time. The other improvement is to accelerate the simulation by hierarchical or composite grids on several levels of space steps based on the lattice Boltzmann method. Let's take the multigrid method proposed by Lin and Lai with hierarchical grids on four levels

to initialize Poiseuille flow in a domain with width 2 and height 1 in dimensionless system. All the physical parameters for Poiseuille flow are given as before.

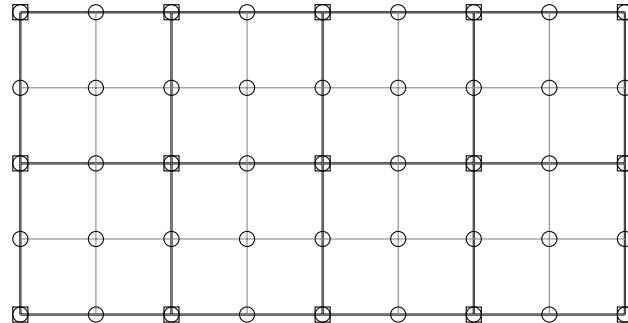


Figure 4.5.3: sketch for multigrid lattice Boltzmann method for initialization

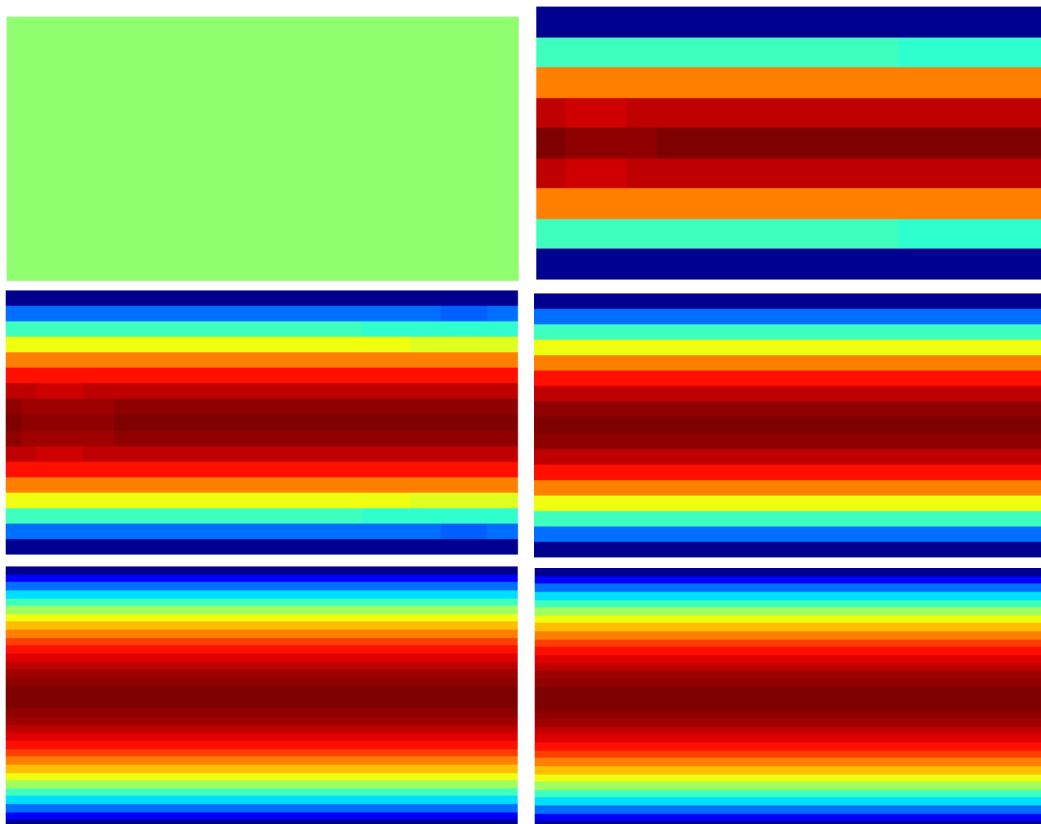


Figure 4.5.4: initialization by multigrid lattice Boltzmann method(MG-LBM) in hierarchical grids for Poiseuille flow with required error 10^{-6} , initial state (left) and final state (right) on each of the three levels

As shown in Figure 4.5.4, each time for grid refinement, half of the space step is taken as the new space step, as shown in Figure 4.5.3. We start with null velocity and

pressure everywhere in the domain and increase the space step from $1/8$ until after three times of refinement to $1/64$.

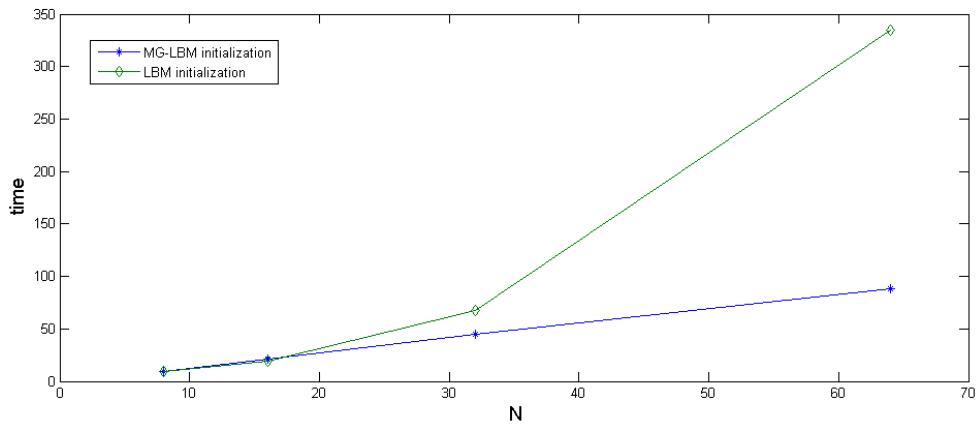


Figure 4.5.5: computational time for initialization by lattice Boltzmann method(LBM) and multigrid lattice Boltzmann method(MG-LBM) with required error 10^{-6}

From the comparison of computational time taken by lattice Boltzmann method and multigrid lattice Boltzmann method, we clearly see that the latter one achieves much better behavior and considerably reduces the computational time spent on initialization than the former one, especially when the grid is very fine.

4.5.3 Aneurysm with a simplified geometric shape

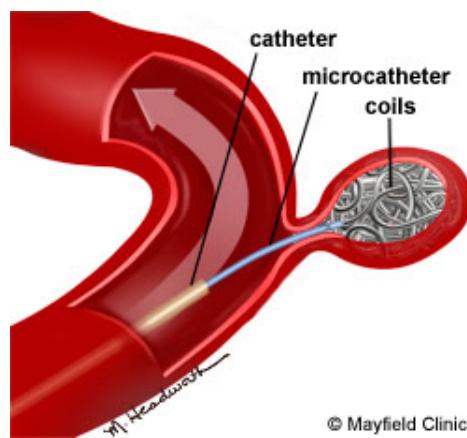


Figure 4.5.6: sketch for simplified medical images of aneurysm

For the last test case, as an illustrative example, we considered the blood flow in an idealized aneurysm. An aneurysm is a localized, blood-filled balloon-like bulge of

a blood vessel, see Figure 4.5.6. When the size of an aneurysm increases, there is a significant risk of rupture, resulting in severe hemorrhage, other complications or even death [83]. In order to illustrate the multiblock and multigrid lattice Boltzmann method on a complex geometry, in this section we have considered an idealized shape for aneurysm as shown in Figure 4.5.6.

Figure 4.5.7 displays a simplified sketch of an idealized aneurysm with two blocks of $0.2cm \times 0.1cm$, and one block of $0.2cm \times 0.2cm$ with curved boundary for the geometry of computational domain. There are multiple choices for the grid as well as interface, such as multiblock with the same or different grids, or interface with or without overlapping. For this specific simulation, we take the scheme developed by Dupuis and Chopard with overlapping on the interface for our simulation.

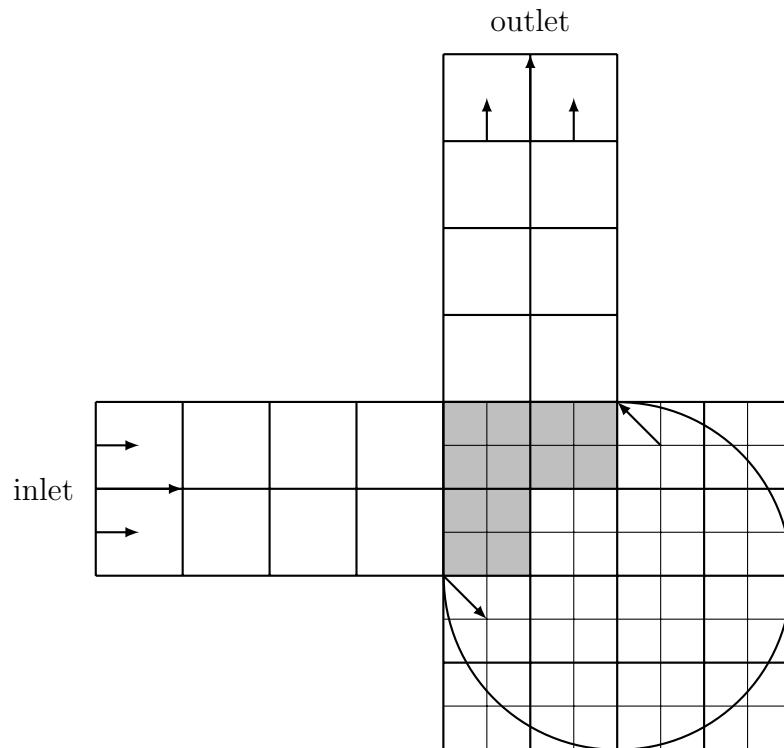


Figure 4.5.7: sketch for multiblock and multigrid for aneurysm in Figure 4.5.6, the gray region stands for the overlapping interface

Blood viscosity is set as $0.035g/(cm \cdot s)$, within the range of that for blood flow, whose viscosity locates in $[3, 4] \times 10^{-2}g/(cm \cdot s)$. The inlet boundary conditions is prescribed by a given periodic velocity within one period of $1s$ as shown in Figure 3.5.7 while a homogeneous Neumann boundary conditions is imposed on the outlet boundary.

We start from constant pressure and zero velocity everywhere and let it run up to 10

periods (10 seconds) until the simulation displays a relative ideal periodic flow. Figure 4.5.8 and 4.5.9 shows the simulation result for both velocity and pressure within one period at the first and eleventh period. From these figures we can tell that the coupling of multiblock and multigrid scheme works pretty well for lattice Boltzmann method, in the sense that information is exchanged smoothly and efficiently. This verification could be served as the solid ground for multiblock and multigrid lattice Boltzmann method to be applied for large computation with multi processors.

Remark 4.5.3 *From Figure (4.5.8) and (4.5.9), we can still see the compressibility effect (the velocity on the inlet boundary is should be more or less the same as on the outlet boundary). In order to reduce this effect, finer grid is required.*

4.6 Conclusion

For the sake of multiscale property of physical problem as well as potential large scale parallel computing by lattice Boltzmann method, both multiblock and multigrid techniques have been investigated and successfully applied in the frame of lattice Boltzmann - BGK model. The crucial step for multiblock and multigrid method is how to transfer information from one block to another or from one grid to another accurately and stably.

As for multiblock method, both interface with overlapping and without overlapping were considered and tested. According to our benchmark, multiblock with overlapping display a better performance in accuracy and efficiency.

As for multigrid method, three kinds of grid refinement schemes were examined and compared. All of them worked pretty well for information transferring from coarse grid to fine grid and vice versa. Comparatively, the grid refinement by Dupuis and Chopard displayed the best behavior with smallest error for information exchanging. The grid refinement by Lin and Lai has been plausibly used for initialization of lattice Boltzmann method.

As a more practical application, both straight boundary and curved boundary, both multiblock and multigrid were applied for the blood flow in an idealized aneurysm.

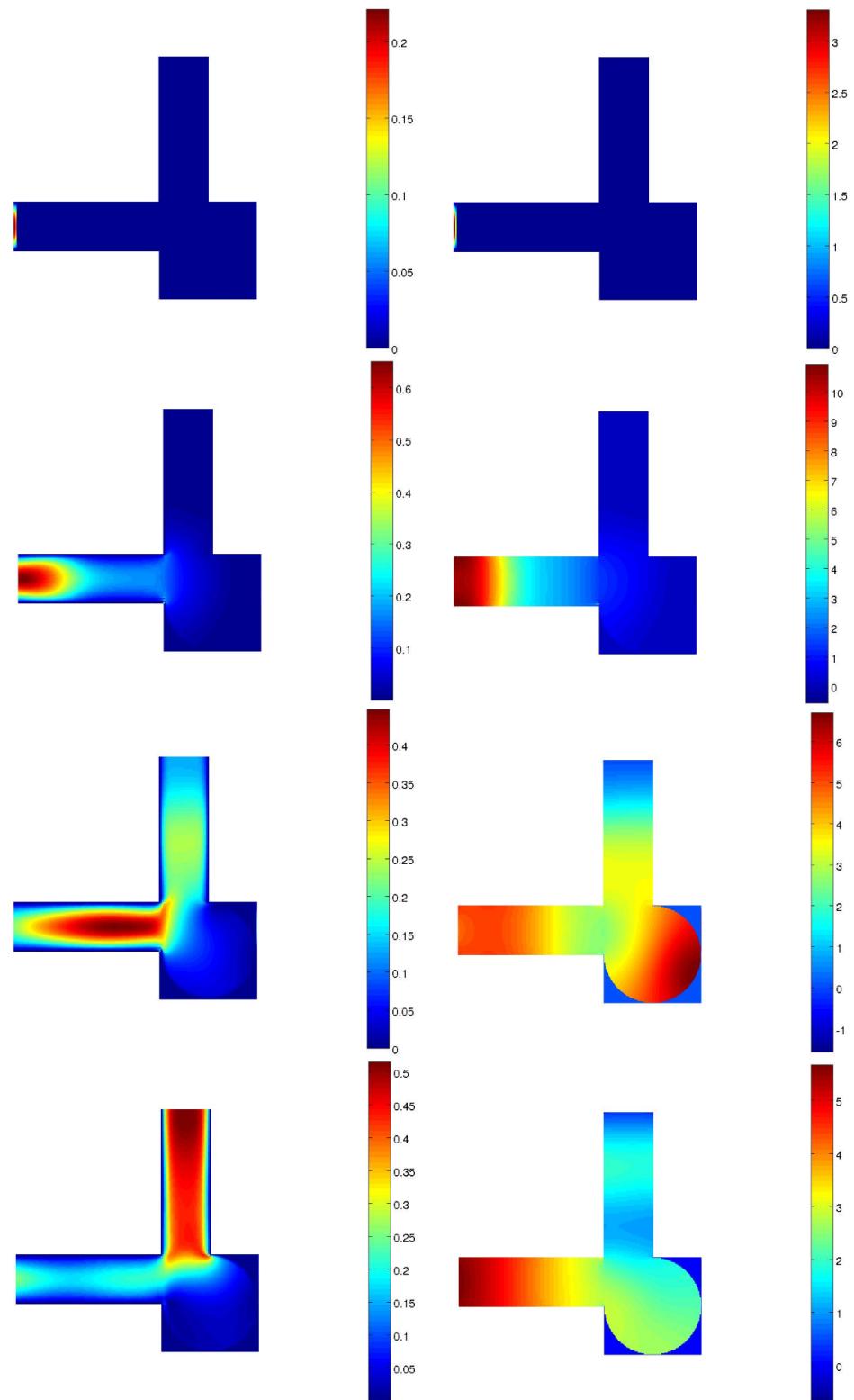


Figure 4.5.8: velocity(Left) and pressure(Right) of blood flow for aneurysm in a period at time $t = 0s, 0.25s, 0.5s, 0.75s$

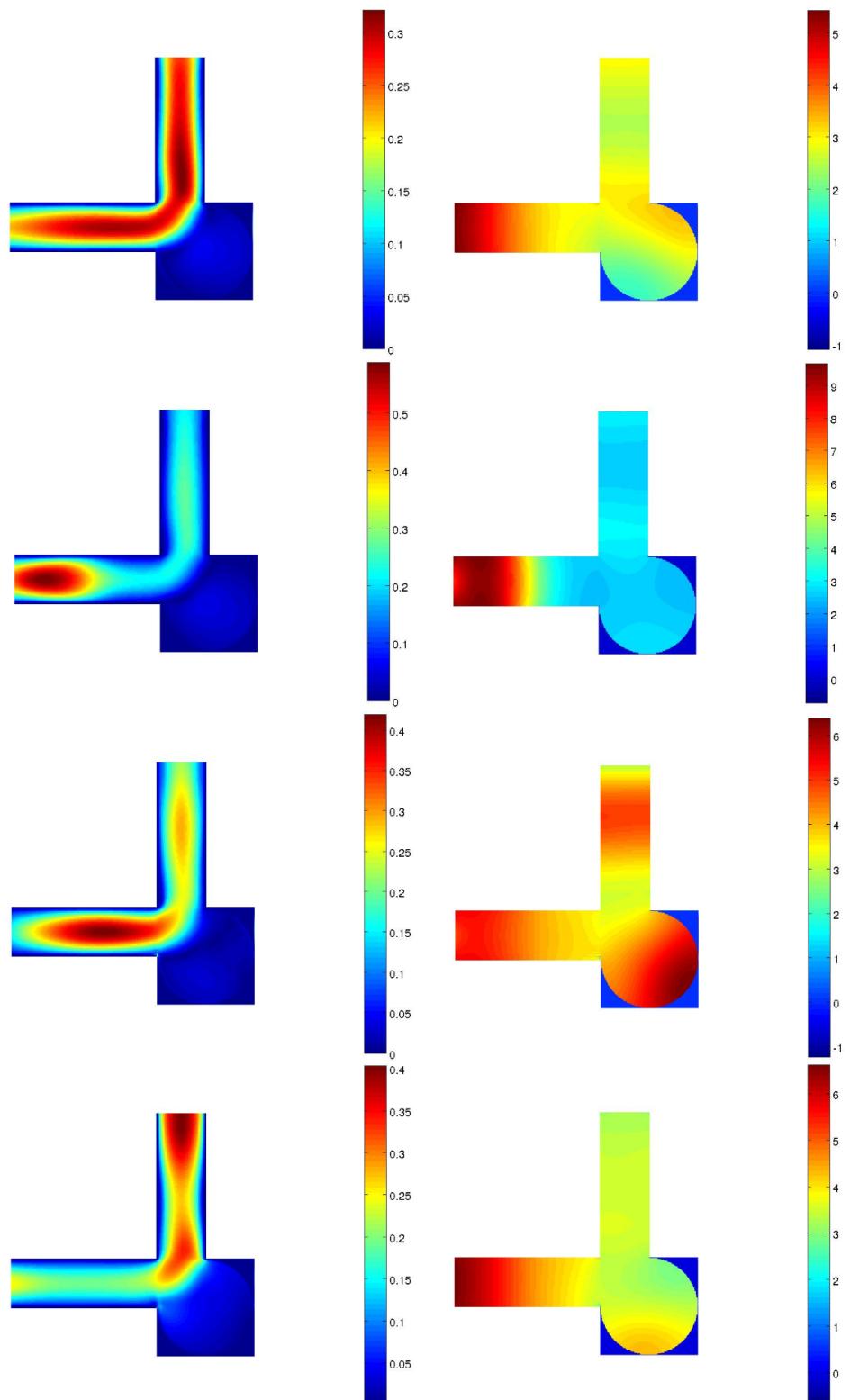


Figure 4.5.9: velocity(Left) and pressure(Right) of blood flow for aneurysm in a period at time $t = 10s, 10.25s, 10.5s, 10.75s$

Chapter 5

Conclusions and Perspectives

Conclusions

The lattice Boltzmann method has flooded over the land of computational fluid dynamics since the avalanche was triggered by the discovery of the kinetic representation of Navier-Stokes equations. In this thesis we restricted ourselves to the appreciation of three of the most shiny waves in the flood - theoretical background of lattice Boltzmann method, how to impose boundary conditions for both straight and curved boundaries as well as multiblock and multigrid schemes for lattice Boltzmann method.

For the theoretical background of the lattice Boltzmann method in Chapter 2, we first introduced the Boltzmann equation in the frame of statistical mechanics and one of its specific linearization and simplification: the Boltzmann - BGK equation. Starting from the Boltzmann - BGK equation we took the zeroth, first and second order moments of velocity with respect to the particle distribution function and arrived at the continuity equation, momentum equation as well as the energy equation. However, the system of continuum equations are not closed because of the presence of momentum stress tensor and energy flux. In order to search for their explicit expression with density, velocity and corresponding derivatives, we examined and compared two different approaches - the Grad's representation approach and Chapman-Enskog expansion approach. Finally in this section we illustrated different lattice structures in 1D, 2D and 3D for different lattice Boltzmann models with the favor of Gauss-Hermite quadrature formulas.

As for the boundary conditions to be imposed in lattice Boltzmann model, we took Dirichlet boundary conditions for example and explained the big difference between conventional numerical methods and the lattice Boltzmann method, in Chapter 3. Concerning the distinctive geometric property of straight boundary and curved boundary in uniform structured grid, we considered various approaches for straight boundary and curved boundary respectively. In the kinetic level, we applied some schemes in the

family of “Distribution Modification” such as bounce back rule, mass and momentum conservation law and so on to explore the full way and half way bounce back dynamics as well as the Zou-He boundary dynamics. While in the family of “Distribution Reconstruction”, we investigated the regularized boundary dynamics and the finite difference dynamics. With the benchmarks of Poiseuille flow, we obtained the second order numerical accuracy for all the boundary dynamics except full way bounce back boundary dynamics. We also compared the stability and efficiency of the different boundary dynamics. The same ideas of distribution modification and distribution reconstruction were also applied for curved boundary but with appropriate interpolation and extrapolation.

In order to pave the way for more practical application of lattice Boltzmann method as well as large scale parallel computation by the lattice Boltzmann method, we examined both multiblock and multigrid schemes in Chapter 4 corresponding to domain decomposition method for conventional numerical approach. Specifically, for multiblock lattice Boltzmann method, both the schemes of with overlapping and without overlapping on the interface were considered, and the conclusion was arrived that overlapping scheme is slightly more efficient and accurate. As for multigrid scheme, we explored the one developed by Filippova and Hänel, the one suggested by Dupuis and Chopard as well as the composite grid refinement scheme proposed by Lin and Lai. Accuracy and efficiency were compared for these different multigrid schemes. We also gave a glimpse to initialization by multigrid lattice Boltzmann method. Meanwhile, as an application, simulation for blood flow in an idealized aneurysm with both straight boundary and curved boundary, both multiblock and multigrid lattice Boltzmann method was conducted.

Perspectives

The lattice Boltzmann method has been nurtured with the various nutrition from statistical mechanics and computational fluid dynamics and developed to its adolescence with exciting theoretical results and hot practical applications up to the present. There are still many promising developments undergoing currently and in the near future.

Beyond the second order temporal and spatial accuracy as well as the Navier-Stokes equations obtained for lattice Boltzmann method in Chapter 2, higher order accuracy and continuum equations related to thermal effects for hydrodynamics is worth to be achieved. Hopefully, the obstacles of low accuracy and restriction to momentum equations will be cleared to pave the way for the lattice Boltzmann method to be applicable for more accurate thermodynamics.

Though the stability, accuracy as well as efficiency of differential approaches for

imposing Dirichlet boundary conditions have been investigated in detail in Chapter 3 and the corresponding verifications were also given numerically, the theoretical analysis, /especially for stability and accuracy for imposing different boundary conditions are still missing in the literatures. We believe that the theoretical analysis will be critical for future development of the lattice Boltzmann method in general, not only for imposing boundary conditions but also for initialization. Moreover, the lattice Boltzmann method is extremely constrained to fixed computational domain for fluid at the moment. Therefore, moving computational domain with coupling fluid and structure interactions are demanding.

The intrinsic parallelism of the lattice Boltzmann method has been developed and applied in its infancy presently, for which multiscale and multigrid scheme in Chapter 4 will be severed as the cornerstone. Large scale parallel computation on the platform with multi processors of CPUs and GPUs will push the lattice Boltzmann method as a new Star in the field of computational fluid dynamics. Meanwhile, the coupling between the lattice Boltzmann method and the conventional numerical methods such as finite difference, finite element, finite volume and spectral element method still remain an open research field.

Appendix A

Lattice Structure in 1D, 2D and 3D

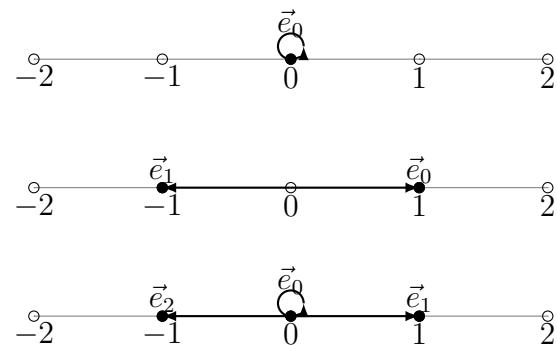


Figure A.0.1: Gauss-Hermite quadrature formulas $E_{1,1}^1, E_{1,3}^2, E_{1,5}^3$

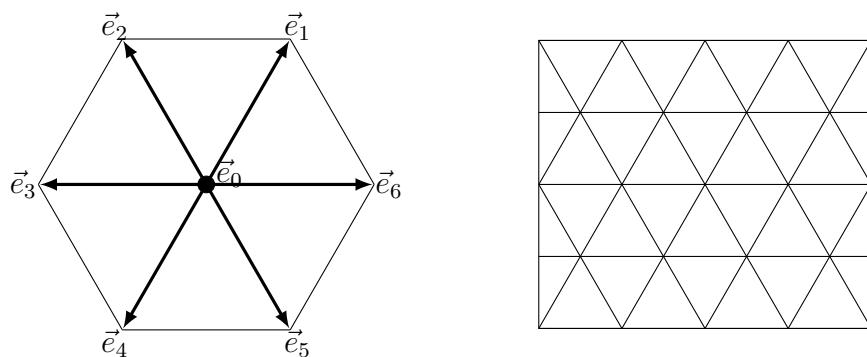
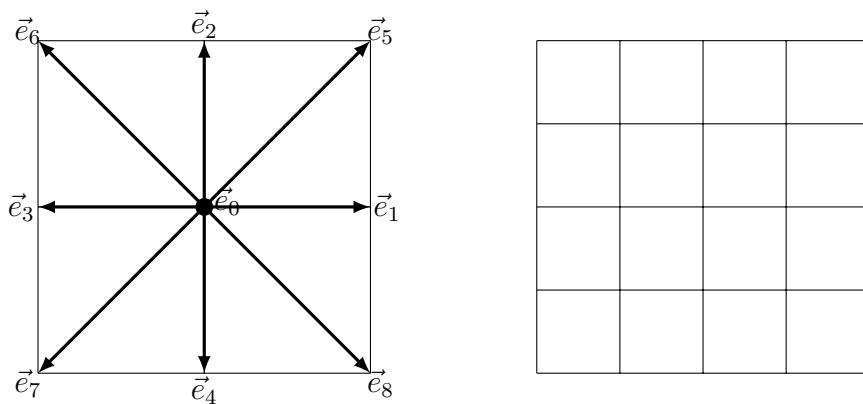
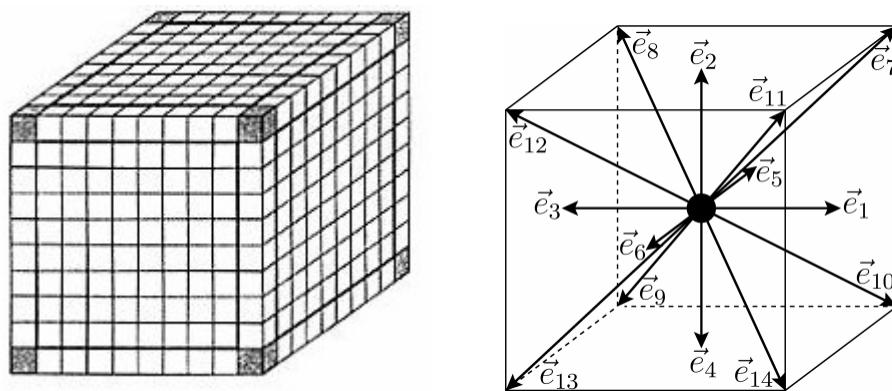
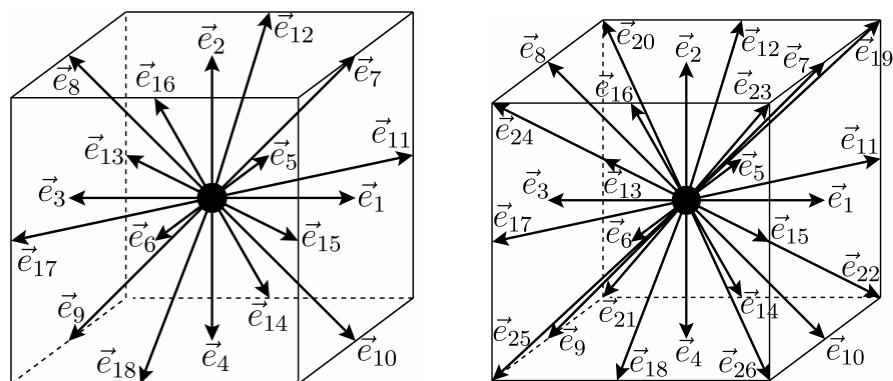


Figure A.0.2: Gauss-Hermite quadrature formulas $E_{2,5}^7$ Left: Structure Right: Mesh

Figure A.0.3: Gauss-Hermite quadrature formulas $E_{2,5}^9$, Left: Structure Right: MeshFigure A.0.4: Gauss-Hermite quadrature formulas $E_{3,5}^{15}$ Left: Structure Right: MeshFigure A.0.5: Gauss-Hermite quadrature formulas Left : $E_{3,5}^{19}$ and Right: $E_{3,5}^{27}$

Bibliography

- [1] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the Navier-Stokes equation. *Phys. Rev. Lett.*, 56(14):15051508, Apr 1986.
- [2] M. Griebel, T. Dornseifer, and T. Neunhoefer. Numerical simulation in uid dynamics: a practical introduction. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.
- [3] F. J. Higuera and J. Jimenez. Boltzmann approach to lattice gas simulations. *Europhys. Lett.*, 9(7):663668, 1989
- [4] J. Hardy, Y. Pomeau, O. de Pazzis. Time evolution of a twodimensional model system. I. Invariant states and time correlation functions. *J. Math. Phys.* 14, 1746, 1973
- [5] D. Chandler. Introduction to Modern Statistical Mechanics. Oxford University Press, 1987.
- [6] S. Succi. The lattice Boltzmann equation for fluid dynamics and beyond, Oxford University Press, 2002
- [7] G. R. McNamara and G. Zanetti. Use of the Boltzmann Equation to Simulate Lattice-Gas Automata. *Phys. Rev. Lett.* 61, 23322335, 1988
- [8] D. dHumieres. Generalized lattice-Boltzmann equations. *Progress in Astronautics and Aeronautics*, 159:450458, 1992.
- [9] D. dHumieres, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.S.Luo. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Phil. Trans. R. Soc. A*, 360:437451, 2002.
- [10] U. Frisch, D. d'Humieres, B. Hasslacher,P. Lallemand, Y. Pomeau, J.P. Rivet. Lattice gas hydrodynamics in two and three dimensions. *Complex Systems*, 1:649-707, 1987

- [11] D. Wolf-Gladrow. Lattice-Gas Cellular Automata and Lattice Boltzmann Models - An Introduction. Springer, 2005
- [12] AJC. Ladd. Lattice-Boltzmann simulations of particle-fluid suspensions. *Journal of Statistical Physics*, 2001
- [13] A. Gunstensen, D. Rothman, S. Zaleski. Lattice Boltzmann model of immiscible fluids. *Physical Review A*, 1991
- [14] N. N. Bogoliubov. Kinetic Equations. *Journal of Physics USSR* 10 (3): 265274. 1946
- [15] J. Koelman. A simple lattice Boltzmann scheme for Navier-Stokes fluid flow. *EPL (Europhysics Letters)*, 1991
- [16] S. Ansumali and I. V. Karlin. Single relaxation time model for entropic lattice Boltzmann methods. *Phys. Rev. E*, 65(5):056312, May 2002.
- [17] H. Yu, K. Zhao. Lattice Boltzmann method for compressible flows with high Mach numbers. *Phys. Rev. E* 61, 38673870, 2000
- [18] M. Sbragaglia, R. Benzi, L. Biferale, H. Chen, X. Shan, S. Succi. Lattice Boltzmann method with self-consistent thermo-hydrodynamic equilibria. *Journal of Fluid Mechanics*, 2009.
- [19] B. M. Boghosian, P. J. Love, P. V. Coveney, I. V. Karlin, S. Succi, and J. Yepez. Galilean-invariant lattice-Boltzmann models with H theorem. *Phys. Rev. E*, 68(2):025103, Aug 2003.
- [20] C. Cercignani. The Boltzmann Equation and Its Applications. Springer, New York. 1988
- [21] S. Chen, H. Chen, D. Martnez, W. Matthaeus. Lattice Boltzmann model for simulation of magnetohydrodynamics. *Phys. Rev. Lett.* 67, 37763779, 1991
- [22] P. L. Bhatnagar, E. P. Gross, and M. Krook. A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Phys. Rev.* 94, 511525, 1954
- [23] Y.Y. Al-Jahmany. Comparative study of lattice-Boltzmann and finite volume methods for the simulation of non-Newtonian fluid flows through a planar contraction. PhD thesis. 2004
- [24] H. Grad. Note on the N-dimensional Hermite polynomials. *Commun. Pure Appl. Maths*, 9:325, 1949.

- [25] H. Grad. On the kinetic theory of rarefied gases. *Commun. Pure Appl. Maths*, 9:331, 1949.
- [26] S. Chapman and T. G. Cowling. *The mathematical theory of nonuniform gases*. Cambridge University Press, Cambridge, 1970.
- [27] X. Shan and H. Chen. Lattice Boltzmann model for simulating flows with multiple phases and components. *Phys. Rev. E* 47, 18151819, 1993
- [28] C. Canuto, A. Quarteroni, M.Y. Hussaini, T.A. Zang. *Spectral methods in fluid dynamics*. Berlin. 1988.
- [29] X. Shan, X. Yuan, H. Chen. Kinetic theory representation of hydrodynamics: a way beyond the NavierStokes equation. *Journal of Fluid Mechanics*, 550 , pp 413-441, 2006
- [30] A. Quarteroni. *Numerical models for differential problems*. Springer, 2009
- [31] O. C. Zienkiewicz, R. L. Taylor, and P. Nithiarasu. *The Finite Element Method for Fluid Dynamics*, Sixth Edition. Butterworth-Heinemann, Oxford, 6th edition, December 2005.
- [32] S. Gellera, M. Krafczyka, J. Tölke, S. Turekb, J. Hronb. Benchmark computations based on lattice-Boltzmann, finite element and finite volume methods for laminar flows. *Computers and Fluids Volume 35, Issues 8-9*, 2006
- [33] X. He, L.S. Luo. Lattice Boltzmann Model for the Incompressible Navier-Stokes Equation. *Journal of Statistical Physics*, Vol. 88, Nos. 3/4, 1997.
- [34] D. Grunau, S. Chen, K. Egger. A Lattice Boltzmann Model for Multi-phase Fluid Flows. *arXiv:comp-gas/9303001v2* 1993
- [35] D. Christodoulou. The Euler Equations of Compressible Fluid Flow. *Bulletin of the American Mathematical Society* 44 (4): 581602, October 2007
- [36] P. J. Dellar. Incompressible limits of lattice Boltzmann equations using multiple relaxation times. *Journal of Computational Physics Volume 190 Issue 2*, 20 September 2003
- [37] L.S., Luo. Lattice Boltzmann Method for Computational Fluid Dynamics. lecture notes, Aug. 2003.
- [38] S. Rjasanow, W. Wagner. *Stochastic Numeric for the Boltzmann Equation*, Springer, 2005

- [39] N.I.Prasianakis. Lattice Boltzmann method for thermal compressible flows. PhD thesis, ETH, 2008
- [40] A. Caiazzo. Asymptotic Analysis of lattice Boltzmann method for Fluid-Structure interaction problems. PhD thesis, 2007
- [41] <http://en.wikipedia.org/wiki/TOP500>, or <http://www.top500.org/>
- [42] J. Lätt. Hydrodynamic Limit of Lattice Boltzmann Equations, PhD thesis, 2007
- [43] X. He, and L.S. Luo. Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation. Phys. Rev. E 56, 68116817, 1997
- [44] J. Eggels. Direct and large-eddy simulation of turbulent fluid flow using the lattice-Boltzmann scheme. International Journal of Heat and Fluid Flow Volume 17, Issue 3, June 1996
- [45] E. Hairer, S.P. Nrsett, G. Wanner. Solving ordinary differential equations: Nonstiff problems. Second revised edition, Springer, 2008
- [46] S.S. Chikatamarla. Hierarchy of lattice Boltzmann models for fluid mechanics. PhD thesis, ETH, 2008
- [47] D.P. Ziegler, Boundary condition for lattice Boltzmann simulations. Journal of Statistical Physics, Vol. 71, Nos. 5/6, 1993
- [48] F. Dubois, P. Lallemand. Towards higher order lattice Boltzmann schemes. J. Stat. Mech. 2009
- [49] T. Inamuro, M. Yoshina and F. Ogino A non-slip boundary condition for lattice Boltzmann simulations, Phys. Fluids, 7, 2928-2930, 1995
- [50] Q. Zou and X. He On pressure and velocity boundary conditions for the lattice Boltzmann BGK model, Phys. Fluids, 9, 1592-1598, 1997
- [51] P. A. Skordos Initial and boundary conditions for the lattice Boltzmann method Phys. Rev. E, 48, 4823-4842, 1993
- [52] Z. Guo, C. Zheng and B. Shi. An extrapolation method for boundary conditions in lattice Boltzmann method. Phys. Fluids, 14, 2007-2010, 2002.
- [53] O. Filippova, D. Hänel, Boundary tting and local grid refinement for lattice-BGK models, International Journal of Modern Physics C 9, 12711279,1998

- [54] R. Mei, L.-S. Luo, W. Shyy An accurate curved boundary treatment in the lattice Boltzmann method, *Journal of Computational Physics* 155, 307330, 1999
- [55] M. Bouzidi, M. Firdaouss, P. Lallemand. Momentum transfer of a lattice Boltzmann uid with boundaries. *Phys Fluids* 2001;13:34529
- [56] M. Junk, Z. Yang. Outflow boundary conditions for the lattice Boltzmann method. *Progress in Computational Fluid Dynamics*, Volume 8, Number 1-4, 2008.
- [57] D. Yu, R. Mei, L.S. Luo and W. Shyy. Viscous flow computations with the method of lattice Boltzmann equation. *Progress in Aerospace Sciences* Volume 39, Issue 5, July 2003, Pages 329-367
- [58] J. Verschaeve, B. Müller A curved no-slip boundary condition for the lattice Boltzmann method. *Journal of Computational Physics*, 2010, article in press.
- [59] J. Lätt, B. Chopard, O. Malaspinas, M. Deville, A. Michler. Straight velocity boundaries in the lattice Boltzmann method. *Phys. Rev. E* 77, 056703, 2008
- [60] J. Lätt, B. Chopard. Lattice Boltzmann Method with regularized non-equilibrium distribution functions. arXiv:physics/0506157v1 [physics.flu-dyn] 20 Jun 2005
- [61] P.A. Skordos. Initial and boundary conditions for the lattice Boltzmann method. *Phys. Rev. E* 48, 48234842, 1993
- [62] R. Mei, L.S. Luo, P. Lallemand, D. Humieres. Consistent initial conditions for lattice Boltzmann simulations. *Comp. uids*, 35:855862, 2006.
- [63] P. Lallemand, L.S. Luo. Lattice Boltzmann method for moving boundaries. *Journal of Computational Physics* Volume 184, Issue 2, 20 January 2003, Pages 406-421
- [64] Y. Li, R. Shock, R. Zhang, H. Chen. Numerical study of ow past an impulsively started cylinder by the lattice-Boltzmann method. *Journal of Fluid Mechanics* 519 (2004) 273300.
- [65] <http://www.lbmmethod.org/> (Jan. 2011)
- [66] J. Sanders, E. Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison-Wesley Professional, 2010
- [67] M. Junk, Z. Yang. One-point boundary condition for the lattice Boltzmann method. *Physical Review E* 72 (6), 066701, 2005

- [68] A. Dupuis, P. Chatelain, P. Koumoutsakos. An immersed boundary-lattice-Boltzmann method for the simulation of the flow past an impulsively started cylinder. *Journal of Computational Physics* 227, 44864498, 2008
- [69] B. Chopard, A. Dupuis. A mass conserving boundary condition for lattice Boltzmann models. *International Journal of Modern Physics B* 17,103 107, 2003
- [70] M. Junk. A finite difference interpretation of the lattice Boltzmann method. *Numerical Methods for Partial Differential Equations* Volume 17, Issue 4, pages 383-402, July 2001.
- [71] J.C.G. Verschaeve. Analysis of the lattice Boltzmann BhatnagerGrossKrook no-slip boundary condition: ways to improve accuracy and stability. *Physical Review E* 80, 036703, 2009
- [72] J. Lätt Choice of units in lattice Boltzmann simulations. technical report, April 2008,
- [73] P. Lallemand, L.S. Luo. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Phys. Rev. E* 61, 65466562, 2000
- [74] U. Ghia, K.N. Ghia, C.T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics* (ISSN 0021-9991), vol. 48, p. 387-411, Dec. 1982
- [75] J.R. Womersley. Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known. *J Physiol*, 127(3): 553-563, 1955
- [76] D.R. Golbert. Modelos de Lattice-Boltzmann Aplicados a Simulacao Computacional do Escoamento de Fluidos. Master Thesis, 2009
- [77] A. J. C. Ladd, R. Verberg. Lattice-Boltzmann Simulations of Particle-Fluid Suspensions. *Journal of Statistical Physics*, Vol. 104, Nos. 5/6, Sept. 2001
- [78] C. Manwart, U. Aaltosalmi, A. Koponen, R. Hilfer, and J. Timonen Lattice-Boltzmann and finite-difference simulations for the permeability for three-dimensional porous media. *Phys. Rev. E* 66, 016702, 2002
- [79] C.L. Lin, Y.G. Lai. Lattice Boltzmann method on composite grids. *Phys. Rev. E* 62, 22192225 ,2000

- [80] A. Dupuis, B. Chopard. Theory and applications of an alternative lattice Boltzmann grid refinement algorithm. *Phys. Rev. E* 67, 066707, 2003
- [81] M. Rohde, D. Kandhai, J. J. Derksen, H. E. A. van den Akker A generic, mass conservative local grid refinement technique for lattice-Boltzmann schemes. *International Journal for Numerical Methods in Fluids* Volume 51, Issue 4, pages 439468, 10 June 2006
- [82] J. Lätt, O. Malaspinas and M. Deville. Lecture notes at EPFL for the course of Simulating uid ow with the lattice Boltzmann method, Spring 2009
- [83] Source from WikiPedia, <http://en.wikipedia.org/wiki/Aneurysm>, October 2010
- [84] A. Peters, S. Melchionna, E. Kaxiras, J. Lätt, J. Sircar, M. Bernaschi, M. Bison, S. Succi. Multiscale Simulation of Cardiovascular flows on the IBM Bluegene/P: Full Heart-Circulation System at Red-Blood Cell Resolution. *SC '10 Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, 2010
- [85] A. Quarteroni, L. Formaggia. Mathematical Modelling and Numerical Simulation of the Cardiovascular System. In *Modelling of Living Systems*, Series Handbook of Numerical Analysis Series, 2003
- [86] D. Alemani, J. Buffle, B. Chopard, J. Galceran. Study of Three Grid Refinement Methods in the Lattice Boltzmann Framework for Reactive-Diffusive Processes. *International Journal of Modern Physics C*, Volume 18, Issue 04, pp. 722-731, 2007
- [87] C. Lenz, A. Rebel, K. Waschke, R. Koehler, T. Frietsch T. Blood viscosity modulates tissue perfusion: sometimes and somewhere. *Transfus Altern Transfus Med* 9 (4): 265272,2008