DEVELOPMENT OF LATTICE BOLTZMANN METHOD FOR COMPRESSIBLE FLOWS

QU KUN

(B. Eng., M. Eng., Northwestern Polytechnical University, China)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2008

Acknowledgements

I would like to thank Professor Shu Chang and Professor Chew Yong Tian, my supervisors, for their guidance and constant support during this research.

I had the pleasure of meeting Professor Luo Li-shi. He expressed his interest and gave me a better perspective in my work.

I am grateful to my parents, my elder sister and my lovely niece for their patience and love. Without them this work would never have come into existence.

Of course, I wish to thank the National University of Singapore for providing me with the research scholarship, which makes this study possible.

Finally, I wish to thank the following: Dr. Peng Yan (for her friendship and discussion); Dr. Duan Yi of CASC, Dr. Zhao YuXin of NUDT (for their discussion on high resolution upwind schemes); Dr. Su Wei and Zeng XianAng of NWPU (for their help on the 3D multiblock solver); Shan YongYuan, Huang MingXing, Qu Qing, Huang JunJie, Cheng YongPan, Liu Xi, Zhang ShenJun, Zeng HuiMing, Huang HaiBo, Wang XiaoYong, Xu ZhiFeng, ...(for all the good and bad times we had together); and Ao Jing whom I love forever.

Contents

Summary

As an alternative method to simulate incompressible flows, LBM has been receiving more and more attention in recent years. However, its application is limited to incompressible flows due to the used of simplified equilibrium distribution function from the Maxwellian function. Although a few scientists made effort to developing LBM for compressible flows, there is no satisfactory model. The difficulty is that the Maxwellian function is complex and difficult to manipulate. Usually, Taylor series expansion of the Maxwellian function in terms of Mach number is adopted to get a lattice Boltzmann version of polynomial form, which inevitably limits the range of Mach number. To simulate compressible flows, especially for the case with strong shock waves, we have to develop a new way to construct the equilibrium distribution function in the lattice Boltzmann context.

The aim of this work is to develop a new methodology to construct the lattice Boltzmann model and its associated equilibrium distribution functions, and then apply developed model to simulate compressible flows. In this thesis, we start by constructing a simple equilibrium function to replace the complicated Maxwellian function. The simple function is very simple and satisfies all needed relations to recover to Euler/Navier-Stokes(NS) equations. The Lagrangian interpolation is applied to distribute the simple function onto a stencil (lattice points in the velocity space) to get the equilibrium function at each direction. Several models were derived for compressible/incompressible viscous/inviscid flows with this method.

Finite volume method which can provide numerical dissipation to capture shock waves and other discontinuities in compressible flows of high Mach number with coarse grids, is

used to solve the discrete Boltzmann equation in simulations of compressible flows. At the same time, implementations of variant boundary conditions, especially the slip wall and nonslip wall conditions, are presented. The proposed models and the solution technique are verified by their applications to efficiently simulate several viscous incompressible flows, inviscid and viscous compressible flows.

At the same time, the LB models for compressible flows can be applied to develop a new flux vector splitting (FVS) scheme to solve Euler equations. The LBM based FVS scheme was tested in 1D and 3D simulations of Euler equations. Excellent results were obtained.

In a summary, a simple, general and flexible methodology is developed to construct a Boltzmann model and its associated equilibrium distribution functions for compressible flows. Numerical experiments show that the proposed model can well and accurately simulate compressible flows with Mach number as high as 10. A LBM based FVS was developed to solve Euler equations. It is believed that this work is a breakthrough in LBM simulation of compressible flows.

List of Figures

List of Tables

List of Symbols

x	Catesian coordinate
y	Catesian coordinate
t	Time
ρ	Density
u	Component of velocity
p	Pressure
e	Internal energy
E	Total energy
T	Temperature
μ	Shear viscosity
μ_b	Bulk viscosity
k	Heat conductivity
γ	Specific heat ratio
L	Reference length
U	Reference velocity
M	Mach number
Re	Reynolds number
Pr	Prandtl number
Ec	Eckert number
C_p	Specific heat at constant pressure, or pressure coefficient
C_v	Specific heat at constant volume

C_f	Skin friction coefficient
g	Equilibrium density distribution function
f	Density distribution function
c	Radius of the circular function
Ω	Collision operator
λ	Rest energy
τ	Relaxation time of BGK collision model
$\boldsymbol{\xi}$	Particle velocity
$\mathbf{e}_i$	Discrete velocity vector of lattice Boltzmann model
P	Particle momentum
D	Spatial dimension
R	Gas constant
$\mathbf{C}$	Peculiar velocity
ε	Disturbance parameter in Chapman-Enskog analysis
ϕ	Assigning function
$\mathbf{\Pi}$	Viscosity stress tensor
Q	Heat diffusion and dissipation
$\mathbf{Q}$	Mean flow variables
$\mathbf{F}$	Mean flow flux

List of Abbreviation

CFD	Computational Fluid Dynamics
NS	Navier-Stokes
FDM	Finite Difference Method
FVM	Finite Volume Method
FEM	Finite Element Method
LBM	Lattice Boltzmann Method
LBE	Lattice Boltzmann Equation
LGA	Lattice Gas Automata
BGK	Bhatnagar-Gross-Krook
TVD	Total Variation Diminishing
DVBE	Discrete Velocity Boltzmann Equation
CF-VIIF	Circular Function for Viscous Incompressible Isothermal Flow
CF-ICF	Circular Function for Inviscid Compressible Flow
CF-VCF	Circular Function for Viscous Compressible Flow
PIC	Particle In Cell
VIC	Vortex In Cell
MUSCL	Monotone Upstream Scheme for Conservation Laws
CFL	Courant-Friedrich-Levy
GAMM	Gesellschaft für Angewandte Mathematik und Mechanik
ENO	Essentially Non-Oscillatory
WENO	Weighted Essentially Non-Oscillatory

MPI	Message Passing Interface
FVS	Flux Vector Splitting
IMEX	IMplicit-EXplicit
AMR	Adaptive Mesh Refine
MRT	Multi-Relaxation-Time

Chapter 1

Introduction

In this chapter, we will give a brief introduction to computational fluid dynamics and the lattice Boltzmann method. Then a review on development of LBM in compressible flows is presented. After that, the objective of this thesis and the organization of the thesis are described.

1.1 Computational fluid dynamics

Nowadays, computational fluid dynamics (CFD) has been developed into an important subject of fluid dynamics as computers are becoming more and more powerful. The core of CFD is to numerically solve governing equations of fluid dynamics with proper initial conditions and boundary conditions to get the behavior of dependent variables (density, velocity, pressure, temperature ...) in flow fields.

For Newtonian fluids, the governing equation is a set of the second order partial difference equations (PDE), which are usually called compressible Navier-Stokes equations written as

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_\alpha}{\partial x_\alpha} = 0 \tag{1.1a}$$

$$\frac{\partial \rho u_\alpha}{\partial t} + \frac{\partial \rho u_\alpha u_\beta}{\partial x_\beta} = -\frac{\partial p}{\partial x_\alpha} + \frac{\partial \Pi_{\alpha\beta}}{\partial x_\beta} \tag{1.1b}$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial \rho E u_\alpha}{\partial x_\alpha} = -\frac{\partial p u_\alpha}{\partial x_\alpha} + \frac{\partial}{\partial x_\beta}\left(k\frac{\partial T}{\partial x_\beta} - \Pi_{\alpha\beta}u_\alpha\right) \tag{1.1c}$$

where

$$\begin{aligned} p &= \rho RT = (\gamma - 1)\,\rho e \\ E &= C_v T + \frac{1}{2}u^2 \\ C_v &= R/(\gamma - 1) \\ \Pi_{\alpha\beta} &= \mu\left(\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} - \frac{2}{3}\frac{\partial u_\chi}{\partial x_\chi}\delta_{\alpha\beta}\right) + \mu_b\frac{\partial u_\chi}{\partial x_\chi}\delta_{\alpha\beta} \end{aligned}$$

and μ is the viscosity, μ_b is the bulk viscosity, k is the thermal conductivity, γ the specific heat ratio, C_v is the constant-volume specific heat capacity, R is the gas constant. For incompressible flows, since the work done by viscous stress and pressure is usually very small and can be neglected, Equs. (1.1) can be simplified to

$$\frac{\partial u_\alpha}{\partial x_\alpha} = 0 \tag{1.2a}$$

$$\frac{\partial u_\alpha}{\partial t} + u_\alpha\frac{\partial u_\beta}{\partial x_\beta} = \frac{1}{\rho}\left[-\frac{\partial p}{\partial x_\alpha} + \frac{\partial}{\partial x_\beta}\left(\mu\frac{\partial u_\alpha}{\partial x_\beta} + \mu\frac{\partial u_\beta}{\partial x_\alpha}\right)\right] \tag{1.2b}$$

$$\frac{\partial T}{\partial t} + \frac{\partial T u_\alpha}{\partial x_\alpha} = \frac{\partial}{\partial x_\alpha}\left(k\frac{\partial T}{\partial x_\alpha}\right) \tag{1.2c}$$

Equs (1.1) can be reduced to Euler equations when viscous effects are neglected.

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_\alpha}{\partial x_\alpha} = 0 \\ &\frac{\partial \rho u_\alpha}{\partial t} + \frac{\partial(\rho u_\alpha u_\beta + p)}{\partial x_\alpha} = 0 \\ &\frac{\partial \rho E}{\partial t} + \frac{\partial(\rho E + p)u_\alpha}{\partial x_\alpha} = 0 \end{aligned}$$

With other assumptions, many other types of simplified governing equations can be derived, such as the potential equation, the parabolized NS equations, the thin-layer NS equations...

Traditionally, the finite difference method (FDM), the finite volume method (FVM) and the finite element method (FEM) are the three most popular numerical discretization methods used in CFD. The fundamental idea of FDM is to approximate derivatives in governing equations by a local Taylor series expansion at grid points in the adopted grid. In a uniform rectangular grid, FDM is accurate, efficient and easy to implement. And it can be applied to structural grids in complex domains with coordinate transformation. But FDM can not be applied in unstructured meshes, which hinders its applications in complex domains. In contrast, FEM is natural for unstructured meshes, so it is more suitable for complex domains than FDM. In FEM, the whole domain is divided into many small elements. In every element, unknown variables are approximated by linear combination of a set of base functions. Usually, the Galerkin method, also named as the weighted residual method, is applied to discretize governing equations. In the framework of Galerkin method, it is easy to achieve high order of accuracy. Those FEM with high order polynomials as base functions, called spectral element methods, can achieve very high order spatial accuracy. However, traditional FEM has some difficulties in solving convective problems such as compressible flows if additional stabilizing techniques are not applied explicitly. Unlike FDM and FEM, FVM solves integral form of governing equations. The whole domain is divided into many small volumes. In every small volume, the adopted governing equations of integral form which describe physical conservation laws are discretized. As compared with FDM and FEM, FVM can not only be applied to solve elliptical and parabolic PDE but also hyperbolic system. And FVM can also be applied on unstructured meshes. On the other hand, we have to indicate that it is not easy to implement schemes of high order accuracy (higher than second order) in FVM. So, a new method, the discontinuous Galerkin method, or discontinuous finite element method, was developed in recent years. It is a more general FEM for conservation laws and combines the feature of traditional FVM and FEM.

Beside these methods, which are usually used to solve NS/Euler equations, there are some other approaches in CFD to simulate flows which may not be well described by NS/Euler equations. These approaches include the kinetic method, direct simulation

Monte Carlo, the lattice Boltzmann method, dissipative particle dynamics method, the molecular dynamics method, and so on.

1.2 Lattice Boltzmann method

1.2.1 Basis of lattice Boltzmann method

In the last decade, as a new and promising method of computational fluid dynamics, LBM, developed from lattice gas automata (LGA) [1], was widely studied. It has been applied in isothermal/thermal viscous flows, bubble dynamic simulations, multiphase/multicomponent flows, turbulent flows, flows in porous media, elastic-viscous flows, particle suspension, microflows, etc.

The standard lattice Boltzmann equation (LBE) is written as

$$f_i\left(\mathbf{x}+\mathbf{e}_i, t+1\right) = f_i\left(\mathbf{x}, t\right) + \Omega_i \tag{1.3}$$

where the density distribution function $f_i(\mathbf{x}, t)$ is regarded as a pseudo particle mass, $\mathbf{e}_i$ is the velocity of f_i, and Ω_i is a local collision operator. For example, D2Q9 (which means a nine-velocity model in two dimensional space), a two-dimensional model for isothermal incompressible flows, as shown in Fig. 1.1, has 9 density distribution functions at every node in its lattice. Equ. (1.3) also describes the solution procedure. First, at every node, all pseudo particles ($f_i, i = 0 \ldots 8$) collide together according to the collision operator Ω_i and get updated at the time t . This is the collision step. Then, every updated pseudo particle $f_i + \Omega_i$ at $\mathbf{x}$ streams in the i-th direction with its own velocity $\mathbf{e}_i$. It arrives at the location $\mathbf{x} + \mathbf{e}_i$, after one unit time step, at the time $t + 1$. This is called the streaming step. After that, a new loop of collision-streaming begins. And the mean flow variables can be computed as

$$\begin{cases} \rho = \sum\limits_i f_i \\ \rho\mathbf{v} = \sum\limits_i f_i \mathbf{e}_i \end{cases} \tag{1.4}$$

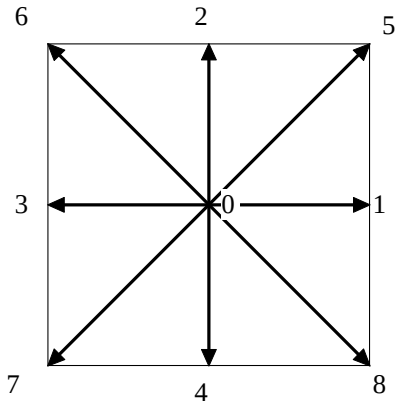

Figure 1.1: Schematic of D2Q9 model

Usually, the BGK collision model [2] is used as the collision operator

$$\Omega_i = -\frac{f_i - f_i^{eq}}{\tau} \tag{1.5}$$

to obtain the BGK-LBE

$$f_i\left(\mathbf{x} + \mathbf{e}_i, t + 1\right) = f_i\left(\mathbf{x}, t\right) - \frac{f_i - f_i^{eq}}{\tau} \tag{1.6}$$

where τ is the relaxation time, f_i^{eq} is the equilibrium distribution function at the i-th direction. f_i^{eq} only depends on the local density ρ, velocity $\mathbf{v}$ and temperature T (in case of thermal models) at $\mathbf{x}$ and t . This collision model means that every density distribution function f_i relaxes to its equilibrium state f_i^{eq} in time τ . So, these two variables are the most important factors in BGK-LBE. f_i^{eq} , usually in polynomial form, is determined in such a way that the macroscopic variables obtained from LBE can satisfy NS equations. τ is related to viscosity of fluid. Besides the BGK model, other collision operators can be used, such as double relaxation time collision [3] and multi-relaxation time collision [4, 5]. These collision operators were proposed to achieve better stability and get more parameters to adjust the properties of the LB models.

1.2.2 Lattice Boltzmann models

Since LBE has a very simple form and f_i^{eq} is in polynomial form, implementation of LBM with computer code is very easy. The difficulty is how to derive a lattice model $\mathbf{e}_i$ and its equilibrium distribution functions, f_i^{eq}. Presently, there are two kinds of deriving methods to construct LB models. The first is the undetermined coefficient method. Another way is the Hermite tensor expansion method.

In the undetermined coefficient method, f_i^{eq} is first assumed as a polynomial with unknown coefficients. Usually, the polynomial is the expanded Maxwellian function as shown below

$$g = \frac{\rho}{(2\pi RT)^{D/2}} \exp\left[-\frac{(\boldsymbol{\xi}-\mathbf{u})^2}{2RT}\right] \tag{1.7}$$

With the physical conservation laws, the isotropic relations of the lattice tensors and some other assumptions, the coefficients in the polynomial expansion can be determined [6, 7].

The Hermite tensor expansion method was proposed by He, Luo, Shan, *etc. al* [8–11]. In [8, 9], He and Luo began from the continuous BGK-Boltzmann equation

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi}\cdot\nabla f = -\frac{f-g}{\tau} \tag{1.8}$$

and turned it into

$$f(\mathbf{x}+\boldsymbol{\xi}\delta t, \boldsymbol{\xi}, t+\delta t) - f(\mathbf{x}, \boldsymbol{\xi}, t) = -\frac{f(\mathbf{x}, \boldsymbol{\xi}, t) - g(\mathbf{x}, \boldsymbol{\xi}, t)}{\tau} \tag{1.9}$$

by integration. Unlike the undetermined coefficient method, there is no need to assume a polynomial form of f_i^{eq} in this method. Alternatively, they expanded the Maxwellian function with Taylor series under the assumption of small Mach number

$$\begin{aligned} g \quad &\approx \frac{\rho}{(2\pi RT)^{D/2}} \exp\left(-\boldsymbol{\xi}^2/2RT\right) \\ &\times \left[1 + \frac{(\boldsymbol{\xi}\cdot\mathbf{u})}{RT} + \frac{(\boldsymbol{\xi}\cdot\mathbf{u})^2}{(RT)^2} - \frac{\mathbf{u}^2}{2RT}\right] \end{aligned}$$

By setting

$$\psi\left(\boldsymbol{\xi}\right)=\frac{\rho}{\left(2\pi RT\right)^{D/2}}\times\left[1+\frac{\left(\boldsymbol{\xi}\cdot\mathbf{u}\right)}{RT}+\frac{\left(\boldsymbol{\xi}\cdot\mathbf{u}\right)^{2}}{\left(RT\right)^{2}}-\frac{\mathbf{u}^{2}}{2RT}\right]$$

the Maxwellian function can be approximated by

$$f^{eq}\approx\exp\left(-\boldsymbol{\xi}^{2}/2RT\right)\psi\left(\boldsymbol{\xi}\right) \tag{1.10}$$

where D is the dimension of the space, $\boldsymbol{\xi}$ is the particle velocity, $\mathbf{u}$ is the flow mean velocity, R is the gas constant, T and ρ are mean flow density and temperature. Up to the third order moment of f^{eq} in $\boldsymbol{\xi}$, $I=\int_{-\infty}^{+\infty}f^{eq}\boldsymbol{\xi}\boldsymbol{\xi}\boldsymbol{\xi}d\boldsymbol{\xi}$, should be computed exactly during the process of recovering the isothermal incompressible NS equations by Chapman-Enskog analysis. Since $\psi(\boldsymbol{\xi})$ is a second order polynomial of $\boldsymbol{\xi}$, the integral can be written as

$$I=\int_{-\infty}^{+\infty}e^{-\boldsymbol{\xi}^{2}/2RT}P^{(5)}(\boldsymbol{\xi})d\boldsymbol{\xi}$$

$$\text{set }\boldsymbol{\varsigma}=\boldsymbol{\xi}/\sqrt{2RT}\Rightarrow I=\int_{-\infty}^{+\infty}e^{-\boldsymbol{\varsigma}^{2}}P^{(5)}(\boldsymbol{\varsigma})d\boldsymbol{\varsigma} \tag{1.11}$$

where $P^{(5)}(\boldsymbol{\varsigma})$ is the fifth order polynomial of $\boldsymbol{\varsigma}$. Given proper discretization of $\boldsymbol{\varsigma}$, Equ. (1.11) can be evaluated with desirable accuracy by Gaussian-Hermite quadrature

$$\int_{-\infty}^{+\infty}e^{-\boldsymbol{\varsigma}^{2}}P^{(5)}(\boldsymbol{\varsigma})d\boldsymbol{\varsigma}=\sum_{j}W_{j}P^{(5)}(\boldsymbol{\varsigma}_{j}) \tag{1.12}$$

where $\boldsymbol{\varsigma}_{j}$ and W_{j} are the abscissas and weights of Gaussian-Hermite quadrature. So the equilibrium distribution function f_{i}^{eq} and the discrete velocity vector $\boldsymbol{\xi}$ can be derived. D2Q7, D2Q9 and D3Q27 were derived with this method in [8, 9]. Since this method does not need an assumed polynomial, it is more natural than the undetermined coefficient method. It should be indicated that without the assumption of small Mach number and isothermal condition, the Gaussian-Hermite quadrature can not be applied. For this reason, it is not fit for developing multi-velocity thermal models.

He and Shan *et. al* [10, 11] further studied this method and proposed a new version, in

which Taylor series expansion is not used. Instead, the Maxswellian function is expanded in the velocity space with Hermite tensor

$$g\left(\boldsymbol{\xi}\right) \approx g^{(N)}\left(\boldsymbol{\xi}\right) = \omega\left(\boldsymbol{\xi}\right) \sum_{n=0}^{N} \frac{1}{n!} a^{(n)} \mathcal{H}^{(n)}\left(\boldsymbol{\xi}\right) \tag{1.13}$$

where $\mathcal{H}^{(n)}\left(\boldsymbol{\xi}\right)$ is the n-th order Hermite polynomial of $\boldsymbol{\xi}$, $a^{(n)}$ is its coefficient and $\omega\left(\boldsymbol{\xi}\right)$ is the weight function. The third order expansion of the Maxwellian function is

$$\begin{aligned} g \quad \approx & \omega\left(\boldsymbol{\xi}\right) \rho \{1 + \boldsymbol{\xi} \cdot \mathbf{u} + \underbrace{\frac{1}{2}\left[\left(\boldsymbol{\xi} \cdot \mathbf{u}\right)^2 - u^2 + \left(\theta - 1\right)\left(\boldsymbol{\xi}^2 - D\right)\right]}_{\text{2nd order}} \\ & + \underbrace{\frac{\boldsymbol{\xi} \cdot \mathbf{u}}{6}\left[\left(\boldsymbol{\xi} \cdot \mathbf{u}\right) - 3u^2 + 3\left(\theta - 1\right)\left(\boldsymbol{\xi}^2 - D - 2\right)\right]}_{\text{3rd order}} \} \end{aligned}$$

The second order expansion is needed for isothermal flows, and the third order expansion is for thermal flows. Similar to [8, 9], the discrete velocities can be determined from Gaussian-Hermite polynomial.

1.3 LB models for compressible flows

LBM has been used to simulate incompressible flows since it was invented. To extend LBM to simulate compressible flows, there are two aspects of work to be done. First, it is to develop LB models for compressible flows. Second, it is to find feasible numerical methods. Compressible flows are thermal flows in nature. But thermal LB models are not as mature as isothermal LB models. For incompressible thermal flows, there are several feasible models. But for compressible flows, the current models still encounter some difficulties and there is no satisfactory model for compressible flows so far. In this section, we will review the current LB models for thermal flows (both incompressible and compressible) to evaluate their ideas, advantages and disadvantages, from which we may explore new LB models for compressible flows.

1.3.1 Current LB models for incompressible thermal flows

Present LBM for incompressible thermal flows can be divided into three classes: the multi-velocity models, the multi-distribution-function models and the hybrid method.

Multi-velocity models are natural extension of the isothermal models. They use only the density distribution function in the collision-streaming procedure. f_i^{eq} of these models contains higher order terms of particle velocity in order to recover the energy equation and additional particle velocities are necessary to obtain a higher order isotropic lattice since higher order lattice tensors are needed in Chapman-Enskog analysis. Alexander, Chen, Sterling [7] expanded the Maxwellian function to the third order of velocity and developed the first multi-velocity thermal model with 13 velocities. Although it could provide the basic mechanisms of heat transfer, it had some nonlinear deviations and did not recover the right energy equation. Afterwards, Chen [12, 13] proposed 1D5V and 2D16V models which can get rid of the nonlinear deviations and could recover the right energy equation by assuming another form of polynomial with more terms. Nevertheless, multi-velocity models suffer severe numerical instability and temperature variation is limited to a narrow range. Recently, Shan [11] stated that multi-velocity models could be derived by the third order Hermite tensor expansion of the Maxwellian function. However, it still needs further verification.

Unlike multi-velocity models of single density distribution function, multi-distribution-function models contain another distribution function of internal energy or temperature, while velocity and pressure are computed with standard LBM. It is known that the temperature can be regarded as a passive-scalar component transported by the velocity field when the compression work and the viscous heat dissipation are neglected. Based on this knowledge, some researchers proposed passive-scalar approach to simulate thermal flows. Shan [14] derived the scalar equation for temperature based on two-component models. Shan simulated two-dimensional and three-dimensional Rayleigh-Benard convection. The results agreed well with previous numerical data obtained using other methods. This approach shows better stability than multi-velocity models. Afterwards, He [15] proposed a

two-distribution-function model. Different from the passive-scalar approach, the temperature distribution function was derived from the Maxwellian function and compression work and viscous heat dissipation were incorporated. And the numerical results of this model agree excellently with those in previous studies. Its numerical stability is similar to that of the passive-scalar approach. Peng *etc. al* [16] further simplified He's model to neglect the compression work and viscous heat dissipation so that the computational effort is greatly reduced. This modified thermal model has no gradient term and is easier to be implemented. As compared with multi-velocity models, multi-distribution-function models are able to simulate flows of larger range of temperature variation. Therefore multi-distribution-function models are much more popular than multi-velocity models.

The third way, the hybrid method, is to simulate temperature field by other approaches, such as FDM [17–19]. The method was first proposed by Filippova in [16, 17] to simulate low Mach number reactive flows with significant density changes. In [19], Lallemand and Luo proposed their hybrid thermal LBE. With multi-relaxation-time (MRT) collision operator, their method gave excellent results and better stability than multi-velocity thermal models. The hybrid method has better efficiency since the number of distribution functions is less than that in the other two methods. It should be indicated that this way is not a pure LBM and it is a kind of compromise.

All these models are related to the Maxwellian function. Except for Shans multi-velocity models, all the other models adopt Taylor expansion of the Maxwellian function with the small Mach number assumption. Hence they can only be applied to simulate incompressible flows. Shan's multi-velocity model is quite new and has not been tested for real problems, and its capability in simulating compressible flows is still unknown. Minoru [20, 21] gave a detailed derivation of multi-velocity models and presented several models. His models gave better results than the models in [7, 12, 13]. He also simulated some cases of compressible flows in [22]. But these cases are not persuadable since no classic case in the community of simulating compressible flows was presented. It seems that these LB models could not be applied to simulate compressible flows and it might be difficult to deriving LB models for compressible flows from the Maxwellian function.

Moreover, we believe that it is impossible to derive a satisfactory LB model for compressible flows from Equ. (1.7). The reason is that Equ. (1.7) only describes monatomic molecules which have D transversal degrees of freedom while diatomic and polyatomic molecules have more degrees of freedom. For example, diatomic molecules have 5 degrees of freedom (3 transversal degrees and 2 rotating degrees) under usual condition. These extra degrees of freedom are important and they determine specific heat ratio γ of a gas. Considering the extra degrees of freedom, the Maxwellian function should be [23]

$$g\left(\xi\right)=\frac{\rho}{\left(2\pi RT\right)^{\frac{K+D}{2}}}\exp\left[-\frac{\left(\boldsymbol{\xi}-\mathbf{u}\right)^{2}+\sum\limits_{i=1}^{K}\xi_{i}^{2}}{2RT}\right] \tag{1.14}$$

where D is space dimension, $\boldsymbol{\xi}_i, i=1\cdots K$ are extra degrees of freedom, and K is the number of extra degrees $K=b-D$. Notably, D is the dimension of the mean flow. In case of two dimensional flows, the other transversal degree of freedom is regarded as an extra degree. Here, b is the total number of degrees of freedom and it determines the specific heat ratio of a gas as

$$\gamma=\frac{b+2}{b}$$

For example, $b=3$ for monatomic gases, $b=5$ for diatomic gases and $b=5,6$ for linear and nonlinear triatomic gases, respectively. In case of incompressible flows, the extra degrees have little effect and K can be set as zero. Thus, Equ. (1.14) is reduced to Equ. (1.7). Consequently, the modified Maxwellian function Equ. (1.14) is more complicated than Eq.(1.7). It is more difficult to mathematically manipulate this modified Maxellian function. Nevertheless, we believe that deriving LB models from a known equilibrium function is more natural than the undetermined coefficients method. This idea should be very useful for developing LB models for compressible flows.

1.3.2 Current LB models for compressible flows

As introduced in the last subsection, the polynomial forms of f_i^{eq} usually come from the Maxwellian function by Taylor series expansion on Mach number. Thus Mach number should be very small. Otherwise, the truncation error of the series expansion could be very large. The small Mach number condition limits the application of LBM to incompressible flows. Thus, it seems that the expanded Maxwellian function results in difficulties for developing LB models for compressible flows. If the polynomial is assumed without the Maxwellian function, LB models for compressible flows might be obtained by means of the under determined coefficient method. The question is how to propose a good polynomial form. Previous studies [24, 25] showed that f_i^{eq} derived from this method contains many free parameters which have to be tuned carefully to make simulations stable. Nevertheless, towards this direction, some LB models for compressible flows were proposed. In this subsection, we briefly introduce four LB models for compressible flows. Although there are some other models, these models are typical models. In the following, we only introduce these four models.

The model of Yan *et al.* [24]

Yan, Chen and Hu [24] proposed a 2D 9-bit model with two energy levels. Their model can recover Euler equations. In this model, the lattice is assumed the same as that of D2Q9 model (Fig. 1.1), but there are two particles, f_{iA} and f_{iB} $(i = 1 \ldots 8)$ at each lattice velocity direction except at the static site $(i = 0)$. Thus there are 17 particles. These 17 particles are grouped into 3 energy-levels: ε_A for f_{iA} , ε_B for f_{iB} , and ε_D for the static particle $(i = 0)$. Macroscopic variables are defined as

$$\rho = \sum_{i,X=A,B,D} f_{iX} \tag{1.15}$$

$$\rho\mathbf{u} = \sum_{i,X=A,B,D} f_{iX}\mathbf{e}_i \tag{1.16}$$

$$\frac{1}{2}\rho\mathbf{u}^2 + \rho E = \sum_{i,X=A,B,D} f_{iX}\varepsilon_X \tag{1.17}$$

and the flux conditions of momentum and energy are:

$$\rho\mathbf{u}\mathbf{u} + p\mathbf{I} = \sum_{i,X=A,B,D} f_{iX}^{eq}\mathbf{e}_i\mathbf{e}_i \tag{1.18}$$

$$\left(\frac{1}{2}\rho\mathbf{u}^2 + \rho E + p\right)\mathbf{u} = \sum_{i,X=A,B,D} f_{iX}^{eq}\varepsilon_i\mathbf{e}_i \tag{1.19}$$

The polynomial form of f_i^{eq} is assumed the same as that of D2Q9. But the coefficients are different for different energy-levels. These coefficients can be determined from the above relations and isotropy of the lattice tensors. The Sod and Lax shock tube problems were successfully simulated with the streaming-collision procedure by this model. Its drawback is that there are a number of free parameters in the model to be specified.

The model of Shi *et al.* [25]

Shi *et al.* [25] constructed a 2D 9-bit model which can recover Euler equations. They used the same lattice with D2Q9 (Fig. 1.1), and the form of f_i^{eq} was assumed to be the same as that of D2Q9 model. The conservation laws and flux relations are defined in Equ. (1.21). They also introduced three rest-energy-levels: ε_A (for $i = 1, 2, 3, 4$), ε_B (for $i = 5, 6, 7, 8$) and ε_D (for $i = 0$). The rest energy of particles, standing for energy of extra degrees of freedom, can make heat special ratio adjustable. With these relations and isotropy of the lattice tensors as well as some other assumptions, the unknown coefficients can be determined.

Instead of using the collision-streaming procedure, Harten's minmod TVD finite difference scheme [26] was used in this model to solve the discrete velocity Boltzmann equation (DVBE) of the BGK type

$$\frac{\partial f_i}{\partial t} + \mathbf{e}_i \cdot \nabla f_i = -\frac{f_i - f_i^{eq}}{\tau} \tag{1.20}$$

The energy levels in this model should be chosen carefully to guarantee positivity of f_i^{eq}.

$$\rho = \sum_i f_i^{eq} \tag{1.21a}$$

$$\rho \mathbf{u} = \sum_i f_i^{eq} \mathbf{e}_i \tag{1.21b}$$

$$\frac{1}{2}\rho \mathbf{u}^2 + \rho E = \sum_i f_i^{eq} (\frac{1}{2}\mathbf{e}_i^2 + \varepsilon_i) \tag{1.21c}$$

$$\rho \mathbf{u}\mathbf{u} + p\mathbf{I} = \sum_i f_i^{eq} \mathbf{e}_i \mathbf{e}_i \tag{1.21d}$$

$$\left(\frac{1}{2}\rho \mathbf{u}^2 + \rho E + p\right)\mathbf{u} = \sum_i f_i^{eq} (\frac{1}{2}\mathbf{e}_i^2 + \varepsilon_i)\mathbf{e}_i \tag{1.21e}$$

The models of Kataoka and Tsutahara [27, 28]

Kataoka and Tsutahara [27] proved that, in the limit of small Knudsen number, DVBE could approach Euler equations in smooth region and if stiff region could not be resolved, DVBE could approach the weak form of Euler equations as long as a consistent numerical scheme is used to discretize DVBE. This important conclusion suggests that discontinuity-capturing schemes could be applied to discretize DVBE in order to capture discontinuities on coarse grids. They also developed some LB models for compressible flows. For the two dimensional inviscid flows, their method is similar to the work of Shi *et al.* [25]. The polynomial of f_i^{eq} is the same as that of D2Q9, but the rest energy is only available on the rest particle. Although the model has 9 velocity vectors, the configuration of the lattice is different from that of D2Q9.

Besides LB models for inviscid compressible flows, a two dimensional 16-velocity model for viscous compressible flows was presented in [28]. The form of f_i^{eq} is approximated by a high order polynomial, which satisfies the conservation laws and flux relations, as well as two dissipative relations of momentum and energy. Similar to the work of Shi *et al.* [25], the second order upwind FDM scheme was used to solve DVBE, and two shock tube problems and Couette flows were simulated effectively. However, their models showed numerical instability when Mach number exceeds 1. The reason for this is not clear.

Although these three models have many limitations, they give some useful hints for simulation of compressible flows by LBM. First, the use of the Maxwellian distribution function or its expanded form might not be necessary in the LBM simulation of com-

pressible lows. The polynomial forms of f_i^{eq} in these models for inviscid compressible flows are all assumed as that of D2Q9. Although f_i^{eq} of D2Q9 can be derived from the Maxwellian function, its form loses the original meaning in case of compressible flows with adjustable γ value. It seems that these LB models for inviscid flows just occasionally take the polynomial form of f_i^{eq} of D2Q9. The second is that, the rest energy should be introduced to make γ be adjustable. If there is no rest energy, the total energy of a particle only consists of transversal energy and cannot consider other degrees of freedom, making γ fixed as $\frac{D+2}{D}$. The third is that, it might be more feasible to solve DVBE by FDM, FVM or FEM to simulate compressible flows with discontinuities. According to Kataoka's proof [27], DVBE approaches to Euler equations with an error of $O(\varepsilon)$, which means that Knudsen number ε should be very small. However, the dimensionless width of a discontinuity is of the order of $O(\varepsilon)$. So the mesh size should be much smaller than $O(\varepsilon)$ if there is no artificial viscosity. It is impractical to simulate discontinuities in such fine meshes. Fortunately, Kataoka and Tsutahara proved that DVBE discretized with a consistent numerical scheme of the p-th order spatial accuracy is consistent with the weak form solution of Euler equations even if the mesh size is much larger than $O(\varepsilon)$ and the error is $\max(O(\Delta x^p), O(\varepsilon))$ [27]. Since we have to capture discontinuities, the adopted schemes should have TVD, TVB or ENO/WENO characteristic. Many works have been done in solving hyperbolic conservation laws with FDM, FVM and FEM in the last decades. Modern FDM, FVM and FEM have many advanced features in simulation of compressible flows with high Mach number and discontinuities. These methods can be adopted to efficiently solve DVBE.

On the other hand, it has to be indicated that so far, only results of subsonic cases with weak shock waves were presented in these models. Simulating high Mach number compressible flows with strong shock waves by LBM is still a challenging issue. Further work is needed in this field.

The adaptive LB model of Sun [29–33]

Towards simulating compressible flows, Sun [29–33] developed the adaptive LB model. The pattern of the lattice velocities of this model varies with the mean flow velocity and internal energy. The adaptive LB model contains several discrete velocity vectors which are symmetrically located around the mean velocity in the velocity space. As shown in Fig. 1.2, the density is equally distributed on all the discrete velocity vectors.

This is just like the molecular velocity in the kinetic theory: $\boldsymbol{\xi} = \mathbf{V} + \mathbf{C}$ where $\boldsymbol{\xi}$ is the molecular velocity, $\mathbf{V}$ is the mean velocity and $\mathbf{C}$ is the peculiar velocity. This adaptive LBM permits mean flow to have high Mach number. However, the relaxation parameter τ can only be set as one because the discrete velocity vectors vary with mean velocity. Thus the viscosity can not be adjusted by changing τ . The model can only be used to simulate inviscid flows if the viscosity terms are regarded as numerical dissipation. Some cases of compressible flows with weak or strong shock waves were successfully simulated by the adaptive LBM. On the other hand, we have to indicate that unlike the conventional LBM, density, momentum and energy are all needed to be transported with nonlinear convection (streaming) in the adaptive LBM. Therefore, it is more like a special flux vector splitting (FVS) scheme, rather than a pure LBM.

Nevertheless, the adaptive LBM is very illuminative. The form of its f_i^{eq} seems very simple and has nothing to do with the Maxwellian function, but can recover to NS equation. It prompts us that the Maxwellian function could be substituted with some other functions while keeping the property of recovering Euler/NS equations. This idea is very important for us to develop new lattice models to simulate compressible flows.

1.4 Objective of this thesis

The objectives of this thesis are to develop a general and simple methodology to construct new LB models and associated equilibrium distribution functions for compressible flows, explore relative boundary conditions and numerical methods, and simulate compressible flows with developed models.

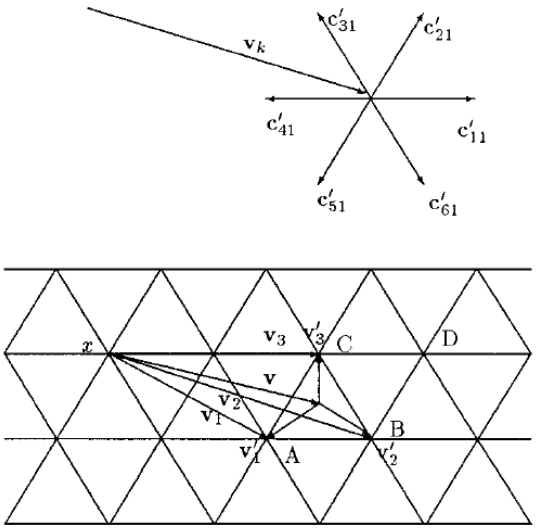

Figure 1.2: Streaming in the adaptive LBM of Sun

The first task of this work is to develop a new deriving method which is suitable to construct LB models for compressible flows. Due to limitation of the two traditional deriving methods, it is difficult to construct satisfactory LB models for compressible flows. Therefore, it is necessary to develop a new deriving method. This method should need fewer assumptions so as to decrease ambiguities, and it should be easy to implement, general to extend to different problems (1D/2D/3D, inviscid/viscous).

The second is to develop proper boundary conditions and numerical method for LBM simulation of compressible flows. Compared with incompressible flows, compressible flows are more complicated. They are thermal flows naturally and may have discontinuities, such as shock waves and contact interfaces which may interact with shear layers (for viscous flows), producing very complex phenomena. Compressible flows need more boundary conditions of which some are complicated and need sophisticated mathematical processing. Hence, apart from the LB models, much work is also needed to implement boundary conditions and numerical discretization.

Last task is to verify the above work. They will be implemented into computer code to simulate compressible flows to validate all the deriving method, models, boundary conditions and numerical methods.

1.4.1 Organization of this thesis

The thesis is organized as follows:

Chapter 2 preliminarily describes a new deriving method. A simple function is proposed to replace the Maxwellian function. This function has a very simple form and is easy to manipulate. It can be discretized onto a lattice to derive the LB model. The constraints of discretization, as well as how to determine the configuration of a proper lattice are also discussed. A new D2Q13 LB model for isothermal incompressible flows is derived to examine the idea. With this model, the lid driven cavity flow was simulated by means of the traditional streaming-collision procedure and the finite difference method.

In Chapter 3, the deriving method is extended to develop LB models for inviscid compressible flows. Another simple function is constructed. With the same discretizing method, some LB models for inviscid compressible flows are derived. At the same time, numerical methods, including spatial discretization schemes, time integral methods and implementation of boundary conditions are discussed. Some inviscid compressible flows with weak and strong shock waves are simulated successfully by the present models and numerical methods.

Chapter 4 presents how to construct a LB model for viscous compressible flows. We construct the third simple function to replace the Maxwellian function. Several numerical results are shown to validate our models and numerical methods.

In Chapter 5, a new idea of combination of the LB model and a classical Euler FVM solver is proposed. A new flux vector splitting (FVS) scheme is developed based on D1Q4L2 LB model. One dimensional and multidimensional numerical tests produce excellent results.

Chapter 6 concludes the present work and suggests recommendation for the future study.

Chapter 2

A New Way to Derive Lattice Boltzmann Models for Incompressible Flows

This chapter proposes a new way to derive lattice Boltzmann models. In this method, a simple circular function replaces the Maxwellian function. Thanks for its simplicity, it can be easily manipulated to derive LB models. Based on the idea, a new D2Q13 LB model for incompressible flows is derived in this chapter.

2.1 A simple equilibrium distribution function, CF-VIIF

It has been known that f_i^{eq} in LBM is usually related to the Maxwellian function. As LBM is a kind of discrete velocity method in which the continuous velocity space is replaced by a set of lattice velocities, an interesting question is whether the Maxwellian function or its simplified form is really needed in LBM. In fact, the Maxwellian function is in the exponential form, which cannot be directly applied in LBM. As shown in Chapter 1, its simplification to the polynomial form is necessary in LBM. On the other hand, we may be able to find a simple form of equilibrium function, g_s, which can make the Boltzmann

equation of BGK type

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla f = \left(g_s - f\right)/\tau \tag{2.1}$$

recover the macroscopic governing equations (Euler/NS equations). If this g_S is very simple and easy to manipulate mathematically, it might be used to derive LB models which are difficult to construct with the traditional methods described in Chapter 1. In this chapter, in order to examine this idea, we will develop a LB model for two-dimensional viscous isothermal incompressible flows as this problem is the simplest case.

Before looking for the simple function, we firstly study the Maxwellian function. For isothermal incompressible flows, in deriving NS equations from Chapman-Enskog analysis, the continuity equation and the momentum equation are the results of the zeroth and first order moments in the velocity space

$$\begin{aligned} &\int \left[\partial_t f + \boldsymbol{\xi} \cdot \nabla f = -\frac{f-g}{\tau}\right] \begin{bmatrix} 1 \\ \boldsymbol{\xi} \end{bmatrix} d\boldsymbol{\xi} \\ \Rightarrow &\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0 \\ \partial_t (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v}\mathbf{v} + p\mathbf{I}) = \nabla \cdot \boldsymbol{\Pi} \end{cases} \end{aligned}$$

In Chapam-Enskog expansion, only the first two terms of approximations for f are considered. Therefore, given the equilibrium distribution function, the Maxwellian function g, up to the third order moment of g must be able to be computed. The moments (from the zeroth order to the third order) of the Maxwellian function are listed in Equ. (2.2)

$$\int g d\boldsymbol{\xi} = \rho \tag{2.2a}$$

$$\int g \xi_\alpha d\boldsymbol{\xi} = \rho u_\alpha \tag{2.2b}$$

$$\int g \xi_\alpha \xi_\beta d\boldsymbol{\xi} = \rho u_\alpha u_\beta + p\delta_{\alpha\beta} \tag{2.2c}$$

$$\int g \xi_\alpha \xi_\beta \xi_\varphi d\boldsymbol{\xi} = p\left(u_\alpha \delta_{\beta\varphi} + u_\beta \delta_{\varphi\alpha} + u_\varphi \delta_{\alpha\beta}\right) + \rho u_\alpha u_\beta u_\varphi \tag{2.2d}$$

Indeed, these relations guarantee that the BGK-Boltzmann equation, Equ. (2.1), can

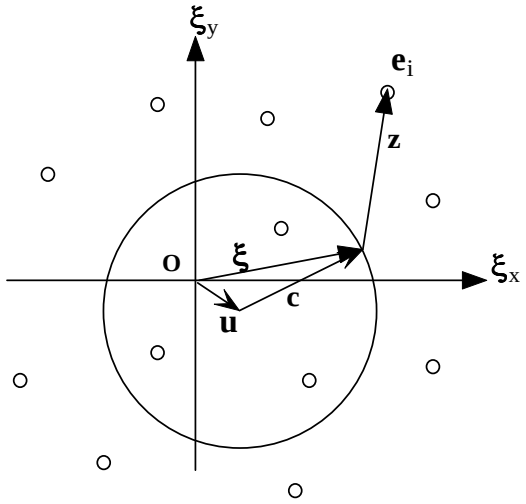

Figure 2.1: The schematic view of the circular function. The small circles are discrete velocities in the velocity space. $\mathbf{e}_i$ is one of them. $\mathbf{u}$ is the mean velocity, $\mathbf{c}$ is the effective peculiar velocity and $\mathbf{z}_i$ is the vector from any point on the circle to $\mathbf{e}_i$.

recover the isothermal incompressible Navier-Stokes equations.

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial \rho u_\alpha}{\partial x_\alpha} &= 0 \\ \frac{\partial \rho u_\alpha}{\partial t} + \frac{\partial \rho u_\alpha u_\beta}{\partial x_\alpha} + \frac{\partial p}{\partial x_\alpha} &= \frac{\partial}{\partial x_\beta}\left[\mu\left(\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha}\right)\right] \end{aligned}$$

where μ is the viscosity $\mu = \rho RT\tau$. Obviously, the simple function that we are looking for, should satisfy these four moment relations.

For 2D problems, we can consider a very simple function as (Fig. 2.1)

$$g = \begin{cases} \dfrac{\rho}{2\pi c} & \text{if } \|\boldsymbol{\xi} - \mathbf{u}\| = c = \sqrt{D\dfrac{p}{\rho}} \\ \\ 0 & \text{else} \end{cases} \tag{2.3}$$

where $D = 2$ is the spatial dimension. This function means that all density and momentum concentrate on a circle with radius of c and center of $\mathbf{u}$.

The velocity on the circle can be written as $\boldsymbol{\xi} = \mathbf{u}+\mathbf{c}$ and the integral in the continuous

velocity space can be reduced to the integral along the circle. Thus, for a small arc ds on the circle, the density $d\rho$ and momentum dP_α are

$$\begin{aligned} d\rho &= \frac{\rho}{2\pi c} ds = \frac{\rho}{2\pi} d\theta \\ dP_\alpha &= \frac{\rho}{2\pi c} \xi ds = \frac{\rho}{2\pi} \left(u_\alpha + c_\alpha\right) d\theta \end{aligned}$$

It is easy to verify that this circular function satisfies the following relations

$$\oint g d\boldsymbol{\xi} = \oint d\rho = \rho \tag{2.4a}$$

$$\oint g \xi_\alpha d\boldsymbol{\xi} = \oint dP_\alpha = \rho u_\alpha \tag{2.4b}$$

$$\oint g \xi_\alpha \xi_\beta d\boldsymbol{\xi} = \oint dP_\alpha \left(\mathbf{u} + \mathbf{c}\right)_\beta = \rho u_\alpha u_\beta + p\delta_{\alpha\beta} \tag{2.4c}$$

$$\begin{aligned} \oint g \xi_\alpha \xi_\beta \xi_\varphi d\boldsymbol{\xi} &= \oint dP_\alpha \left(\mathbf{u} + \mathbf{c}\right)_\beta \left(\mathbf{u} + \mathbf{c}\right)_\varphi \\ &= p\left(u_\alpha \delta_{\beta\varphi} + u_\beta \delta_{\varphi\alpha} + u_\varphi \delta_{\alpha\beta}\right) + \rho u_\alpha u_\beta u_\varphi \end{aligned} \tag{2.4d}$$

where

$$p = \frac{1}{D} \rho c^2$$

These relations are the same as Equ. (2.2). So, this circular function can replace the Maxwellian function to recover the isothermal incompressible NS equations. In this work, we name it Circular Function for Viscous Isothermal Incompressible Flows (CF-VIIF).

2.2 Discretizing CF-VIIF to derive a LB model

It was shown in the last section that CF-VIIF satisfies the constraints (2.2) and the isothermal incompressible Navier-Stokes equations can be recovered. However, CF-VIIF cannot be directly applied in LBM. Although CF-VIIF is greatly simplified as compared to the Maxwellian function, it is still a continuous function, and the integrals in the velocity space are performed along the circle. In the context of LBM, the discrete lattice velocities are given and fixed, and the integrals are replaced by summations over all the

lattice velocity directions. It is expected that the equilibrium distribution function in a lattice model can be obtained by discretizing CF-VIIF onto a lattice in such a way that the constraints (2.2) can be satisfied in the context of LBM when the integrals are replaced by the summations. In this section, we will study in the process what conditions of discretization should be satisfied.

2.2.1 Conditions of discretization

Here, based on CF-VIIF, we try to derive a LB model for incompressible isothermal flows.

Suppose that in the velocity space $\xi_x - \xi_y$, there are N discrete velocities, $\{\mathbf{e}_i, i = 1 \dots N\}$ (Fig. 2.1). CF-VIIF will be discretized to all $\mathbf{e}$. For any $d\rho$ on the circle, it has a contribution $\phi_i(\boldsymbol{\xi})d\rho$ on $\mathbf{e_i}$, where $\phi_i(\boldsymbol{\xi})$ is called assigning function. The contribution of the whole circle to $\mathbf{e}_i$ can be written as

$$\rho_i = \oint \frac{\rho}{2\pi c}\phi_i\left(\boldsymbol{\xi}\right) ds = \frac{\rho}{2\pi}\int_0^{2\pi} \phi_i\left(u + c\cos\theta, v + c\sin\theta\right) d\theta \tag{2.5}$$

ρ_i could be the particle equilibrium distribution function f_i^{eq} in the $\mathbf{e}_i$ direction, f_i^{eq}.

In context of LBM, Equ. (2.4) are replaced by summations

$$\sum_{i=1}^{N} \rho_i = \oint d\rho \tag{2.6a}$$

$$\sum_{i=1}^{N} \rho_i e_{i\alpha} = \oint dP_\alpha \tag{2.6b}$$

$$\sum_{i=1}^{N} \rho_i e_{i\alpha} e_{i\beta} = \oint \left(dP_\alpha\right)\left(\mathbf{u} + \mathbf{c}\right)_\beta \tag{2.6c}$$

$$\sum_{i=1}^{N} \rho_i e_{i\alpha} e_{i\beta} e_{i\chi} = \oint \left(dP_\alpha\right)\left(\mathbf{u} + \mathbf{c}\right)_\beta \left(\mathbf{u} + \mathbf{c}\right)_\chi \tag{2.6d}$$

As shown in Fig. 2.1, the relationship between the discrete velocity $\mathbf{e}_i$ and the original velocity $\boldsymbol{\xi}$ is

$$\mathbf{e}_i = \boldsymbol{\xi} + \mathbf{z}_i(\boldsymbol{\xi}) \tag{2.7}$$

Substituting Equ. (2.5) and Equ. (2.7) into Equ. (2.6a) gives

$$\sum_{i=1}^{N} \rho_i = \sum_{i=1}^{N} \oint \frac{\rho}{2\pi c} \phi_i \left(\boldsymbol{\xi}\right) ds \equiv \oint \frac{\rho}{2\pi c} ds$$

Thus

$$\sum_{i=1}^{N} \phi_i \left(\boldsymbol{\xi}\right) = 1 \tag{2.8}$$

is a sufficient condition for equation Equ. (2.6a). Then substituting Equ. (2.5), Equ. (2.7) and (2.8) into (2.6b)

$$\sum_{i=1}^{N} \rho_i e_{i\alpha} = \oint \frac{\rho}{2\pi c} \boldsymbol{\xi} ds + \oint \frac{\rho}{2\pi c} \sum_{i=1}^{N} \mathbf{z}_i(\boldsymbol{\xi})_\alpha \phi_i \left(\boldsymbol{\xi}\right) ds \equiv \oint \frac{\rho}{2\pi c} \boldsymbol{\xi} ds$$

Similarly, we can give a sufficient condition of Equ. (2.6b)

$$\sum_{i=1}^{N} \phi_i \left(\boldsymbol{\xi}\right) \mathbf{z}_i(\boldsymbol{\xi}) = 0 \tag{2.9}$$

Furthermore, substituting Equ. (2.5) and Equ. (2.7-2.9) into (2.6c) leads to

$$\sum_{i=1}^{N} \rho_i e_{i\alpha} e_{i\beta} = \oint \frac{\rho}{2\pi c} \xi_\alpha \, \xi_\beta \, ds + \frac{\rho}{2\pi c} \oint \sum_{i=1}^{N} \mathbf{z}_i(\boldsymbol{\xi})_\alpha \mathbf{z}_i(\boldsymbol{\xi})_\beta \phi_i \left(\boldsymbol{\xi}\right) ds \equiv \oint \frac{\rho}{2\pi c} \xi_\alpha \, \xi_\beta \, ds$$

which can give

$$\sum_{i=1}^{N} \phi_i \left(\boldsymbol{\xi}\right) \mathbf{z}_i(\boldsymbol{\xi}) \mathbf{z}_i(\boldsymbol{\xi}) = 0 \tag{2.10}$$

Finally, substituting Equ. (2.5), Equ. (2.7-2.10) into (2.6d) gives

$$\sum_{i=1}^{N} \phi_i \left(\boldsymbol{\xi}\right) \mathbf{z}_i(\xi) \mathbf{z}_i(\boldsymbol{\xi}) \mathbf{z}_i(\boldsymbol{\xi}) = 0 \tag{2.11}$$

Equs (2.8-2.11) are the constraints for assigning function $\phi_i(\boldsymbol{\xi})$. Once we find $\phi_i(\boldsymbol{\xi})$, f_i^{eq} can be determined from Equ. (2.5).

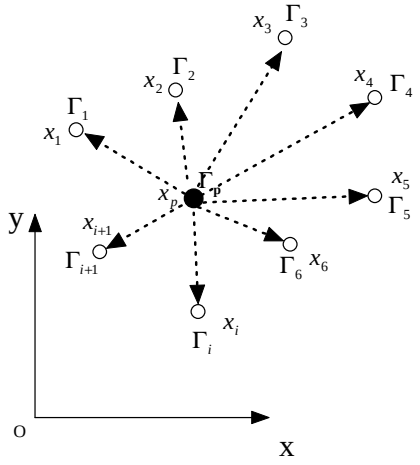

Figure 2.2: Schematic view of assigning a particle Γ_p at $\mathbf{x}_p$ onto several other points $\mathbf{x}_i$.

2.2.2 Constructing assignment functions

The use of assigning functions to assign a variable from one point to several other points is widely used in particle methods, such as particle in cell (PIC) method, vortex method (VM) and vortex in cell (VIC) method. It was found [34] that assigning a particle Γ_p at $\mathbf{x}_p$ with a self-contained set of n-th order Lagrangian polynomials onto $n+1$ discrete points $\mathbf{x}_i, i = 0 \ldots n$ (Fig. 2.2) can conserve up to n-th moment of the particle. That is

$$\Gamma_p \mathbf{x}^j = \sum_{i=0}^{n} \Gamma_i \mathbf{x}_i^j \quad j = 0...n \tag{2.12}$$

where $\Gamma_i = \Gamma_p \phi_i(\mathbf{x}_p)$ is the part assigned to point $\mathbf{x}_i$. This means that

$$\sum_{i=0}^{n} \Gamma_i (\mathbf{x} - \mathbf{x}_p)^j = 0 \quad j = 0...n \tag{2.13}$$

So Equ. (2.8-2.11) can be satisfied, when a self-contained set of third order Lagrangian polynomials is taken as the assigning function. And the interpolating stencil made of those points in the velocity space serves as the velocity vectors of a LB model. Thus, if we find the set of polynomials and its stencil, the lattice is formed and determination of f_i^{eq} is trivial.

The actual form of polynomials depends on the configuration of the lattice in the velocity space. According to the constraints shown in Section 2.2.1, we need to construct a third order two-dimensional polynomial. In the two-dimensional space $x-y$, a general form of the third order polynomial can be written as

$$P(x,y) = a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 + a_7x^3 + a_8x^2y + a_9xy^2 + a_{10}y^3 \quad (2.14)$$

where $a_{10} \neq 0$. To determine the 10 coefficients $a_i, i = 1 \ldots 10$, Equ. (2.14) has to be collocated at 10 points. A 10-node triangular element could be a choice. However, the velocity vectors determined from these 10 points may be difficult to perform the collision-streaming process. To remove this difficulty, we try to use a square element. The base functions of 9-node square element whose shape is the same as D2Q9 lattice, contain the terms $\left\{1, x, y, x^2, xy, y^2, x^2y, xy^2, x^2y^2\right\}$, lack of x^3 and y^3 . As compared with D2Q9 lattice, D2Q13 lattice has 4 additional velocities, $(0,2)$, $(0,2)$, $(-2,0)$ and $(0,-2)$. Consequently, the base functions of the 13-node element based on D2Q13 lattice have 3 more terms, $a_{11}x^4 + a_{12}x^2y^2 + a_{13}y^4$, than Equ. (2.14). The polynomial for the 13 points can be written as

$$\begin{aligned} \phi_i(x,y) &= a_{i,1} + a_{i,2}x + a_{i,3}y + a_{i,4}x^2 + a_{i,5}xy + a_{i,6}y^2 + a_{i,7}x^3 + a_{i,8}x^2y \\ &\quad + a_{i,9}xy^2 + a_{i,10}y^3 + a_{i,11}x^4 + a_{i,12}x^2y^2 + a_{i,13}y^4 \end{aligned} \quad (2.15)$$

which contains all the terms of Equ. (2.14). So, this 13-node element can be used. All the base functions of the 13-node element, $\boldsymbol{\phi}(x,y)$, can be written as the

$$\boldsymbol{\phi}(x,y) = \mathbf{At} \quad (2.16)$$

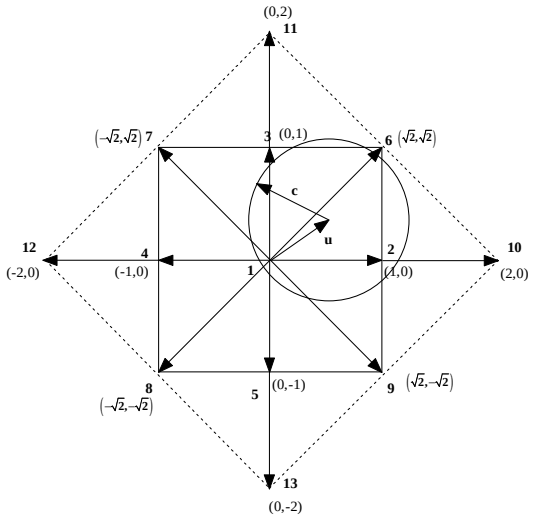

Figure 2.3: The scheme of D2Q13 lattice..

where

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,13} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,13} \\ \vdots & \vdots & \ddots & \vdots \\ a_{13,1} & a_{13,2} & \cdots & a_{13,13} \end{pmatrix}$$

$$\mathbf{t} = \left[1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3, x^4, x^2y^2, y^4\right]^T$$

Since $\phi_i(x, y)$ is a Lagrangian interpolation polynomial, it should satisfy

$$\begin{aligned} \phi_i\left(x_j, y_j\right) &= \delta_{ij} \qquad i, j = 1, 2, ..., 13 \\ &\Rightarrow \mathbf{AT} = \mathbf{I} \end{aligned} \tag{2.17}$$

where (x_j, y_j) is the location of node j of the element and

$$\mathbf{T} = \left[\mathbf{t}\left(x_1, y_1\right), \mathbf{t}\left(x_2, y_2\right), ...\mathbf{t}\left(x_{13}, y_{13}\right)\right]$$

So the coefficients can be computed from a linear system

$$\mathbf{A} = \mathbf{T}^{-1} \tag{2.18}$$

By substituting Equ. (2.18) into Equ. (2.16), we can obtain all the base functions as

$$\begin{aligned}
\phi_1(x,y) &= 1 - \frac{5}{4}x^2 - \frac{5}{4}y^2 + \frac{1}{4}x^4 + x^2y^2 + \frac{1}{4}y^4 && \text{(2.19a)}\\
\phi_2(x,y) &= \frac{2}{3}x + \frac{2}{3}x^2 - \frac{1}{6}x^3 - \frac{1}{2}xy^2 - \frac{1}{6}x^4 - \frac{1}{2}x^2y^2 && \text{(2.19b)}\\
\phi_3(x,y) &= \frac{2}{3}y + \frac{2}{3}y^2 - \frac{1}{2}x^2y - \frac{1}{6}y^3 - \frac{1}{2}x^2y^2 - \frac{1}{6}y^4 && \text{(2.19c)}\\
\phi_4(x,y) &= -\frac{2}{3}x + \frac{2}{3}x^2 + \frac{1}{6}x^3 + \frac{1}{2}xy^2 - \frac{1}{6}x^4 - \frac{1}{2}x^2y^2 && \text{(2.19d)}\\
\phi_5(x,y) &= -\frac{2}{3}y + \frac{2}{3}y^2 + \frac{1}{2}x^2y + \frac{1}{6}y^3 - \frac{1}{2}x^2y^2 - \frac{1}{6}y^4 && \text{(2.19e)}\\
\phi_6(x,y) &= \frac{1}{4}yx + \frac{1}{4}x^2y + \frac{1}{4}xy^2 + \frac{1}{4}x^2y^2 && \text{(2.19f)}\\
\phi_7(x,y) &= -\frac{1}{4}yx + \frac{1}{4}x^2y - \frac{1}{4}xy^2 + \frac{1}{4}x^2y^2 && \text{(2.19g)}\\
\phi_8(x,y) &= \frac{1}{4}yx - \frac{1}{4}x^2y - \frac{1}{4}xy^2 + \frac{1}{4}x^2y^2 && \text{(2.19h)}\\
\phi_9(x,y) &= -\frac{1}{4}yx - \frac{1}{4}x^2y + \frac{1}{4}xy^2 + \frac{1}{4}x^2y^2 && \text{(2.19i)}\\
\phi_{10}(x,y) &= -\frac{1}{12}x - \frac{1}{24}x^2 + \frac{1}{12}x^3 + \frac{1}{24}x^4 && \text{(2.19j)}\\
\phi_{11}(x,y) &= -\frac{1}{12}y - \frac{1}{24}y^2 + \frac{1}{12}y^3 + \frac{1}{24}y^4 && \text{(2.19k)}\\
\phi_{12}(x,y) &= \frac{1}{12}x - \frac{1}{24}x^2 - \frac{1}{12}x^3 + \frac{1}{24}x^4 && \text{(2.19l)}\\
\phi_{13}(x,y) &= \frac{1}{12}y - \frac{1}{24}y^2 - \frac{1}{12}y^3 + \frac{1}{24}y^4 && \text{(2.19m)}
\end{aligned}$$

By substituting above equations into Equ. (2.5), the integral can be evaluated analytically to get f_i^{eq} since $\phi_i(x,y)$ is a polynomial

$$
\begin{cases}
f_1^{eq} = \dfrac{1}{16}\rho(4u^4 + 5c^4 - 20c^2 + 16 + 16u^2v^2 + 4v^4 - 20v^2 + 20c^2v^2 + 20c^2u^2 - 20u^2) \\
f_2^{eq} = -\dfrac{1}{24}\rho\left(-16u^2 + 3c^4 + 4u^4 - 8c^2 + 6c^2v^2 + 12uv^2 + 18c^2u^2 + 12c^2u + 12u^2v^2 - 16u + 4u^3\right) \\
f_3^{eq} = -\dfrac{1}{24}\rho\left(12u^2v^2 + 6c^2u^2 + 18c^2v^2 + 12c^2v + 12u^2v + 3c^4 - 8c^2 + 4v^4 - 16v^2 + 4v^3 - 16v\right) \\
f_4^{eq} = -\dfrac{1}{24}\rho\left(-16u^2 + 3c^4 + 4u^4 - 8c^2 + 6c^2v^2 - 12uv^2 + 18c^2u^2 - 12c^2u + 12u^2v^2 + 16u - 4u^3\right) \\
f_5^{eq} = -\dfrac{1}{24}\rho\left(-4v^3 - 12c^2v - 12u^2v + 3c^4 - 8c^2 + 4v^4 - 16v^2 + 16v + 12u^2v^2 + 18c^2v^2 + 6c^2u^2\right) \\
f_6^{eq} = \dfrac{1}{32}\rho\left(8uv^2 + 8vu + 4c^2u + 4c^2v + 8u^2v + c^4 + 8u^2v^2 + 4c^2v^2 + 4c^2u^2\right) \\
f_7^{eq} = \dfrac{1}{32}\rho\left(c^4 + 4c^2v - 8uv^2 - 8vu + 4c^2u^2 - 4c^2u + 4c^2v^2 + 8u^2v^2 + 8u^2v\right) \\
f_8^{eq} = \dfrac{1}{32}\rho\left(c^4 - 4c^2v - 8uv^2 + 8vu + 4c^2u^2 - 4c^2u + 4c^2v^2 + 8u^2v^2 - 8u^2v\right) \\
f_9^{eq} = \dfrac{1}{32}\rho\left(c^4 - 4c^2v + 8uv^2 - 8vu + 4c^2u^2 + 4c^2u + 4c^2v^2 + 8u^2v^2 - 8u^2v\right) \\
f_{10}^{eq} = -\dfrac{1}{192}\rho\left(16u + 8u^4 - 4c^2 - 8u^2 + 24c^2u^2 + 3c^4 + 16u^3 + 24c^2u\right) \\
f_{11}^{eq} = \dfrac{1}{192}\rho\left(-16v + 24c^2v - 8v^2 + 8v^4 - 4c^2 + 24c^2v^2 + 16v^3 + 3c^4\right) \\
f_{12}^{eq} = \dfrac{1}{192}\rho\left(16u + 8u^4 - 4c^2 - 8u^2 + 24c^2u^2 + 3c^4 - 16u^3 - 24c^2u\right) \\
f_{13}^{eq} = \dfrac{1}{192}\rho\left(16v - 24c^2v - 8v^2 + 8v^4 - 4c^2 + 24c^2v^2 - 16v^3 + 3c^4\right)
\end{cases}
\tag{2.20}
$$

2.3 Chapman-Enskog analysis

In the last section, we have derived a D2Q13 model. Its f_i^{eq} satisfies

$$\begin{cases} \sum\limits_i f_i^{eq} = \rho \\ \sum\limits_i f_i^{eq} e_{i\alpha} = \rho u_\alpha \\ \sum\limits_i f_i^{eq} e_{i\alpha} e_{i\beta} = \rho u_\alpha u_\beta + p\delta_{\alpha\beta} \\ \sum\limits_i f_i^{eq} e_{i\alpha} e_{i\beta} e_{i\varphi} = \rho u_\alpha u_\beta u_\varphi + p\left(u_\alpha \delta_{\beta\varphi} + u_\alpha \delta_{\beta\varphi} + u_\alpha \delta_{\beta\varphi}\right) \end{cases} \tag{2.21}$$

With this model, the following DVBE

$$\partial_t f_i + e_{i\alpha}\partial_\alpha f_i = \left(f_i^{eq} - f_i\right)/\tau \tag{2.22}$$

can recover the isothermal incompressible NS equations by means of Chapman-Enskog analysis.

At first, the following expansions are introduced

$$\begin{aligned} f_i &= f_i^{(0)} + \varepsilon f_i^{(1)} \\ \partial t &= \partial_{t_0} + \varepsilon \partial_{t_1} \\ \partial_\alpha &= \partial_{\alpha 1} \end{aligned}$$

Substituting above equations into Equ. (2.22) gives

$$\left(\partial_{t_0} f_i^{(0)} + e_{i\alpha}\partial_{\alpha 1} f_i^{(0)} + f_i^{(1)}/\tau\right) + \varepsilon\left(\partial_{t_1} f_i^{(0)} + \partial_{t_0} f_i^{(1)} + e_{i\alpha}\partial_{\alpha 1} f_i^{(1)}\right) + O\left(\varepsilon^2\right) = 0$$

Since ε is an arbitrary small number, we have

$$\partial_{t_0} f_i^{(0)} + e_{i\alpha}\partial_{\alpha 1} f_i^{(0)} + f_i^{(1)}/\tau = 0 \tag{2.23a}$$

$$\partial_{t_1} f_i^{(0)} + \partial_{t_0} f_i^{(1)} + e_{i\alpha}\partial_{\alpha 1} f_i^{(1)} = 0 \tag{2.23b}$$

From Equ. (2.23a), we can get macroscopic equations in the time scale t_0

$$\begin{aligned}
&\sum_i \left(\partial_{t_0} f_i^{(0)} + e_{i\alpha}\partial_{\alpha 1} f_i^{(0)} + f_i^{(1)}/\tau\right)\begin{bmatrix} 1 \\ e_{i\beta} \end{bmatrix} = 0 \\
&\Rightarrow \begin{cases} \partial_{t_0}\rho + \partial_{\alpha 1}\left(\rho u_\alpha\right) = 0 \\ \partial_{t_0}\left(\rho u_\alpha\right) + \partial_{\alpha 1}\left(\rho u_\alpha u_\beta + p\delta_{\alpha\beta}\right) = 0 \end{cases}
\end{aligned} \tag{2.24}$$

From Equ. (2.23b), we can get macroscopic equations in the time scale t_1

$$\begin{aligned}
&\sum_i \left(\partial_{t_1} f_i^{(0)} + \partial_{t_0} f_i^{(1)} + e_{i\alpha}\partial_{\alpha 1} f_i^{(1)}\right)\begin{bmatrix} 1 \\ e_{i\beta} \end{bmatrix} = 0 \\
&\Rightarrow \begin{cases} \partial_{t_1}\rho = 0 \\ \partial_{t_1}\left(\rho u_\alpha\right) + \partial_{\alpha 1}\sum_i f_i^{(1)} e_{i\alpha} e_{i\beta} = 0 \end{cases}
\end{aligned} \tag{2.25}$$

Here

$$\begin{aligned}
\sum_i f_i^{(1)} e_{i\alpha} e_{i\beta} &= -\tau \sum_i \left(\partial_{t_0} f_i^{(0)} + e_{i\chi}\partial_{\chi 1} f_i^{(0)}\right) e_{i\alpha} e_{i\beta} \\
&= -\tau \sum_i \left[\partial_{t_0} f_i^{(0)} e_{i\alpha} e_{i\beta} + \partial_{\chi 1}\left(f_i^{(0)} e_{i\alpha} e_{i\beta} e_{i\chi}\right)\right]
\end{aligned} \tag{2.26}$$

The first term of Equ. (2.26) can be expanded as

$$\begin{aligned}
\sum_i \left(\partial_{t_0} f_i^{(0)} e_{i\alpha} e_{i\beta}\right) &= \partial_{t_0}\left[\rho u_\alpha u_\beta + \frac{1}{2}c^2\rho\delta_{\alpha\beta}\right] \\
&= -\frac{1}{2}c^2\rho\delta_{\alpha\beta}\partial_{\chi 1} u_\chi - \frac{1}{2}c^2\left(u_\chi\delta_{\alpha\beta}\partial_{\chi 1}\rho + u_{\beta\chi}\partial_{\alpha 1}\rho + u_\alpha\partial_{\beta 1}\rho\right) \\
&\quad -\partial_{\chi 1}\left(\rho u_\alpha u_\beta u_\chi\right)
\end{aligned}$$

and the second term of Equ. (2.26) can be written as

$$\begin{aligned}
&\sum_i \partial_{\chi 1}\left(f_i^{(0)} e_{i\alpha} e_{i\beta} e_{i\chi}\right) \\
&= \partial_{\chi 1}\left[\frac{1}{2}\rho c^2\left(u_\alpha \delta_{\beta\chi} + u_\beta \delta_{\chi\alpha} + u_\chi \delta_{\alpha\beta}\right) + \rho u_\alpha u_\beta u_\chi\right] \\
&= \frac{1}{2}\rho c^2\left(\delta_{\alpha\beta}\partial_{\chi 1} u_\chi + \partial_{\alpha 1} u_{\beta\chi} + \partial_{\beta 1} u_\alpha\right) \\
&+ \frac{1}{2}c^2\left(u_\chi \delta_{\alpha\beta}\partial_{\chi 1}\rho + u_{\beta\chi}\partial_{\alpha 1}\rho + u_\alpha \partial_{\beta 1}\rho\right) \\
&+ \partial_{\chi 1}\left(\rho u_\alpha u_\beta u_\chi\right)
\end{aligned}$$

So, we get

$$\begin{aligned}
\partial_{\beta 1}\sum_i f_i^{(1)} e_{i\alpha} e_{i\beta} &= \partial_{\beta 1}\left[\frac{1}{2}\rho c^2\left(\partial_{\alpha 1} u_{\beta\chi} + \partial_{\beta 1} u_\alpha\right)\right] \\
&= -\nabla_1 \cdot \left[p\tau\left(\nabla_1 \mathbf{v} + \nabla_1^T \mathbf{v}\right)\right]
\end{aligned}$$

Thus the macroscopic equations in the time scale t_1 can be obtained as

$$\begin{cases}
\partial_{t_1}\rho = 0 \\
\partial_{t_1}\left(\rho u_\alpha\right) = \nabla_1 \cdot \left[p\tau\left(\nabla_1 \mathbf{v} + \nabla_1^T \mathbf{v}\right)\right]
\end{cases}$$

Combining the equations of the two time scales, we can get the macroscopic equations as

$$\begin{cases}
\dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \\
\dfrac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v} + p\mathbf{I}) = \nabla \cdot \left[\mu\left(\nabla \mathbf{v} + \nabla^T \mathbf{v}\right)\right]
\end{cases} \tag{2.27}$$

where

$$\mu = p\tau$$

In case of a small Mach number, ρ varies little and can be regarded as constant. Then, Equ. (2.27) can be reduced to isothermal incompressible NS equations

$$\begin{cases}
\nabla \cdot \mathbf{v} = 0 \\
\dfrac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\nabla \mathbf{v} = -\dfrac{1}{\rho}\nabla p + \nabla \cdot \left[\nu\left(\nabla \mathbf{v} + \nabla^T \mathbf{v}\right)\right]
\end{cases}$$

It should be pointed out that the form of f_i^{eq} is not involved in the above derivation

process. This D2Q13 model is just a discrete form of CF-VIIF which satisfies Equs. (2.4a -2.4d) and recovers the viscous isothermal NS equations. Another point we should indicate is that f_i^{eq} can recover Equ. (2.22) without the assumption of small Mach number and the isothermal condition (c = const) because of the simple form of CF-VIIF. This is a major advantage of our derivation method. As shown in the next chapter, we will further extend this method to develop lattice Boltzmann models for compressible flows with shock waves.

2.4 Numerical tests

To validate this model, we simulated the lid-driven cavity flow. This is an isothermal incompressible flow problem. The following conditions should be satisfied

$$\|\mathbf{u}\| << c = \text{const} \tag{2.28a}$$

$$\|\mathbf{u}\| + c < \sqrt{2} \tag{2.28b}$$

Equ. (2.28a) means the small Mach number and isothermal conditions, while Equ. (2.28b) makes sure that the circle is inside the interpolation stencil to avoid extrapolation.

2.4.1 Simulating the lid-driven cavity flow with collision-streaming procedure

First, the traditional collision-streaming(CS) process was used. Because of the numerical viscosity of CS, the viscosity should be modified as

$$\nu = p\left(\tau - \frac{1}{2}\right)\rho = \frac{1}{2}c^2\left(\tau - \frac{1}{2}\right)$$

The velocity of the top lid is set as 0.08 and $c = 0.8$ to satisfy Equ. (2.28). The initial density is set as 1. On the top boundary, f_i is always set as the equilibrium state f_i^{eq}. On the other walls, the bounce-back condition is used.

For the case of Re = 1000, three vortices were computed. And the locations of the

		Primary		Lower left		Lower right	
		x/L_{lid}	y/L_{lid}	x/L_{lid}	y/L_{lid}	x/L_{lid}	y/L_{lid}
$Re = 1000$	Ghia	0.5313	0.5625	0.0859	0.0781	0.8594	0.1094
	Hou	0.5333	0.5647	0.0902	0.0784	0.8667	0.1137
	Present	0.5301	0.5639	0.0832	0.0774	0.8625	0.1130
$Re = 5000$	Ghia	0.5117	0.5352	0.0703	0.1367	0.8086	0.0742
	Hou	0.5176	0.5373	0.0784	0.1373	0.8078	0.0745
	Present	0.5159	0.5360	0.0721	0.1425	0.8059	0.0724

Table 2.1: Locations of vortices for the lid-driven cavity flow of $Re = 1000$ and 5000

vortices, as well as previous results by Ghia *et al.* [35] (with FDM NS solver) and Hou *et al.* [6] (with D2Q9 LBM), are listed in Tab. 2.1. Our results agree with the referenced data excellently. Fig. 2.4 presents the streamlines and normalized velocity profiles along the two centerlines of the box. The solid symbols are the results of Ghia *et al.* [35]. Good agreement between our results and the benchmark solutions can be observed. On the other hand, it should be indicated that the stability of collision-streaming process of this model is not good. A 401×401 uniform grid is needed in the simulation to get a smooth pressure field. It might be due to the non-smooth CF-VIIF and existence of negative f_i^{eq} resulted from the third order Lagrangian interpolation. Nevertheless, the results agree excellently well with previous studies. This implies that our deriving method and model are correct and feasible. In order to increase the stability of the model, a smooth function and an optimized interpolation stencil might be prefered. In simulation, finite difference method might be also a feasible way since finite difference schemes can introduce numerical dissipation to stabilize the computation.

2.4.2 Simulating the lid-driven cavity flow with finite difference method

In order to investigate the capability of this model to simulate incompressible flows at higher Reynolds numbers, we will use the finite difference method to solve DVBE for simulation of the problem at $Re = 5000$. The smaller CFL number and numerical dissipation of the finite difference scheme might stabilize the simulation.

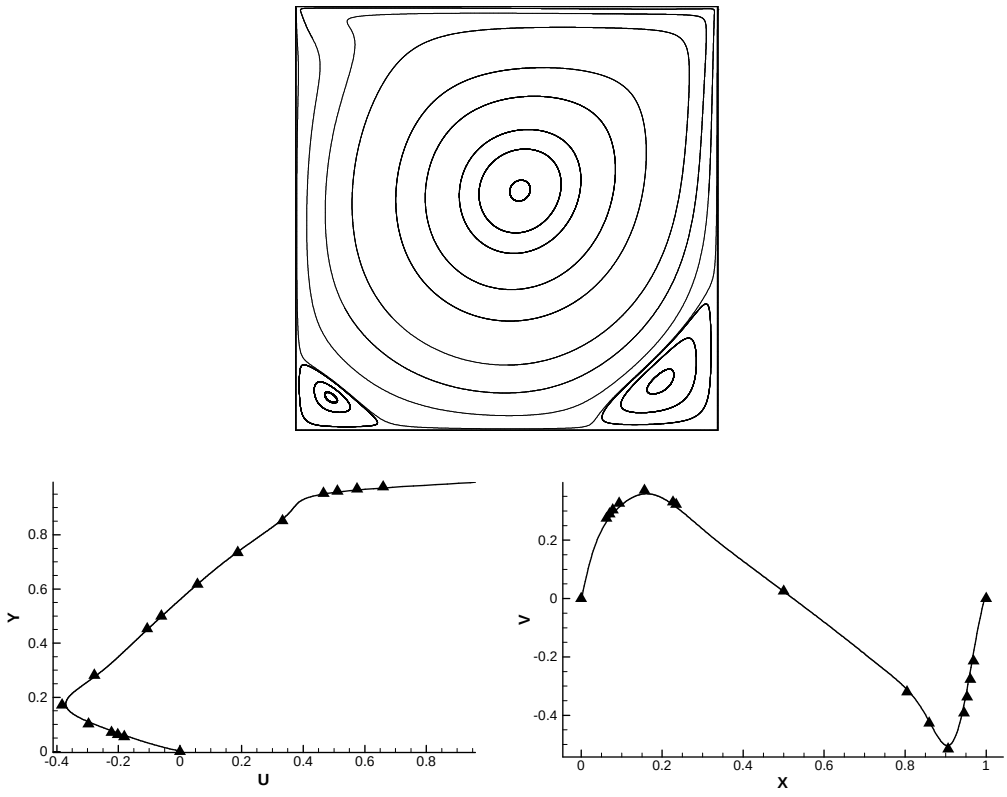

Figure 2.4: Streamlines and velocity profiles along the two central line of the lid-driven cavity flow of $Re = 1000$ (computed with streaming-collision procedure).

The explicit finite difference LBM for curvilinear coordinates developed by Guo and Zhao [36] was applied here to simulate the lid-driven cavity flow of $Re = 5000$. In this FDM LBM, DVBE is first integrated to give

$$f_i^{n+1} - f_i^n + \Delta t \mathbf{e}_i \cdot \nabla f_i^n = \Delta t \left[\theta \Omega_i^{n+1} + (1-\theta)\,\Omega_i^n\right] \tag{2.29}$$

where $t^{n+1} = t^n + \Delta t$, Ω is the collision operator and $0 \leq \theta \leq 1$. Here, $\theta = 0$ or 1 means that the collision term is treated completely explicit or implicit, respectively. To achieve the second-order approximation, θ is set as $\frac{1}{2}$. But, with this setting, Ω^{n+1} needs be computed implicitly. To avoid it, they introduced the following function

$$h_i = f_i + \omega\theta(f_i - f_i^{eq}) \tag{2.30}$$

where $\omega = \Delta t/\tau$. Then the following semi-discretized DVBE can be obtained

$$h_i^{n+1} + \Delta t \mathbf{e}_i \cdot \nabla f_i^n = (1 - \omega + \omega\theta) f_i^n + \omega(1-\theta) f_i^{eq,n} \tag{2.31}$$

where

$$f_i^n = \frac{1}{1+\omega\theta}(h_i^n + \omega\theta f_i^{eq,n})$$

The macroscopic density and velocity can be computed from h_i directly as

$$\rho = \sum h_i \qquad \rho\mathbf{v} = \sum h_i \mathbf{e}_i \tag{2.32}$$

In a general curvilinear coordinate $\xi - \eta$, the advection term is written as

$$\mathbf{e}_i \cdot \nabla f_i = e_{i\alpha} \frac{\partial f_i}{\partial x_\alpha} = c_{i\beta} \frac{\partial f_i}{\partial \xi_\beta} \tag{2.33}$$

where $c_{i\beta} = e_{i\alpha} \dfrac{\partial \xi_\beta}{\partial x_\alpha}$. Numerical discretization can be made by combining the second

order upwind scheme and the central scheme

$$\frac{\partial f_i}{\partial \xi_\beta} = \sigma S_i^u + (1-\sigma) S_i^c$$

where S_i^u and S_i^c are the upwind and central scheme, respectively, $0 \leq \sigma \leq 1$ is a control parameter to adjust the weight of the two schemes. In our simulation, σ is set as 0.05.

Non-equilibrium extrapolation method [37] is applied to implement the non-slip wall boundary conditions. In this method, the distribution function $f_i(\mathbf{x}_b)$ on the wall node $\mathbf{x}_b$ is determined from $f_i(\mathbf{x}_f)$ at $\mathbf{x}_f$ which is the nearest neighboring fluid node of $\mathbf{x}_b$

$$f_i(\mathbf{x}_b) = f_i^{eq}(\mathbf{x}_b) + [f_i(\mathbf{x}_f) - f_i^{eq}(\mathbf{x}_f)] \tag{2.34}$$

Here $f_i^{eq}(\mathbf{x}_b)$ is computed from mean flow variables (ρ_b and $\mathbf{v}_b$). $\mathbf{v}_b$ is known and ρ_b can be approximated with ρ_f which is the density at $\mathbf{x}_f$.

A 101×101 nonuniform grid was used for simulation. The locations of vortices, as well as previous results by Ghia et al.[35] and Hou *et al.* [6], are listed in Tab. 2.1. Fig. 2.5 presents the streamlines and velocity profiles along the two centerlines where the solid symbols are the results of Ghia *et al.*. The two sets of results agree very well. Comparing with the same simulation by Guo and Zhao [36] who used D2Q9 model, a smaller time step had to be used in our simulation. It suggests that the stability of this new model is not good.

2.5 Concluding remarks

In this chapter, we systematically present a new methodology to construct lattice velocity models and associated equilibrium distribution functions. In particular, the D2Q13 model was developed to simulate viscous isothermal incompressible flows. The high accuracy of the model was verified by its application to simulate lid-driven cavity flows.

The new methodology may provide some interesting features for development of new lattice models. First, it has been shown that, there is no need to assume a formula be-

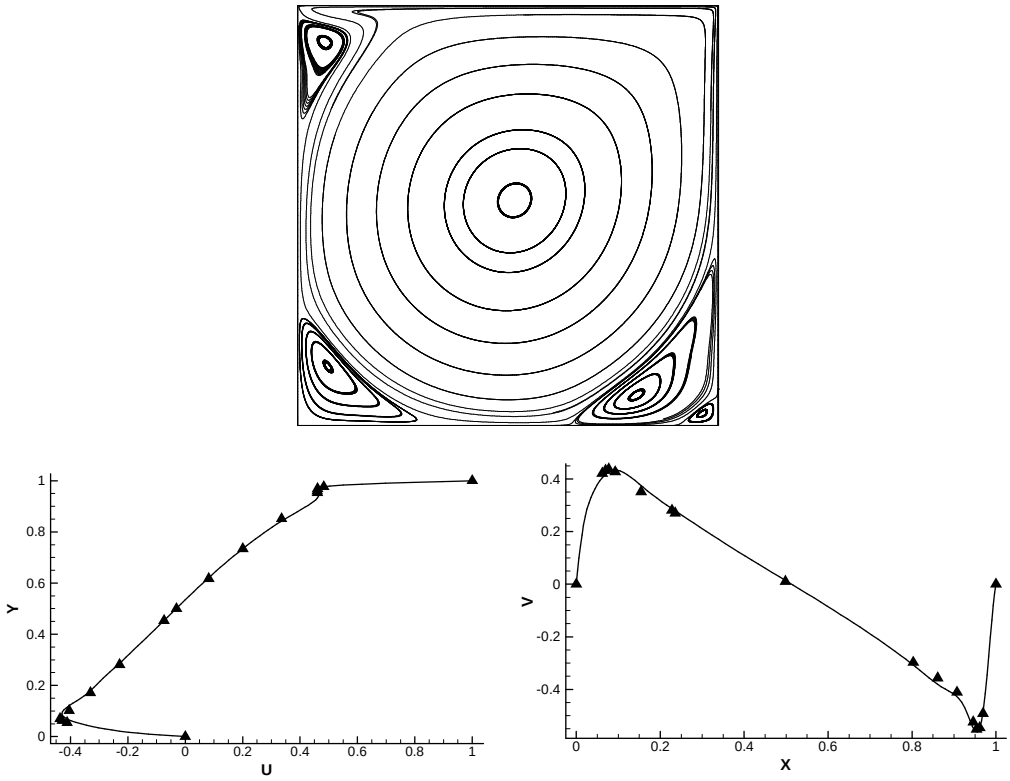

Figure 2.5: Streamlines and velocity profiles along the two central lines of the lid-driven cavity flow of $Re = 5000$ (computed with FDM).

forehand since it can be derived step by step naturally. Also the configuration of the lattice can be obtained with some basic knowledge on polynomial and linear theory. It is expected that with a spherical function and a 3D Lagrange interpolation stencil, the three dimensional LB models can be developed using the same way. The present method is more natural and straightforward than the undetermined coefficient method. Second, we do not need the small Mach number assumption and isothermal assumption as mathematical requirements during the derivation. Although the D2Q13 is derived for isothermal incompressible flows, the integral (2.5) can be evaluated without these assumptions. Only in the phase of simulation, they are imposed as Equs. (2.28a and 2.28b). This means that our model can be derived without these two assumptions. The present approach is different from the traditional deriving methods in which the assumptions of low Mach number and isothermal flows are indispensable. Therefore, our deriving method has fewer limits. LB models for compressible flows might be derived from a proper simplified equilibrium function. Third, it is interesting to note that using a symmetric stencil here is just for convenience of applying the collision-streaming procedure and bounce-back wall boundary condition but not necessary for recovering Navier-Stokes equations. Since FDM, FVM and FEM can handle non-uniform gird and the wall boundary conditions with implementation of non-equilibrium extrapolation method, LB models based on an asymmetric lattice might be feasible. Then, neither of the Maxwellian equilibrium distribution function nor the symmetric lattice is necessary.

In the next chapter, we will discuss how to apply the newly derived method to construct LB models for compressible flows.

Chapter 3

Development of LB Models for Inviscid Compressible Flows

In this chapter, the newly derived method proposed in the last chapter is extended to derive LB models for inviscid compressible flows. The key problem is how to find a simple function which can recover to Euler equations. And after the models are constructed, how to choose a proper numerical method to capture discontinuities in compressible flows of a high Mach number is another challenging issue. This chapter will address these issues and implementation of boundary conditions in details.

3.1 Looking for a simple equilibrium distribution function for inviscid compressible flows

In compressible flows, energy evolution couples with momentum equation and kinetic energy can be converted to internal energy, and vice versa. Thus energy equation should be introduced. For inviscid compressible flows, the governing equations of inviscid com-

pressible flows are Euler equations written as

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_\alpha}{\partial x_\alpha} = 0 \tag{3.1a}$$

$$\frac{\partial \rho u_\alpha}{\partial t} + \frac{\partial (\rho u_\alpha u_\beta + p\delta_{\alpha\beta})}{\partial x_\beta} = 0 \tag{3.1b}$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial \left(\rho E + p\right) u_\alpha}{\partial x_\alpha} = 0 \tag{3.1c}$$

where

$$\begin{aligned} p &= (\gamma - 1)\rho e \\ E &= \frac{1}{2}u^2 + e \\ e &= C_v T \end{aligned}$$

We have argued that in Chapter 1, the Maxwellian, Equ. (1.7) can not be used to derive LB models for compressible flows because it dose not consider rotational degrees of freedom of diatomic or polyatomic molecules. Our work is based on the modified two-dimensional Maxwellian with extra degrees of freedom, Equ. (1.14).

$$g\left(\boldsymbol{\xi}\right) = \frac{\rho}{\left(2\pi RT\right)^{\frac{K+D}{2}}} \exp\left[-\frac{\left(\boldsymbol{\xi} - \mathbf{u}\right)^2 + \sum_{i=1}^{K} \xi_i^2}{2RT}\right]$$

It satisfies the following relations

$$\int g d\boldsymbol{\xi} = \rho \tag{3.2a}$$

$$\int g \xi_\alpha d\boldsymbol{\xi} = \rho u_\alpha \tag{3.2b}$$

$$\int \frac{1}{2} g \xi_\alpha^2 d\boldsymbol{\xi} = \rho \left(\frac{1}{2}u_\alpha^2 + \frac{b}{2}RT\right) \tag{3.2c}$$

$$\int g \xi_\alpha \xi_\beta d\boldsymbol{\xi} = \rho \left(u_\alpha u_\beta + RT\delta_{\alpha\beta}\right) \tag{3.2d}$$

$$\int \frac{1}{2} g \xi_\alpha^2 \xi_\beta d\boldsymbol{\xi} = \rho \left(\frac{1}{2}u_\alpha^2 + \frac{b+2}{2}RT\right) u_\beta \tag{3.2e}$$

Here, R is gas constant, b is the number of total degrees of freedom of a molecule and

related to specific heat ratio, $\gamma = (b+2)/b$. Equs (3.2a-3.2c) are mass, momentum and energy conservation relations, while Equs. (3.2d, 3.2e) are momentum and energy flux relations. With these relations, the BGK Boltzmann equation

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla f = (g - f)/\tau \tag{3.3}$$

can recover Euler equations, Equ. (3.1).

We can see that the conservation relations of mass and momentum Equs. (3.2a, 3.2b), as well as the momentum flux relation Equ. (3.2d), are the same as those of CF-VIIF. So we can still use the circular function in the velocity space to satisfy these three relations. The other two relations Equs. (3.2c, 3.2e) are energy conservation and flux relations. Since only the total energy of those extra degrees makes the difference, we try to add an extra degree of energy into the circular function CF-VIIF.

We consider a very simple circular function, g_s, as shown in in Fig. 3.1

$$g_s = \begin{cases} \dfrac{\rho}{2\pi c} & \text{if } \|\boldsymbol{\xi} - \mathbf{u}\| = c \equiv \sqrt{D(\gamma-1)e} \\ & \text{and } \lambda = e_p = \left[1 - \dfrac{D}{2}(\gamma - 1)\right] e \\ 0 & \text{else} \end{cases} \tag{3.4}$$

where λ is an extra axis standing for the total energy of all the extra degrees, and $D = 2$ is the spatial dimension. Here, e_p is the mean flow rest energy which comes from the extra energy of molecules. This g_s means that all mass, momentum and energy concentrate on a circle located in a 3D space of $\xi_x - \xi_y - \lambda$. So, its energy $d(\rho\zeta)$ consists of two parts: kinetic energy $\dfrac{1}{2}\boldsymbol{\xi}^2$ and the extra energy $\lambda = e_p$.

For a small arc ds on the circle, the mass, momentum and energy are

$$\begin{aligned} d\rho &= \frac{\rho}{2\pi c} ds \\ d(\rho\xi_\alpha) &= \frac{\rho}{2\pi c}(u_\alpha + c_\alpha)\, ds \\ d(\rho\zeta) &= \frac{\rho}{2\pi c}\left[\frac{1}{2}\xi_\chi^2 + \lambda\right] ds = \frac{\rho}{2\pi c}\left[\frac{1}{2}(u_\chi + c_\chi)^2 + e_p\right] ds \end{aligned}$$

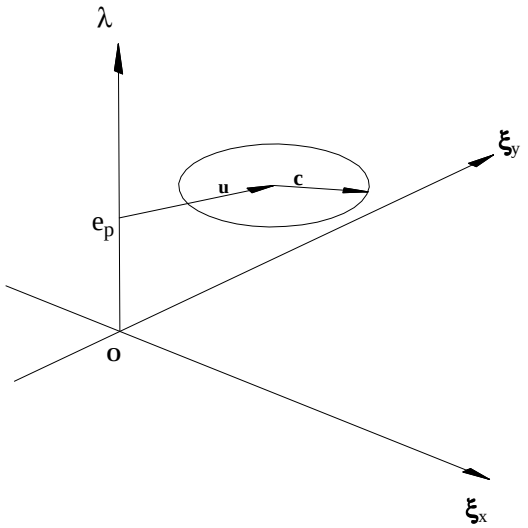

Figure 3.1: The schematic of the circular function g_s. It is located on a plane $\lambda = e_p$ in the $\xi_x - \xi_y - \lambda$ space. **u** is the mean velocity and **c** is the effective peculiar velocity.

where c_α is the component of vector **c** and $c = \| \mathbf{c} \|$.

Taking account

$$ds = cd\theta$$

and

$$\oint \mathbf{c}ds = c^2 \int_0^{2\pi} (\cos(\theta)\mathbf{i} + \sin(\theta)\mathbf{j})d\theta = 0\mathbf{i} + 0\mathbf{j}$$

It is easy to verify that this circular function satisfies the following relations

$$\oint d\rho = \rho \tag{3.5a}$$

$$\oint d(\rho\xi_\alpha) = \rho u_\alpha \tag{3.5b}$$

$$\oint (u_\beta + c_\beta)\, d(\rho\xi_\alpha) = \rho\,(u_\alpha u_\beta + RT\delta_{\alpha\beta}) \tag{3.5c}$$

$$\oint d(\rho\zeta) = \rho\left(u_\chi^2 + \frac{b}{2}RT\right) \tag{3.5d}$$

$$\oint \xi_\alpha d(\rho\zeta) = \left(\rho u_\chi^2 + \frac{b+2}{2}RT\right) u_\alpha \tag{3.5e}$$

These relations are the same as Equ. (3.2). So, we can use this circular function

to replace the Maxwellian function to derive f_i^{eq} for inviscid compressible flows. For simplicity, this function is named as Circular Function for Inviscid Compressible Flows (CF-ICF) in this thesis.

3.2 Deriving lattice models for inviscid compressible flows

We have shown that CF-ICF satisfies the constraints Equ. (3.2) and Euler equations can be recovered. Like what we have done with CF-VIIF in last chapter, it is expected that the equilibrium distribution function in a lattice model can be obtained by discretizing CF-ICF onto a lattice in such a way that the relations Equ. (3.5) can be conserved in the context of LBM when the integrals are replaced by the summations. In this section, we will study what condition of discretization should be satisfied.

3.2.1 Constraints of discretization from CF-ICF to a lattice model

Different from CF-VIIF which is a two variable function (ξ_x and ξ_y), CF-ICF is a function in three-dimensional space $\xi_x - \xi_y - \lambda$. Thus, how to deal with the third variable λ is a new problem.

Suppose that in the $\xi_x - \xi_y - \lambda$ space, there are $1 \dots N$ points, $\mathbf{e}_i$. CF-ICF will be discretized to all $\mathbf{e}_i$. For any $d\rho$ on the circle, it has a contribution $\phi_i(\boldsymbol{\xi}, \lambda)d\rho$ on the discrete velocity $\mathbf{e}_i$, where $\phi_i(\boldsymbol{\xi}, \lambda)$ is the assigning function. Thus, the accumulated density on $\mathbf{e}_i$ is Thus, the accumulated density on $\mathbf{e}_i$ is

$$\rho_i = \frac{\rho}{2\pi c} \oint \phi_i \left(\boldsymbol{\xi}, \lambda\right) ds \tag{3.6}$$

If we use ρ_i as the particle equilibrium distribution function f_i^{eq}, what conditions do $\phi_i(\boldsymbol{\xi}, \lambda)$ have to satisfy? Now, let us study this problem.

For convenience, we consider a case in which all $\mathbf{e}_i$ are located on the plane of $\lambda = e_p$.

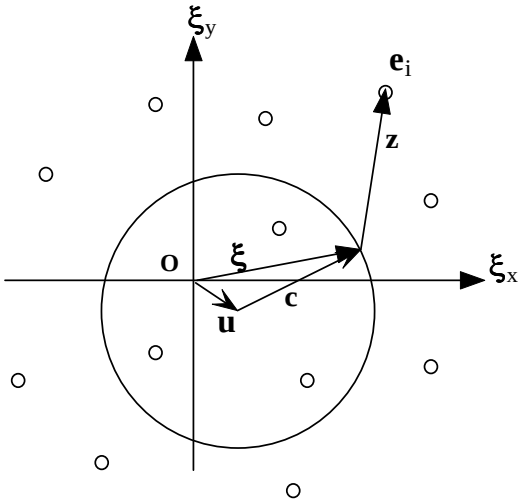

Figure 3.2: Configuration of the circle and the discrete velocity vectors in the velocity space of $\lambda = e_p$. $\mathbf{e}_i$ is one of the discrete velocity vectors, $\mathbf{u}$ is the mean velocity, $\mathbf{c}$ is the effective peculiar velocity and $\mathbf{z}_i$ is the vector from the position on the circle to $\mathbf{e}_i$.

On this plane, $\phi_i(\boldsymbol{\xi}, \lambda)$ is reduced to $\phi_i(\boldsymbol{\xi})$. And Equ. (3.6) can be written as

$$\tilde{\rho}_i = \frac{\rho}{2\pi c} \oint \phi_i \left(\boldsymbol{\xi}\right) ds \tag{3.7}$$

On this plane, Equ. (3.5) can be written as the form of summation

$$\sum_{i=1}^{N} \tilde{\rho}_i = \oint d\rho \tag{3.8a}$$

$$\sum_{i=1}^{N} \tilde{\rho}_i \, e_{i\alpha} = \oint d(\rho \xi_\alpha) \tag{3.8b}$$

$$\sum_{i=1}^{N} \tilde{\rho}_i \, e_{i\alpha} e_{i\beta} = \oint \left(\mathbf{u} + \mathbf{c}\right)_\beta d(\rho \xi_\alpha) \tag{3.8c}$$

$$\sum_{i=1}^{N} \tilde{\rho}_i \left(\frac{1}{2} e_{i\chi}^2 + e_p\right) = \oint d(\rho \zeta) \tag{3.8d}$$

$$\sum_{i=1}^{N} \tilde{\rho}_i \left(\frac{1}{2} e_{i\chi}^2 + e_p\right) e_{i\alpha} = \oint \xi_\alpha d(\rho \zeta) \tag{3.8e}$$

In Fig. 3.2, we can see that the relation between the discrete velocity $\mathbf{e}_i$ and the original

velocity $\boldsymbol{\xi}$ is

$$\mathbf{e}_i = \boldsymbol{\xi} + \mathbf{z}_i(\boldsymbol{\xi}) \tag{3.9}$$

Substituting Equ. (3.9), Equ. (3.7) into Equ. (3.8a), we can get

$$\sum_{i=1}^{N} \tilde{\rho}_i = \oint \frac{\rho}{2\pi c} \sum_{i=1}^{N} \phi_i(\boldsymbol{\xi})\, ds \equiv \oint \frac{\rho}{2\pi c} ds \Rightarrow \sum_{i=1}^{N} \phi_i(\boldsymbol{\xi}) = 1 \tag{3.10}$$

Then substituting Equ. (3.7), (3.9) and (3.10) into (3.8b) gives

$$\begin{aligned} \sum_{i=1}^{N} \tilde{\rho}_i\, e_{i\alpha} &= \oint \frac{\rho}{2\pi c} \xi_\alpha ds + \oint \frac{\rho}{2\pi c} \sum_{i=1}^{N} \mathbf{z}_i(\boldsymbol{\xi})_\alpha \phi_i(\boldsymbol{\xi})\, ds \equiv \oint \frac{\rho}{2\pi c} \xi_\alpha ds \\ &\Rightarrow \sum_{i=1}^{N} \mathbf{z}_i(\boldsymbol{\xi})_\alpha \phi_i(\boldsymbol{\xi}) = 0 \end{aligned} \tag{3.11}$$

Furthermore, substituting Equ. (3.7), Equs. (3.9-3.11) into (3.8c) leads to

$$\begin{aligned} \sum_{i=1}^{N} \tilde{\rho}_i\, e_{i\alpha} e_{i\beta} &= \oint \frac{\rho}{2\pi c} \xi_\alpha \xi_\beta ds + \oint \frac{\rho}{2\pi c} \sum_{i=1}^{N} \mathbf{z}_i(\boldsymbol{\xi})_\alpha \mathbf{z}_i(\boldsymbol{\xi})_\beta \phi_i(\boldsymbol{\xi})\, ds \equiv \oint \frac{\rho}{2\pi c} \xi_\alpha \xi_\beta ds \\ &\Rightarrow \sum_{i=1}^{N} \phi_i(\boldsymbol{\xi})\, \mathbf{z}_i(\boldsymbol{\xi}) \mathbf{z}_i(\boldsymbol{\xi}) = 0 \end{aligned} \tag{3.12}$$

With Equs. (3.10-3.12), Equ. (3.8d) can be satisfied automatically. Finally, substituting Equs. (3.7, 3.9-3.12) into Equ. (3.8e), we can get

$$\sum_{i=1}^{N} \phi_i(\boldsymbol{\xi})\, \mathbf{z}_i(\boldsymbol{\xi}) \mathbf{z}_i(\boldsymbol{\xi}) \mathbf{z}_i(\boldsymbol{\xi}) = 0 \tag{3.13}$$

Equs. (3.10-3.13) are the constraints for the assigning function $\phi_i(\boldsymbol{\xi})$.

We can see that these constraints are the same as those for CF-VIIF developed in the last chapter. So, we can use the same lattice (in Fig. 2.3) and the Lagrangian interpolation polynomials Equ. (2.19) derived in the last chapter. Substituting these base functions, Equ. (2.19), into Equ. (3.7), we can get all $\tilde{\rho}_i$ which are the same as those f_i^{eq} listed in Equ. (2.20).

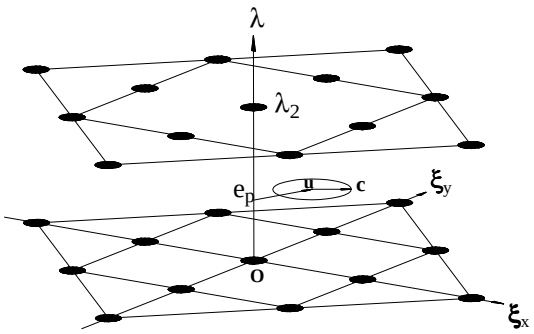

Figure 3.3: D2Q13L2 lattice.

3.2.2 Introduction of energy-levels to get fully discrete f_i^{eq}

Notice that the summation relations, Equ. (3.8), contain e_p which is a mean flow variable dependent on time and space. In a pure lattice model, only f_i^{eq} depends on space and time. So we should find a way to express the summation relations with only f_i^{eq} , $\mathbf{e}_i$ and some other constants.

Since λ is another dimension standing for rest energy, we may introduce several fixed energy-levels to assign e_p. As e_p is appeared linearly in Equ. (3.8d and 3.8e), conservation of the first order moment of $\tilde{\rho}_i$ in λ is enough, which means that we can set two energy-levels, $\lambda_1 = 0$ and $\lambda_2 > e_p$. Linearly assigning every $\tilde{\rho}_i$ onto the two energy-levels gives

$$\begin{cases} \rho_{i1} = \tilde{\rho}_i \left(\lambda_2 - e_p\right) / \lambda_2 \\ \rho_{i2} = \tilde{\rho}_i \, e_p / \lambda_2 \end{cases}$$

which makes $\sum_{v=1}^{2} \rho_{iv}\lambda_v = \tilde{\rho}_i e_p$. Thus ρ_{iv} can satisfy

$$\begin{cases} \sum_{i=1}^{N}\sum_{v=1}^{2} \rho_{iv} = \rho \\ \sum_{i=1}^{N}\sum_{v=1}^{2} \rho_{iv} e_{i\alpha} = \rho u_\alpha \\ \sum_{i=1}^{N}\sum_{v=1}^{2} \rho_{iv} e_{i\alpha} e_{i\beta} = \rho(u_\alpha u_\beta + RT\delta_{\alpha\beta}) \\ \sum_{i=1}^{N}\sum_{v=1}^{2} \rho_{iv} \left(\frac{1}{2}e_{i\chi}^2 + \lambda_v\right) = \rho\left(\frac{1}{2}u_\chi^2 + \frac{b}{2}RT\right) \\ \sum_{i=1}^{N}\sum_{v=1}^{2} \rho_{iv} \left(\frac{1}{2}e_{i\chi}^2 + \lambda_v\right) u_\alpha = \rho\left(\frac{1}{2}u_\chi^2 + \frac{b+2}{2}RT\right) u_\alpha \end{cases} \tag{3.14}$$

and we can use ρ_{iv} as f_{iv}^{eq} . Now we get a two-dimensional 13-velocity and 2-energy-levels lattice model, named D2Q13L2 (Fig. 3.3).

All the derivation can be implemented with Maple or Mathematica. Appendix A.2 presents the Maple scripts to get the base function $\phi_i(x, y)$, f_{iv}^{eq} and optimized C language codes. Here, we can see that there is only one free parameter, λ_2, in our models and it is easily determined as 1 (see Section 3.4).

3.2.3 Deriving a 1D lattice Boltzmann model for 1d Euler equations

In order to simulate 1D shock tube problems, we construct a one dimensional model. Here, a simpler two-point function is used:

$$g_s = \begin{cases} \dfrac{\rho}{2} & \text{if } \xi - u = \pm c = \pm\sqrt{D\left(\gamma-1\right)e} \\ & \text{and } \lambda = e_p = \left[1 - \dfrac{D}{2}\left(\gamma - 1\right)\right] e \\ 0 & \text{else} \end{cases}$$

where $D = 1$. Using 5-node Lagrange interpolation method, we get a D1Q5L2 model as

$$\begin{cases} \rho_0 = \dfrac{\rho\left(d_1{}^2 d_2{}^2 - d_1{}^2 u^2 - d_1{}^2 c^2 - d_2{}^2 u^2 - d_2{}^2 c^2 + u^4 + 6\,u^2c^2 + c^4\right)}{d_1{}^2 d_2{}^2} \\ \rho_1 = \dfrac{\rho\left(3\,d_1 uc^2 - d_1\,d_2{}^2\,u - d_2{}^2 u^2 - d_2{}^2 c^2 + 6\,u^2c^2 + c^4 + u^4 + d_1 u^3\right)}{2\,d_1{}^2\left(d_1{}^2 - d_2{}^2\right)} \\ \rho_2 = \dfrac{\rho\left(-3\,d_1 uc^2 + d_1\,d_2{}^2\,u - d_2{}^2 u^2 - d_2{}^2 c^2 + 6\,u^2c^2 + c^4 + u^4 - d_1 u^3\right)}{2\,d_1{}^2\left(d_1{}^2 - d_2{}^2\right)} \\ \rho_3 = -\dfrac{\rho\left(-d_2\,d_1{}^2\,u + 3\,d_2 uc^2 - d_1{}^2 u^2 - d_1{}^2 c^2 + 6\,u^2c^2 + c^4 + u^4 + d_2 u^3\right)}{2\,d_2{}^2\left(d_1{}^2 - d_2{}^2\right)} \\ \rho_4 = -\dfrac{\rho\left(d_2\,d_1{}^2\,u - 3\,d_2 uc^2 - d_1{}^2 u^2 - d_1{}^2 c^2 + 6\,u^2c^2 + c^4 + u^4 - d_2 u^3\right)}{2\,d_2{}^2\left(d_1{}^2 - d_2{}^2\right)} \end{cases}$$

$$\begin{cases} f_{i0}^{eq} = \rho_i\left(\lambda_2 - e_p\right)/\lambda_2 \\ f_{i1}^{eq} = \rho_i e_p/\lambda_2 \end{cases}$$

where d_1 and d_2 are two integers, $d_1 < d_2$. The 5 particle velocities are 0, d_1, $-d_1$, d_2, $-d_2$. In Appendix A.3, the Maple script to derive this model is presented.

3.3 Chapman-Enskog analysis

It can be easily shown that the present models recover to Euler equations with Chapman-Enskog expansion.

With Chapman-Enskog expansion, the distribution function f_{iv} is approximated by f_{iv}^{eq}

$$f_{iv} = f_{iv}^{eq} + O\left(\varepsilon\right) \tag{3.15}$$

By multiplying Equ. (1.20) with the collision invariant vector $\left[1, e_{i\beta}, \dfrac{e_{i\chi}^2}{2} + \lambda_v\right]^T$ and doing summation, we get

$$\sum_{i=1,v=1}^{i=N,v=2} \left\{ \left[\frac{\partial f_{iv}}{\partial t} + \partial_\alpha\left(f_{iv} e_{i\alpha}\right)\right] \begin{bmatrix} 1 \\ e_{i\beta} \\ \dfrac{e_{i\chi}^2}{2} + \lambda_v \end{bmatrix} \right\} = \sum_{i=1,v=1}^{i=N,v=2} \left[\frac{f_{iv}^{eq} - f_{iv}}{\tau}\right] \begin{bmatrix} 1 \\ e_{i\beta} \\ \dfrac{e_{i\chi}^2}{2} + \lambda_v \end{bmatrix} \tag{3.16}$$

By substituting Equs. (3.14) and (3.15) into Equ. (3.16), we can get the macroscopic equations as

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u_\alpha\\ \rho E\end{bmatrix}+\partial_\alpha\begin{bmatrix}\rho u_\alpha\\ \rho u_\alpha u_\beta+p\\ \rho\left(E+p\right)u_\alpha\end{bmatrix}=O\left(\varepsilon\right) \tag{3.17}$$

So Euler equations can be approximated with the truncation error $O(\varepsilon)$.

3.4 FVM formulations in curvilinear coordinate system

Since DVBE recovers Euler equations in the order of $O(\varepsilon)$, Knudsen number ε should be very small in numerical simulations. And the dimensionless width of an interface (a shock wave or a contact discontinuity) is in the order of ε. In order to resolve the interface, the mesh size should be smaller than ε. The use of such small mesh size is unacceptable. Kataoka and Tsutahara [27] proved that with a consistent numerical scheme of the p-th order accuracy, DVBE approaches to the weak form solution of Euler equations even if the mesh size is much larger than ε and the error is $\max(O(\Delta x^p), O(\varepsilon))$. This means that we can solve DVBE on a coarser grid.

To well capture the discontinuity, artificial dissipation is needed. The artificial dissipation in the present work comes from two parts. One is the model dissipation, $O(\varepsilon)$. The other is the numerical dissipation resulted from the numerical scheme adopted. The model viscosity is related to ε. Since ε is taken as very small, the model dissipation is not enough to capture discontinuities without oscillation. So, the main dissipation comes from the numerical scheme. In this work, a third order MUSCL FVM scheme (Monotone Upstream-centered Scheme for Conservation Laws) [38] is used to solve DVBE.

Since FVM is used to solve DVBE, we write BGK DVBE as the conservative form in 2D general coordinates

$$\frac{\partial\,\hat{f}_k}{\partial t}+\frac{\partial\,\hat{F}_k}{\partial\xi}+\frac{\partial\,\hat{G}_k}{\partial\eta}=\hat{\Omega}_k \tag{3.18}$$

J is the Jacobian of coordinate transformation

$$J = \frac{\partial(\xi,\eta)}{\partial(x,y)} = \begin{vmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} \end{vmatrix}^{-1} \tag{3.19}$$

In Equ. (3.18), $\hat{f}_k$ is the distribution function, where k is the index of all distribution functions

$$\hat{f}_k = \frac{f_k}{J} \tag{3.20}$$

And the flux terms and source term

$$\hat{F}_k = \frac{f_k e_{k\xi}}{J} \qquad \hat{G}_k = \frac{f_k e_{k\eta}}{J} \qquad \hat{\Omega}_k = -\frac{f_k - f_k^{eq}}{\tau J} \tag{3.21}$$

The contravariant velocities are given by

$$\begin{cases} e_{k\xi} = \xi_x e_{kx} + \xi_y e_{ky} \\ e_{k\eta} = \eta_x e_{kx} + \eta_y e_{ky} \end{cases} \tag{3.22}$$

The semi-discretized form is

$$\frac{1}{J}\frac{\partial f_{k,i,j}}{\partial t} = -\left[\hat{F}_{k,i+\frac{1}{2},j} - \hat{F}_{k,i-\frac{1}{2},j}\right] - \left[\hat{G}_{k,i,j+\frac{1}{2}} - \hat{G}_{k,i,j-\frac{1}{2}}\right] - \hat{\Omega}_{k,i,j} \tag{3.23}$$

where i is a cell index, $\hat{F}_{k,i\pm\frac{1}{2},j}$ and $\hat{G}_{k,i,j\pm\frac{1}{2}}$ are numerical fluxes on the interfaces of a cell (i,j). They can be computed with a Riemann solver. For Equ. (3.18), the exact Riemann solver is available and cheap since it is a constant linear hyperbolic system. $\hat{F}_{k,i+\frac{1}{2},j}$ for instance

$$\hat{F}_{k,i+\frac{1}{2},j} = \begin{cases} (f_L)_{k,i+\frac{1}{2},j}\, e_{k\xi}/J & \text{if } e_{k\xi} \geq 0 \\ (f_R)_{k,i+\frac{1}{2},j}\, e_{k\xi}/J & \text{if } e_{k\xi} \leq 0 \end{cases} \tag{3.24}$$

where f_L and f_R are the distribution function just on the left and right side of the interface $(i+\frac{1}{2},j)$. In this work, they are determined by the third order MUSCL with the smooth

limiter to extrapolate the value of $\hat{f}_\alpha^n$ on the two side of an interface

$$\begin{cases} (f_L)_{k,i+\frac{1}{2},j} = f_{\alpha,i,j} + \left\{ \frac{s}{4} \left[(1-\kappa s)\,\Delta_- + (1+\kappa s)\,\Delta_+ \right] \right\}_i \\ (f_R)_{k,i+\frac{1}{2},j} = f_{\alpha,i+1,j} - \left\{ \frac{s}{4} \left[(1-\kappa s)\,\Delta_+ + (1+\kappa s)\,\Delta_- \right] \right\}_{i+1} \end{cases} \tag{3.25}$$

where $\kappa = 1/3$ and s is the Van Albada limiter [39]

$$s = \frac{2\Delta_+\Delta_- + \varepsilon^2}{\Delta_+^2 + \Delta_-^2 + \varepsilon}$$

Here ε is a small number (we set $\varepsilon = 10^{-6}$) preventing division by zero in region of null gradient and

$$\begin{aligned} (\Delta_+)_i &= f_{k,i+1,j} - f_{k,i,j} \\ (\Delta_-)_i &= f_{k,i,j} - f_{k,i-1,j} \end{aligned}$$

With $(f_L)_{i+\frac{1}{2}}$ and $(f_R)_{i+\frac{1}{2}}$, the numerical flux on the interface $i+\frac{1}{2}$ can be computed according to the exact Riemann solver Equ. (3.24). Computing $\hat{\Omega}_{k,i,j}$ is easy when a full explicit time scheme is applied for time advancement. However, in order to resolve the dimensionless relaxation time τ, the time step Δt should be smaller enough as compared to τ value.

In this work, the Euler forward scheme is applied for the temporal discretization. Although it is a first order explicit scheme, it is good enough not only for steady flows but also for many unsteady problems since the time step Δt is limited by τ, making the CFL number very small. Because no convergence acceleration methods were applied, the computing is very time consuming.

Because the lattice of the LB model is fixed, the problems to be simulated should be normalized to the scale of the lattice. So, dimensionless equations should be used. There are three independent reference variables, reference density ρ_0, length L_0 and internal energy e_0 (note, this e_0 is not a discrete velocity vector). The other reference and

dimensionless variables are defined as follows

$$
\begin{aligned}
&U_0 = \sqrt{e_0} \qquad t_0 = \frac{L_0}{U_0} \\
&\breve{t} = \frac{t}{t_0} \qquad \breve{x} = \frac{x}{L_0} \qquad \breve{\rho} = \frac{\rho}{\rho_0} \\
&\breve{e} = \frac{e}{e_0} \qquad \breve{\tau} = \frac{\tau}{t_0} \qquad \breve{\lambda}_2 = \frac{\lambda_2}{e_0} \quad \breve{u} = \frac{u}{U_0}
\end{aligned}
$$

Here $\breve{\tau}$ is the Knudsen number and is very small for inviscid flows (we set $\breve{\tau} = 10^{-4} \sim 10^{-3}$). So the dimensionless time step is seriously limited (we set $\Delta\breve{t} = 0.3\breve{\tau}$). The value of e_0 is very important. It determines U_0 and normalizes e to $\breve{e}$ from which $\breve{c}$ is computed. And in order to make the circle located inside the lattice to avoid extrapolation, we need $|\breve{\mathbf{u}}| + \breve{c} < \sqrt{2}$ where $\sqrt{2}$ is the shortest distance from the original point to the edges of the D2Q13 lattice. So, e_0 should be big enough. Determining this parameter is easy and it has clear physical and mathematical meaning. For safety, e_0 can be set a little bit greater than the maximum specific stagnation internal energy, $\max(e^*)$, in the whole flow field. Also, λ_2 should be big enough. In our simulations, we set $\lambda_2 = e_0$ for simplicity. So $\breve{\lambda}_2 = 1$ and we have the following relations

$$
\begin{cases}
\breve{\rho} = \sum\limits_{i=1}^{13} \sum\limits_{v=1}^{2} f_{iv}^{eq} \\
\breve{\rho}\breve{u} = \sum\limits_{i=1}^{13} \sum\limits_{v=1}^{2} f_{iv}^{eq} \mathbf{e}_i \\
\breve{\rho}\breve{E} = \sum\limits_{i=1}^{13} \left[\sum\limits_{v=1}^{2} f_{iv}^{eq} \dfrac{\mathbf{e}_i^2}{2} + f_{i2}^{eq} \right]
\end{cases}
$$

and

$$
\begin{cases}
\breve{e} = \breve{E} - \frac{\breve{u}^2}{2} \\
\breve{p} = (\gamma - 1)\, \breve{\rho}\breve{e}
\end{cases}
$$

3.5 Boundary conditions

In our simulations in this work, many boundary conditions are involved, such as subsonic inlet/outlet, supersonic inlet/outlet and adiabatic inviscid wall boundary conditions. For all the boundary conditions, we can use the following way to implement them. At first,

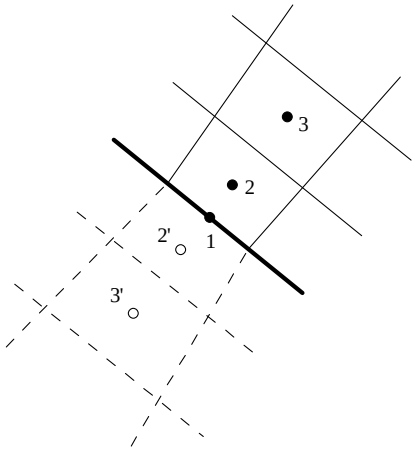

Figure 3.4: Implementation of slip wall condition. The thick line is the wall, cells drew with thin solid lines are cells in fluid domain, and cells drew with dash lines are ghost cells inside the wall.

we need to determine mean flow variables (ρ, u, v and e) on the boundary using available means of traditional CFD. Then, we can compute f_{iv}^{eq} on the boundary according to ρ, u, v and e on the boundary. Since inviscid flows are considered and shear stress, heat conduction and work done by shear stress are negligible, the non-equilibrium part of f_i which results in the viscous effects, can be neglected. This guarantees that the use of f_{iv}^{eq} on the boundary is accurate enough.

We also implement the slip adiabatic wall boundary condition with a reflection-projection method. On a rectangular grid, it is easy to perform the specular reflecting operation to implement slip-wall condition because every velocity has its mirror peer. However, this is not the case for curvilinear grids. We have a little more work to do for the curvlinear gird. Fig. 3.4 shows the wall and the nearby cells. The thick line is the wall, cells drew with thin solid lines are cells in fluid domain. We first reflect every velocity of the center of cell 2, $\mathbf{e}_i$. Usually, the reflected velocity $\mathbf{e}_i'$ is not coincident with any node in the D2Q13L2 lattice (Fig. 3.5). So we assign (project) its corresponding $f_{iv}(2)$ onto every node of the D2Q13L2 lattice. And the accumulated value on the velocity $\mathbf{e}_j$,

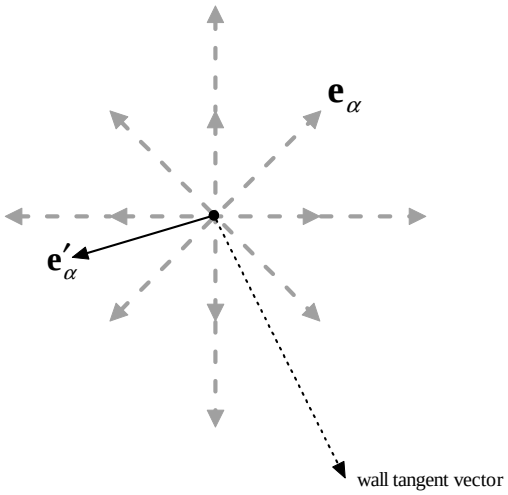

Figure 3.5: Reflection-projection method for the inviscid wall boundary condition.

$f_{jv}(2')$ is

$$f_{jv}(2') = \sum_{i=1}^{13} f_{iv}\phi_j\left(e_{ix}, e_{iy}\right)$$

Then distribution function on the wall, f_{iv}^w, can be computed by averaging these two values

$$f_{iv}^w = \frac{f_{iv}(2) + f_{iv}(2')}{2}$$

Thus the flux through the wall can be computed. Alternatively, with the same method, after computing the distribution function $f_{iv}(3')$ in the ghost cell $3'$, the flux through the wall can be computed with MUSCL scheme when cells 2 and $2'$ are internal cells. It is not difficult to prove that this reflection-projection operation satisfies the inviscid wall condition and is of the second order accuracy.

3.6 Numerical results

This section presents numerical results of many test cases, including different dimensional problems (one dimensional and two dimensional), different Mach numbers (subsonic, transonic, supersonic and hypersonic flows), different geometries (rectangular and irregular

domains). All these numerical tests show that our deriving method, models, numerical procedure and implementation of boundary conditions are right.

3.6.1 Sod shock tube

The first test case is the Sod shock tube problem. This is a one-dimensional problem. So, we simulated it with the D1Q5L2 model. The initial condition is

$$\begin{aligned} (\rho_L, u_L, e_L) &= (1, 0, 2.5) \qquad && -0.5 < x < 0 \\ (\rho_R, u_R, e_R) &= (0.125, 0, 2) \qquad && 0 < x < 0.5 \end{aligned}$$

In this case, the reference internal energy $e_0 = 4 > \max(e^*) = 2.5$, $\rho_0 = 1$, $L_0 = 1$ and $\breve{\varepsilon} = 10^{-4}$. The mesh size $\Delta x = 1/200$ and time step $\Delta \breve{t} = \breve{\varepsilon}/4$. Before the waves propagate to the two boundary ends, the distribution functions at the boundary can be set as the equilibrium distribution functions computed from the initial value of macroscopic variables. The initial value of f_i is set as equilibrium function. The computed density, velocity, pressure and internal energy profiles at $t = 0.22$ are shown as symbols in Fig. 3.6. Also displayed in this figure are the exact solutions (solid lines). Clearly, the present results agree excellently well with the exact solution.

3.6.2 Lax shock tube

The second test case is the Lax shock tube problem, which is also a 1D problem and is solved by the D1Q5L2 model. The initial condition of the problem is given as

$$\begin{aligned} (\rho_L, u_L, e_L) &= (0.445, 0.698, 19.82) \qquad && -0.5 < x < 0 \\ (\rho_R, u_R, e_R) &= (0.5, 0, 2.855) \qquad && 0 < x < 0.5 \end{aligned}$$

We set $e_0 = 30 > \max(e^*) = 19.82$, $\rho_0 = 1$, $L_0 = 1$, $\breve{\varepsilon} = 10^{-4}$. The mesh size is the same as those of the Sod case. And time step is $\Delta \breve{t} = \breve{\varepsilon}/4$. The initial value of f_i is set as equilibrium function. The computed density, velocity, pressure and internal energy profile at $t = 0.14$ are shown and compared with the exact solution in Fig. 3.7

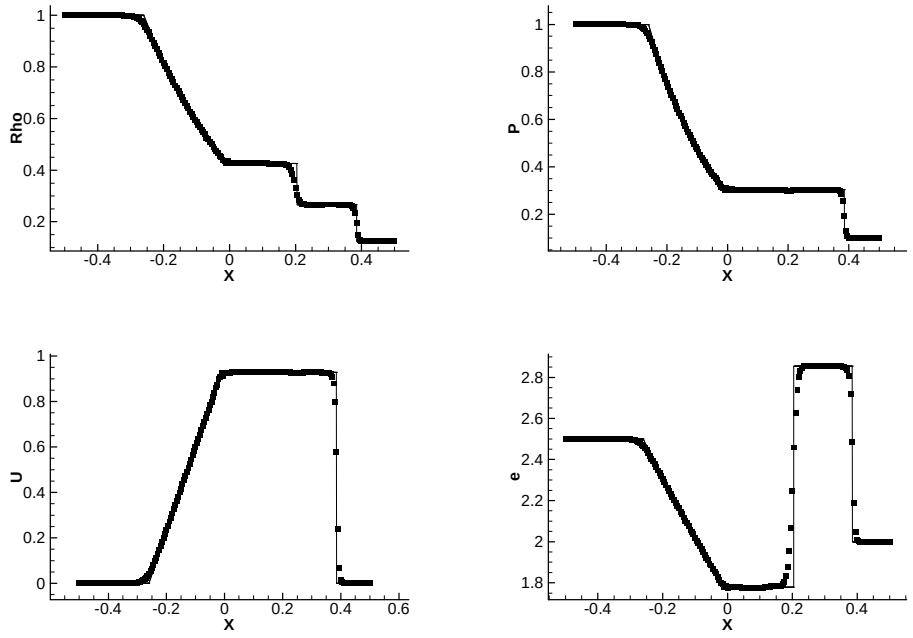

Figure 3.6: Density (left up), pressure (right up), velocity (left bottom) and internal energy (right bottom) profiles of Sod case.

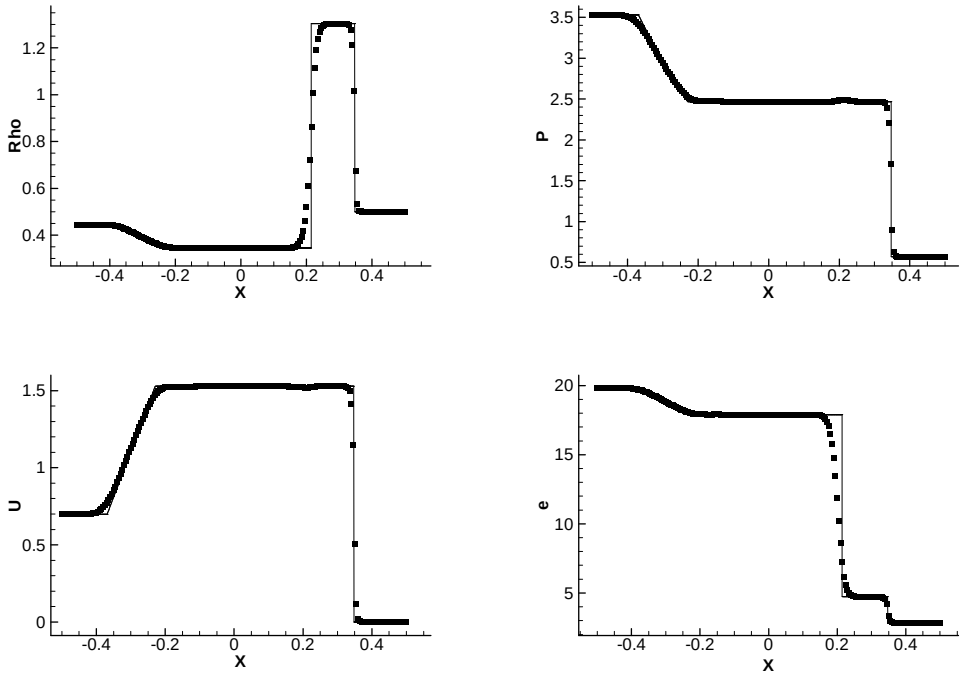

Figure 3.7: Density (left up), pressure (right up), velocity (left bottom) and internal energy (right bottom) profiles of Lax case.

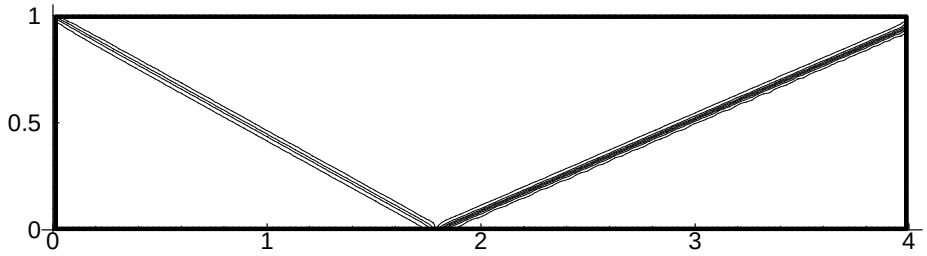

Figure 3.8: Density contour of shock reflection on a plane.

3.6.3 A 29° shock reflecting on a plane

The two cases presented above are actually 1D problems. To further validate the present approach, we simulated a 2D supersonic flow in which a 29° shock of 2.9 Mach number reflects on a wall. The computational domain is a rectangle of length 4 and height 1. Uniform mesh of 150×100 is used. The left and top boundary conditions are

$$(\rho_L, u_L, v_L, e_L) = (1, 2.9, 0.0, 1.785714)$$
$$(\rho_T, u_T, e_T) = (1.69997, 2.61934, -0.50633, 2.247378)$$

The right boundary condition is supersonic outflow where extrapolation is applied. At the bottom, it is a slip wall below which 2 level of mirror points are set and the value of particle distribution functions are obtained by mirror reflecting their symmetric particle distribution functions. In the computation, we set $\rho_0 = 1$, $L_0 = 1$, $\varepsilon = 10^{-4}$ and $e_0 = 8 > \max(e^*) = 7.78$. Still $\breve{\varepsilon} = 10^{-4}$, and the time step is time step is $\Delta\breve{t} = \breve{\varepsilon}/4$. Fig. 3.8 presents the density contour in which the shock wave is captured excellently.

3.6.4 Double Mach reflection

In the last case of shock reflection, the pressure ratio cross the shock wave is 2.14, which is not high enough to generate a strong shock wave. Now, we simulate a case of high pressure ratio, the double Mach reflection problem (pressure ratio is 116.5), to show the ability of the present model to simulate strong shock waves. For this case, a normal shock

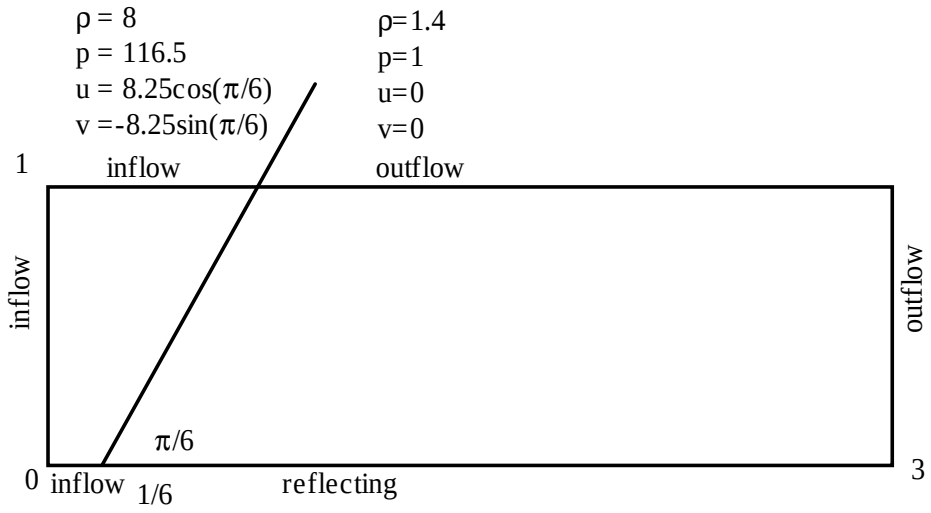

Figure 3.9: Schematic diagram of the double Mach reflection case.

wave with Mach number of 10 passes through a 30° wedge (Fig. 3.9). Uniform mesh size of 300×100 is used. The reference variables are set as $\rho_0 = 1$, $L_0 = 1$, $\breve{\varepsilon} = 10^{-4}$, $\Delta\breve{t} = \breve{\varepsilon}/4$ and $e_0 = 75 > \max(e^*) = 72.8$.

The computed density and pressure contours are shown in Fig. 3.10. These results are in good agreement with those obtained by using the upwind scheme to solve Euler equations [40]. The complex features such as oblique shocks and triple points are well captured.

3.6.5 Flow past a bump in a channel

All the cases presented above are simulated on uniform Cartesian grids. In order to validate the capability of our method for its application on curvilinear grids, the following cases will be studied.

First, we consider a flow in the GAMM channel (Fig. 3.11) which has a 10% circular bump ($h = 0.1$) in it. The inflow Mach number is 0.675. Fig. 3.12 shows the 80×22 grid of the channel.

In the computation, total energy $e^* = 1.96$ for $u = 0.675$ and $\rho = 1$ of $Mach = 0.675$. The reference variables are $\rho_0 = 1$, $L_0 = h$, $\breve{\varepsilon} = 10^{-3}$, $\Delta\breve{t} = \breve{\varepsilon}/4$ and $e_0 = 2 > \max(e^*) =$

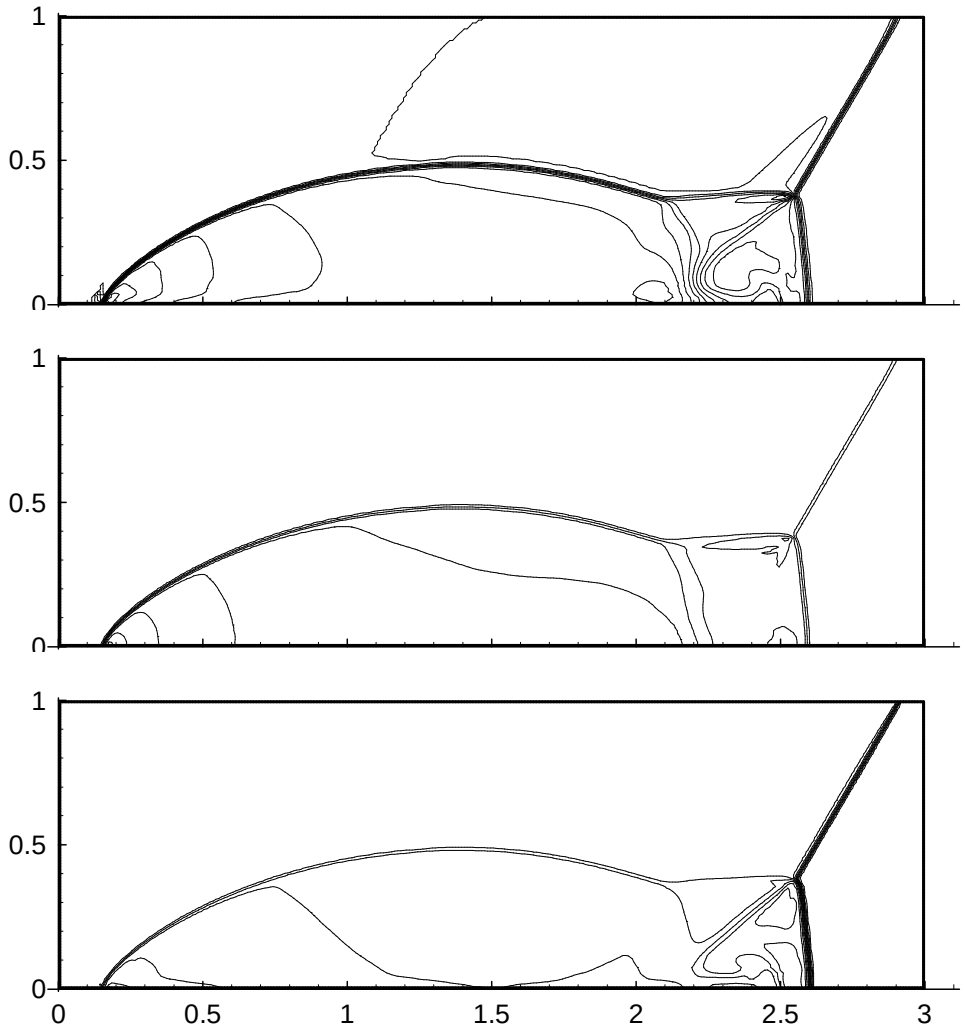

Figure 3.10: Density (top), pressure (middle) and internal energy (bottom) contours of the double Mach reflection case.

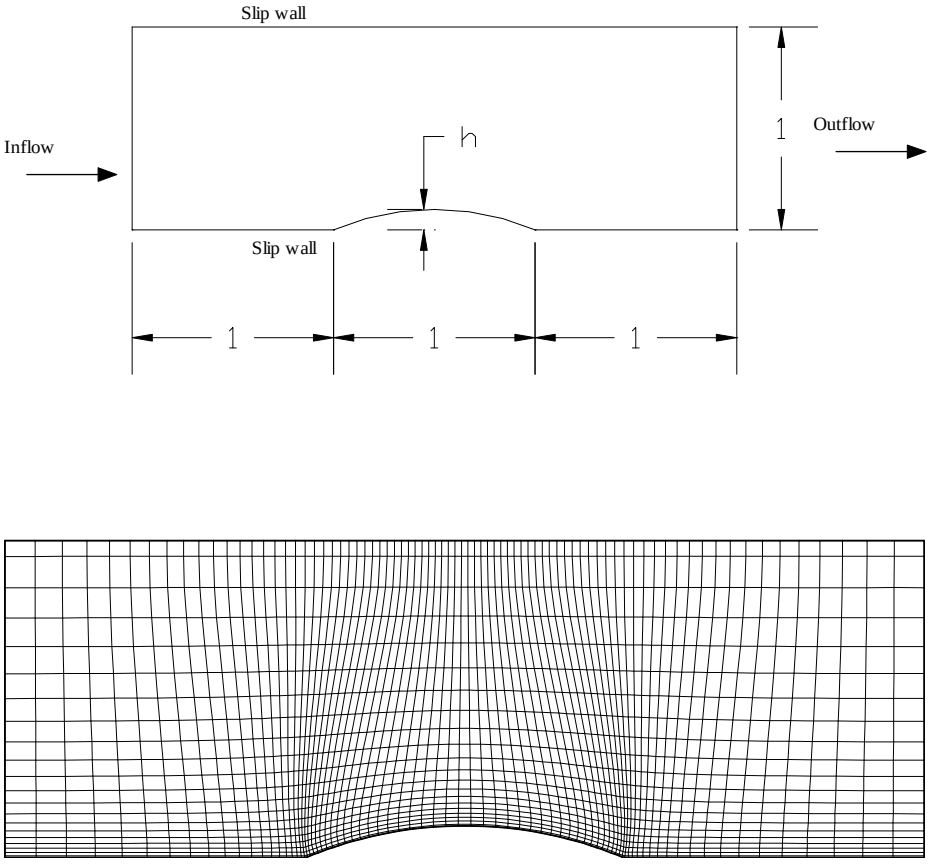

Figure 3.12: The structural curvilinear grid of channel with bump of 10%

1.96

The left boundary is subsonic inlet, the right boundary is set as extrapolating outflow, the top and bottom boundaries are slip adiabatic walls. The computed Mach number contour is shown in Fig. 3.13. And Fig. 3.14 shows the Mach number profiles on the walls. The solid dot line in this figure is the result computed by solving Euler equation with WENO scheme [41]. Obviously, the present results agree well with those obtained by the Euler solver.

3.6.6 Flows around Rae2822 airfoil

Two simulations were performed for this case. For the first one, $M_\infty = 0.75$ and $\alpha = 3^\circ$, while for the second one, $M_\infty = 0.729$, $\alpha = 2.31^\circ$. In these simulations, a 225×65 C-type

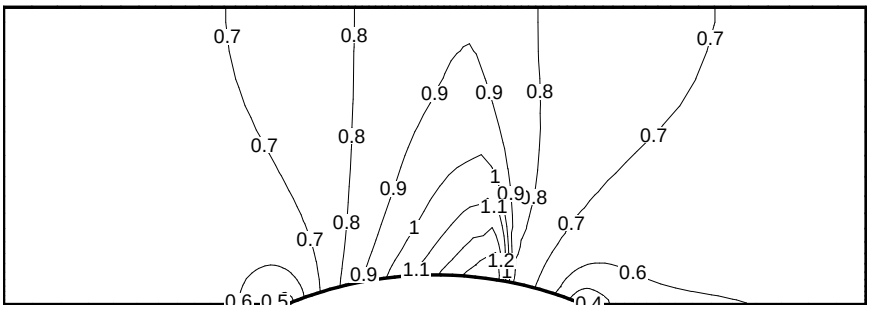

Figure 3.13: Mach number contour of $M_\infty = 0.675$ flow in the channel of 10%.

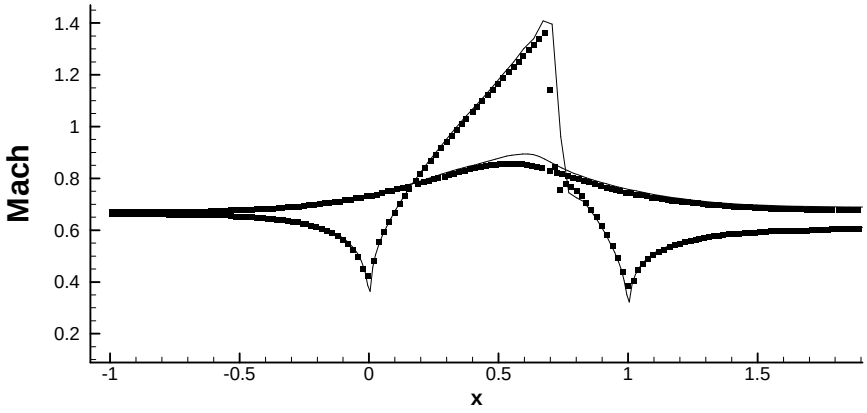

Figure 3.14: Distribution of Mach number along walls.

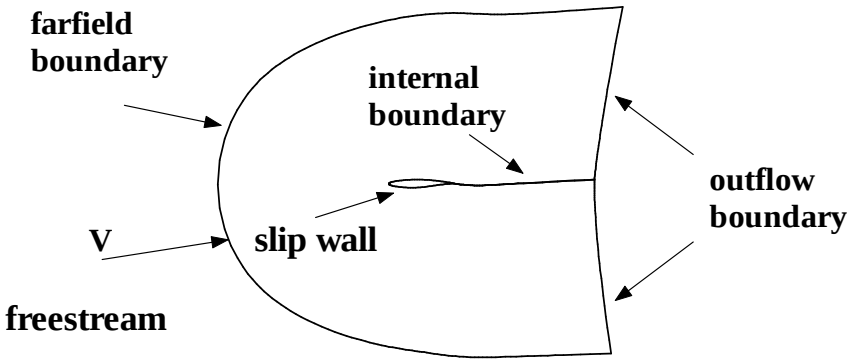

Figure 3.15: Boundary conditions of flow around Rae2822 airfoil.

grid was used. And the boundary conditions are shown in Fig. 3.15 in which the outer boundary is about 20 times of the chord length far from the airfoil.

In the computation, for $u_\infty = Mach$ and $\rho_\infty = 1$, the internal energy can be determined as

$$e_\infty = \frac{p_\infty}{(\gamma - 1)\rho_\infty} = \frac{1/\gamma}{(\gamma - 1) \times 1} = \frac{1}{\gamma(\gamma - 1)}$$

Then total energy e^* can be obtained according to the Mach number. The reference variables are $\rho_0 = 1$, $L_0 =$ chordlength, $\breve{\varepsilon} = 10^{-3}$, $\Delta\breve{t} = \breve{\varepsilon}/4$ and $e_0 > \max(e^*)$

The pressure contours of the first case are shown in Fig. 3.16. The shock wave is captured clearly. And the pressure coefficient profile is presented in Fig. 3.16 in which the results computed with the same grid by means of solving Euler solver with Jameson's central scheme [42] are shown as solid symbols. These two results agree excellently. For the second case, the pressure contours and pressure coefficient profiles are shown in Figs. 3.18 and 3.19 respectively. And the solid dots in Fig. 3.19 are the experimental result [43] which has some small difference from our result because the flow in the experiment is a turbulent flow in which the shock wave interacts with the boundary layers. Therefore, its stiffness and location are different from those of the numerical inviscid simulation.

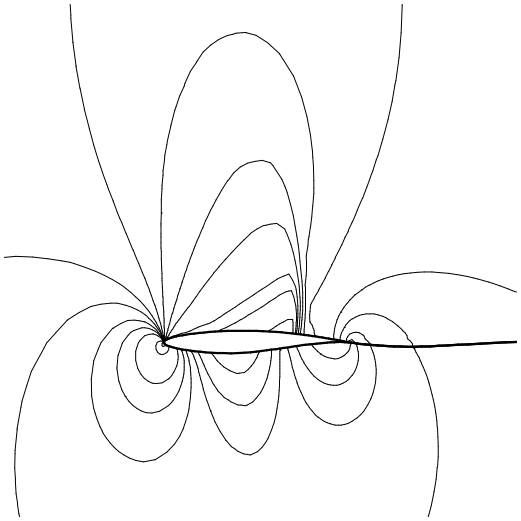

Figure 3.16: Pressure contours of flow over Rae2822 airfoil ($M_\infty = 0.75$ and $\alpha = 3^\circ$)

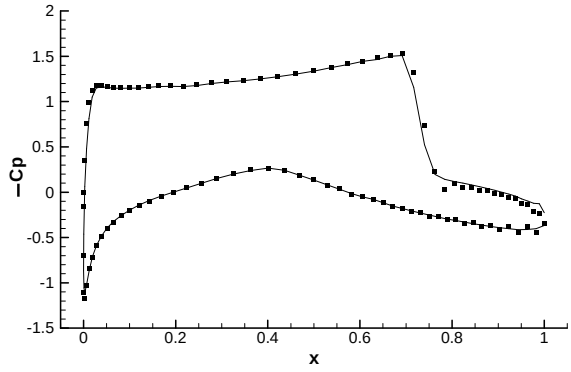

Figure 3.17: Pressure coefficient profiles of flow over Rae2822 airfoil ($M_\infty = 0.75$ and $\alpha = 3^\circ$)

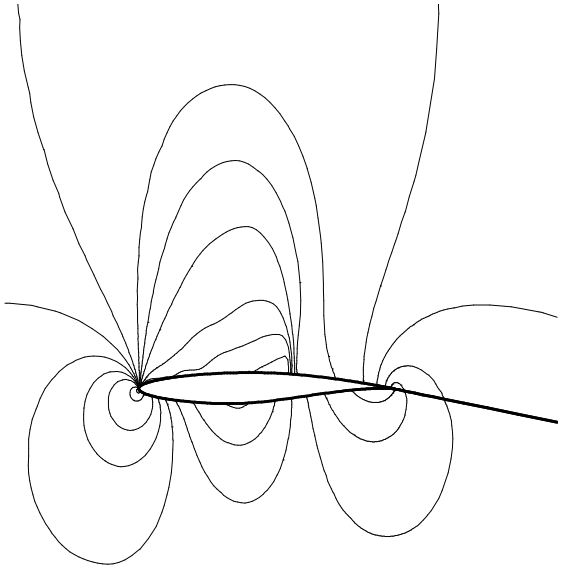

Figure 3.18: Pressure contours of flow over Rae2822 airfoil ($M_\infty = 0.729$ and $\alpha = 2.31^\circ$)

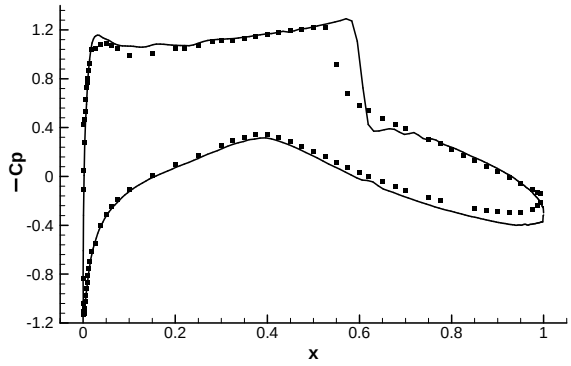

Figure 3.19: Pressure coefficient profiles of flow over Rae2822 airfoil ($M_\infty = 0.729$ and $\alpha = 2.31^\circ$)

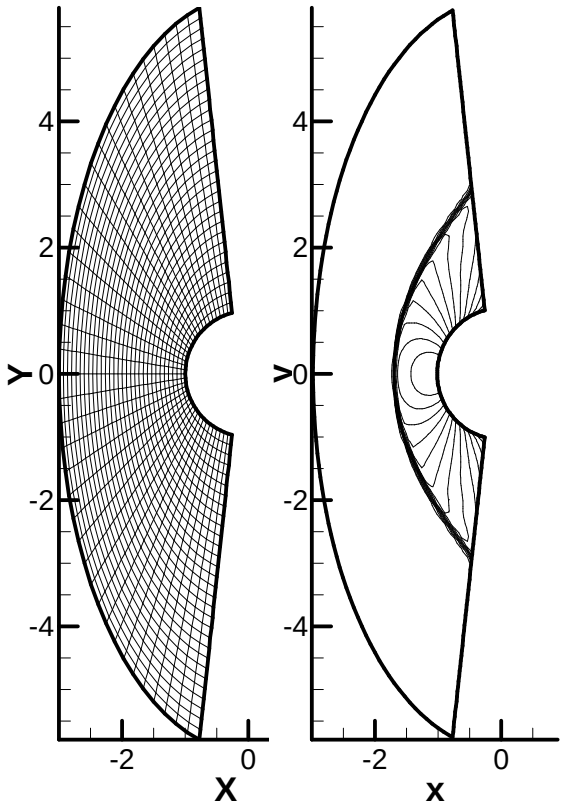

Figure 3.20: Mach 3 flow around a cylinder. Grid and pressure contour.

3.6.7 Supersonic flow over a two dimensional cylinder

Finally, supersonic flow over a cylinder is simulated for two different Mach numbers ($M_\infty = 3, 5$). A 61×81 mesh, shown in Fig. 3.20 was generated analytically [44].

$$\begin{cases} x = -(R_x - (R_x - 1)\xi)\cos(\theta(2\eta - 1)) \\ y = \quad (R_y - (R_y - 1)\xi)\sin(\theta(2\eta - 1)) \end{cases} \tag{3.26}$$

where $R_x = 3$, $R_y = 6$, and $\theta = 5\pi/12$. The flow fields is initialized according to the free-stream state, while reflection wall conditions are imposed at the surface of the cylinder. On the two supersonic outflow boundaries, extrapolating method is applied.

The contours of pressure for $M_\infty = 3$ are plotted in Fig. 3.20. Although the pressure jump is very large, the shock wave is captured without spurious oscillations. Fig. 3.21 shows the profile of pressure along the central line, as well as the result computed from

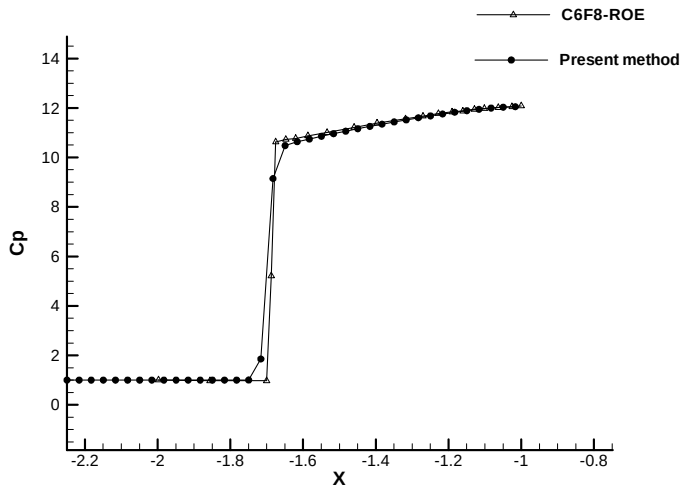

Figure 3.21: Pressure coefficient profile along the central line for Mach 3 flow around a cylinder.

the sixth order compact-Roe scheme with the adaptive filter by Visbal & Gaitonde [45]. Our profile agrees excellently well with theirs. It is noticed that the shock we simulated smears over 3 cells and our profile is not as sharp as the reference data. It is because that the grid used by Visbal & Gaitonde is finer (101×81 in the upper-half domain) than the present work and their sixth order scheme is much more accurate than the third order MUSCL scheme we used here.

For the higher free-stream Mach number case, simulation of flows around blunt bodies with Roe scheme may produce the so-called "carbuncle" [46] which ruins the bow shock waves. Some special treatments such as entropy fixing are needed to fix this problem. The "carbuncle" phenomenon doses not occur in our LBM simulation (Fig. 3.22). The pressure profile and reference data [45] are shown in Fig. 3.23. The location of the shock wave agrees excellently well with the semi-empirical value [47] and the result by Visbal & Gaitonde. Because a much finer grid (201×121 in the upper-half domain) was used in [45], our shock wave is much more smeared than the data extracted from [45].

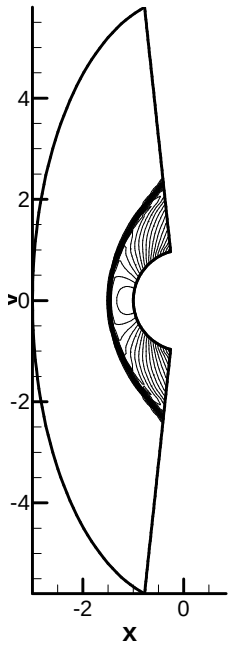

Figure 3.22: Pressure contour of Mach 5 flow around a cylinder.

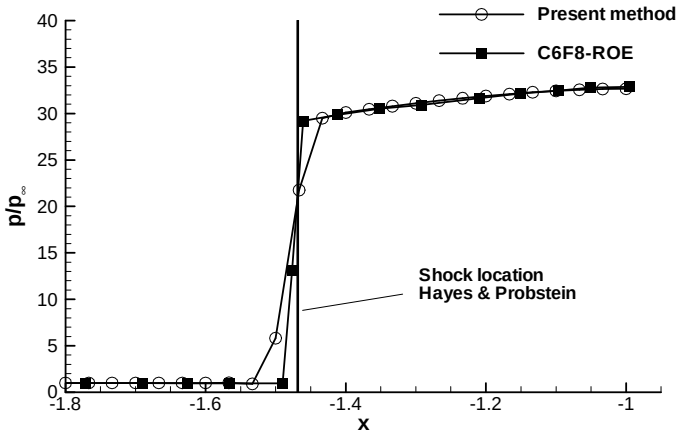

Figure 3.23: Pressure coefficient profile along the central line for Mach 5 flow around a cylinder.

3.7 Concluding remarks

In this chapter, the new deriving method proposed in the last chapter is extended to develop LB models for inviscid compressible flows. A simple function, CF-ICF, is constructed to replace the Maxwellian function as the equilibrium density distribution function. CF-ICF satisfies all statistical relations needed to recover Euler equations. CF-ICF is assigned onto the lattice velocity directions by the Lagrangian interpolating stencil to obtain the equilibrium distribution functions and the associated lattice velocity models. Numerical experiments showed that compressible inviscid flows with strong shock waves can be excellently simulated by the present model.

We can see that, in the derivation of the method, there is no need to assume a formula of f_i^{eq} beforehand and they are naturally derived step by step. Although the number of particle distribution functions of our models is more than previous models, the numerical tests show that they have only one free parameter e_0 and can simulate flows with very high Mach number and strong shock waves. And, the most important, the deriving method proposed in this work is easy to implement and needs fewer assumption. It gives a new way to develop lattice models for compressible inviscid flows. Although we only present one and two dimensional models for inviscid compressible flows, it is natural to extend this deriving method to three dimensional space (just use a sphere shell function and a three dimensional Lagrange interpolating stencil).

Different from Gaussian-Hermite expansion method which is adopted to discretize the Maxwellian function to derive the classic LB models, interpolating approach is applied to discretize CF-ICF in our deriving method. So the idea of assigning functions make the slip wall boundary condition easy to implement.

On the other hand, we have to indicate that the collision term on the right hand side of DVBE is a stiff nonlinear source term. This makes CFL number very small since only a full explicit scheme without any acceleration method is used in this work. So the simulations are very time consuming, hundreds times slower than solving Euler equations with FVM. Maybe some form of implicit methods and acceleration measures (such as

multigrid) should be applied to increase CFL number.

Chapter 4

Development of LB Models for Viscous Compressible Flows

The feasibility and advantages of our derivation method have been demonstrated in the last two chapters. In this chapter, a two dimensional LB model for viscous compressible flows is derived. For viscous compressible flows, heat conduction and the work done by viscous stress should be recovered in the energy equation. Again, we will try to construct another circular function. Its moment integrals have to recover the right energy equation. Then, by using a set of Lagrangian polynomials to discretize it, a LB model for viscous compressible flows, D2Q17L2, is obtained. With this model, some test cases are simulated. Promising results are produced. On the other hand, we have to indicate that, the pressure fields have some errors because of the nonzero bulk viscosity and the unit Prandtl number which come from the BGK collision model.

4.1 Simple equilibrium distribution function for viscous compressible flows

In Chapter 1, we have pointed out that for compressible flows, considering the extra degrees of freedom of gas molecules, the Maxwellian function should be Equ. (1.14)

$$g\left(\boldsymbol{\xi}\right)=\rho\left(2\pi RT\right)^{-\frac{K+D}{2}}\exp\left[-\frac{\left(\boldsymbol{\xi}-\mathbf{u}\right)^{2}+\sum_{i=1}^{K}\xi_{i}^{2}}{2RT}\right]$$

where D is space dimension, $\boldsymbol{\xi}_i, i=1\ldots K$ are extra degrees of freedom, $K=b-D$ [22]. Equation (1.14) satisfies the seven relations of Equ. (4.1) to recover compressible Navier-Stokes equations.

$$\int gd\boldsymbol{\xi} = \rho \tag{4.1a}$$

$$\int g\xi_{\alpha}d\boldsymbol{\xi} = \rho u_{\alpha} \tag{4.1b}$$

$$\int\frac{1}{2}g\xi_{\chi}^{2}d\boldsymbol{\xi} = \rho\left(\frac{1}{2}u_{\chi}^{2}+\frac{b}{2}RT\right) \tag{4.1c}$$

$$\int g\xi_{\alpha}\xi_{\beta}d\boldsymbol{\xi} = \rho\left(u_{\alpha}u_{\beta}+RT\delta_{\alpha\beta}\right) \tag{4.1d}$$

$$\int\frac{1}{2}g\xi_{\chi}^{2}\xi_{\beta}d\boldsymbol{\xi} = \rho\left(\frac{1}{2}u_{\chi}^{2}+\frac{b+2}{2}RT\right)u_{\beta} \tag{4.1e}$$

$$\int g\xi_{\alpha}\xi_{\beta}\xi_{\phi}d\boldsymbol{\xi} = p\left(u_{\alpha}\delta_{\beta\phi}+u_{\beta}\delta_{\phi\alpha}+u_{\phi}\delta_{\alpha\beta}\right)+\rho u_{\alpha}u_{\beta}u_{\phi} \tag{4.1f}$$

$$\begin{aligned}\int\frac{1}{2}g\xi_{\chi}^{2}\xi_{\alpha}\xi_{\beta}d\boldsymbol{\xi} &= \rho RT\left\{\underline{\frac{b+2}{2}RT\delta_{\alpha\beta}}+\frac{1}{2}\left[(b+4)u_{\alpha}u_{\beta}+u_{\chi}^{2}\delta_{\alpha\beta}\right]\right\}\\ &\quad+\frac{1}{2}\rho u_{\chi}^{2}u_{\alpha}u_{\beta}\end{aligned} \tag{4.1g}$$

where $\int\boldsymbol{\xi}=\int_{-\infty}^{+\infty}\cdots\int_{-\infty}^{+\infty}d\xi_{X_1}\cdots d\xi_{X_D}d\xi_1\cdots d\xi_K$ means a integral in the whole velocity space (including extra degrees of freedom) and b is related to specific heat ratio γ as $\gamma=\dfrac{b+2}{b}$.

Starting from CF-ICF, it is easy to obtain following relations

$$\int g d\boldsymbol{\xi} d\lambda = \rho \tag{4.2a}$$

$$\int g \xi_\alpha d\boldsymbol{\xi} d\lambda = \rho u_\alpha \tag{4.2b}$$

$$\int g \left(\frac{1}{2}\xi_\chi^2 + \lambda\right) d\boldsymbol{\xi} d\lambda = \rho \left(\frac{1}{2}u_\chi^2 + \frac{b}{2}RT\right) \tag{4.2c}$$

$$\int g \xi_\chi \xi_\beta d\boldsymbol{\xi} d\lambda = \rho u_\alpha u_\beta + p\delta_{\alpha\beta} \tag{4.2d}$$

$$\int g \left(\frac{1}{2}\xi_\chi^2 + \lambda\right) \xi_\beta d\boldsymbol{\xi} d\lambda = \rho \left(\frac{1}{2}u_\chi^2 + \frac{b+2}{2}RT\right) u_\beta \tag{4.2e}$$

$$\int g \xi_\alpha \xi_\beta \xi_\chi d\boldsymbol{\xi} d\lambda = p\left(u_\alpha \delta_{\beta\chi} + u_\beta \delta_{\chi\alpha} + u_\chi \delta_{\alpha\beta}\right) + \rho u_\alpha u_\beta u_\chi \tag{4.2f}$$

$$\begin{aligned}\int \frac{1}{2} g \left(\xi_\chi^2 + 2\lambda\right) \xi_\alpha \xi_\beta d\boldsymbol{\xi} d\lambda &= \rho RT \left\{ \underline{\frac{1}{2} bRT\delta_{\alpha\beta}} + \frac{1}{2}\left[(b+4)\, u_\alpha u_\beta + u_\chi^2 \delta_{\alpha\beta}\right] \right\} \\ &\quad + \frac{1}{2}\rho u_\chi^2 u_\alpha u_\beta \end{aligned} \tag{4.2g}$$

where $\int g(\boldsymbol{\xi}, \lambda) d\boldsymbol{\xi} d\lambda = \int_0^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(\xi_x, \xi_y, \lambda) d\xi_x d\xi_y d\lambda$. Compared with Equ. (4.1), only the underlined term in Equ. (4.2g) is different from that of Equ. (4.1g). So, we may modify this circle function to make up this small difference.

How to modify CF-ICF? Before answering this question, we should check the role of the underlined term in recovering macroscopic equations. It can be found by means of Chapman-Enskog analysis.

4.1.1 Chapman-Enskog analysis

In this section, instead of presenting the complete derivation procedure, we will only show how to derive the parts of non-equilibrium effects in the momentum and energy equations because we have derived the model to recover Euler equations in the last chapter.

We write Equ. (4.1g) and Equ. (4.2g) into a general form

$$\begin{aligned}&\rho RT \left\{ \frac{1}{2} b\theta RT\delta_{\alpha\beta} + \left[\frac{1}{2}(b+4)\, u_\alpha u_\beta + \frac{1}{2}u_\chi^2 \delta_{\alpha\beta}\right] \right\} + \frac{1}{2}\rho u_\chi^2 u_\alpha u_\beta \\ &= \theta C_v p T \delta_{\alpha\beta} + \left[\left(\frac{C_p}{R} + 1\right) u_\alpha u_\beta + \frac{1}{2}u_\chi^2 \delta_{\alpha\beta}\right] p + \frac{1}{2}\rho u_\chi^2 u_\alpha u_\beta \end{aligned} \tag{4.3}$$

where, C_v and C_p are the specific heats at constant volume and pressure, γ is the specific heat ratio, and R is the gas constant. And θ is a switch parameter, $\theta = \gamma$ for the Maxwellian function and $\theta = 1$ for CF-ICF.

Momentum equation

With the expansion

$$f = f^{(0)} + \varepsilon f^{(1)} + O(\varepsilon^2)$$

the viscous stress is expressed as

$$\begin{aligned} \mathbf{\Pi} &= \int f^{(1)} \xi_\alpha \xi_\beta d\boldsymbol{\xi} \\ &= -\tau \int \left(\partial_t f^{(0)} + \xi_\alpha \partial_\alpha f^{(0)} \right) \xi_\beta \xi_\gamma d\boldsymbol{\xi} \end{aligned} \tag{4.4}$$

The first term in Equ. (4.4) can be converted to the expression of mean flow variables

$$\begin{aligned} \int \partial_t f_i^{(0)} \xi_\beta \xi_\gamma d\boldsymbol{\xi} &= \partial_t \left(\rho \mathbf{v}\mathbf{v} + p\mathbf{I} \right) \\ &= \partial_t \left(\rho \mathbf{v} \right) \mathbf{v} + \rho \mathbf{v} \partial_t \mathbf{v} + \left(\gamma - 1 \right) \partial_t \left(\rho e \right) \mathbf{I} \end{aligned} \tag{4.5}$$

Substituting the momentum and energy equations of Euler equations into Equ.(4.5), we can get

$$\int \partial_t f_i^{(0)} \xi_\beta \xi_\gamma d\boldsymbol{\xi} = -\nabla \cdot \left(\rho \mathbf{v}\mathbf{v}\mathbf{v} \right) - \left(\mathbf{v} \nabla p + \nabla p \mathbf{v} \right) - \nabla \cdot \left(p\mathbf{v} \right) \mathbf{I} - \left(\gamma - 1 \right) p \left(\nabla \cdot \mathbf{v} \right) \mathbf{I} \tag{4.6}$$

The second term in Equ. (4.4) can be converted as

$$\begin{aligned} & \int \xi_\alpha \partial_\alpha f^{(0)} \xi_\beta \xi_\gamma d\boldsymbol{\xi} \\ = \; & \partial_\alpha \left[p \left(u_\alpha \delta_{\beta\chi} + u_\beta \delta_{\chi\alpha} + u_\chi \delta_{\alpha\beta} \right) \right] + \partial_\alpha \rho u_\alpha u_\beta u_\chi \\ = \; & \nabla \cdot \left(\rho \mathbf{v}\mathbf{v}\mathbf{v} \right) + \nabla \cdot \left(p\mathbf{v} \right) \mathbf{I} + p \left(\nabla \mathbf{v} + \nabla^T \mathbf{v} \right) + \mathbf{v} \nabla p + \nabla p \mathbf{v} \end{aligned} \tag{4.7}$$

Combining Equ. (4.6) and Equ. (4.7), we can get the viscous stress as

$$
\begin{aligned}
\mathbf{\Pi} &= -\tau \int \left(\partial_t f_i^{(0)} + \xi_\alpha \partial_\alpha f_i^{(0)}\right) \xi_\beta \xi_\gamma d\xi_x d\xi_y \\
&= -\tau \left[p\left(\nabla \mathbf{v} + \nabla^T \mathbf{v}\right) + (\gamma - 1)\, p\, (\nabla \cdot \mathbf{v})\, \mathbf{I}\right] \\
&= -\tau p \left(\nabla \mathbf{v} + \nabla^T \mathbf{v} - \frac{2}{3}(\nabla \cdot \mathbf{v})\mathbf{I}\right) - \left[\frac{2}{3} - (\gamma - 1)\right] \tau p\, (\nabla \cdot \mathbf{v})\, \mathbf{I}
\end{aligned}
$$

So, the momentum equation is

$$
\partial_t (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v}\mathbf{v} + p\mathbf{I}) = \nabla \cdot \left[\mu \left(\nabla \mathbf{v} + \nabla^T \mathbf{v} - \frac{2}{3}(\nabla \cdot \mathbf{v})\mathbf{I}\right) + \mu_b (\nabla \cdot \mathbf{v})\mathbf{I}\right]
$$

where

$$
\begin{aligned}
\mu &= \tau p \\
\mu_b &= \left(\frac{5}{3} - \gamma\right) \mu
\end{aligned}
$$

We notice that the bulk viscosity $\mu_b = (\frac{5}{3} - \gamma)\mu$. The bulk viscosity vanishes only when $\gamma = \frac{5}{3}$ corresponding to monatomic gas. For diatomic or polyatomic gas, the nonzero bulk viscosity is the same order of the shear viscosity μ. For most cases, μ_b is usually much smaller than μ and can be neglected. However, for some cases, the none zero μ_b might be a problem for simulations of compressible viscous flows, which will be shown later.

Energy equation

The energy of a particle can be written as

$$
\zeta = \begin{cases} \dfrac{1}{2}\boldsymbol{\xi}^2 & \text{for the Maxwellian function} \\ \\ \dfrac{1}{2}\boldsymbol{\xi}^2 + \lambda & \text{for CF} - \text{ICF} \end{cases} \tag{4.8}
$$

The non-equilibrium parts of heat flux is

$$\begin{aligned}\mathbf{q} &= \int f^{(1)}\zeta\xi_\alpha d\boldsymbol{\xi} \\ &= -\tau\int\left(\partial_t f^{(0)}\zeta\xi_\alpha + \xi_\beta\partial_\beta f^{(0)}\zeta\xi_\alpha\right)d\boldsymbol{\xi} \end{aligned} \tag{4.9}$$

The first term in Equ. (4.9) can be written as

$$\begin{aligned}&\partial_t\int f_i^{(0)}\zeta\xi_\alpha d\boldsymbol{\xi} \\ =\ &\partial_t\left[\left(\rho E+p\right)u_\alpha\right] \\ =\ &\left[\partial_t\left(\rho E\right)u_\alpha+\rho E\partial_t u_\alpha\right]+\left[u_\alpha\partial_t p+p\partial_t u_\alpha\right] \\ =\ &-\nabla\cdot\left(\rho E\mathbf{v}\mathbf{v}\right)-E\nabla p-2\nabla\cdot\left(p\mathbf{v}\right)\mathbf{v}-\left(\gamma-1\right)p\mathbf{v}\nabla\cdot\mathbf{v} \\ &-p\mathbf{v}\cdot\nabla\mathbf{v}-\left(\gamma-1\right)e\nabla p\end{aligned} \tag{4.10}$$

The second term can be expressed as

$$\begin{aligned}&\int\zeta\partial_\beta f_i^{(0)}\xi_\alpha\xi_\beta d\boldsymbol{\xi} \\ =\ &\partial_\beta\left\{\theta pC_vT\delta_{\alpha\beta}+\left[\left(\rho C_pT+p\right)u_\alpha u_\beta+\frac{1}{2}pu_\chi^2\delta_{\alpha\beta}\right]+\frac{1}{2}\rho u_\chi^2u_\alpha u_\beta\right\} \\ =\ &\partial_\beta\left\{p\left(\theta C_vT+\frac{1}{2}u_\chi^2\right)\delta_{\alpha\beta}+\rho\left(C_vT+\frac{1}{2}u_\chi^2\right)u_\alpha u_\beta+2pu_\alpha u_\beta\right\}\end{aligned}$$

Set $E'=\theta C_vT+\frac{1}{2}u_\chi^2$, this expression can be written as

$$\begin{aligned}&\int\zeta\partial_\beta f_i^{(0)}\xi_\alpha\xi_\beta d\boldsymbol{\xi} \\ =\ &\nabla\cdot\left(pE'\mathbf{I}\right)+\nabla\cdot\left(\rho E\mathbf{v}\mathbf{v}\right)+2\nabla\cdot\left(p\mathbf{v}\mathbf{v}\right) \\ =\ &\nabla\left(pE'\right)+\nabla\cdot\left(\rho E\mathbf{v}\mathbf{v}\right)+2\nabla\cdot\left(p\mathbf{v}\right)\mathbf{v}+p\mathbf{v}\cdot\left(\nabla^T\mathbf{v}+\nabla\mathbf{v}\right)\end{aligned} \tag{4.11}$$

So, by combining Equ. (4.10 and 4.11), we can get

$$\mathbf{q} = -\tau \left[p\theta \nabla e + (\theta - \gamma)\, e\tau \nabla p \right] - \mathbf{\Pi} \cdot \mathbf{v}$$

The energy equation is

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho E \mathbf{v} + p\mathbf{v}) = \nabla \cdot (\mathbf{\Pi} \cdot \mathbf{v}) + \nabla \cdot (k \nabla T) + \nabla \cdot \left[(\theta - \gamma)\, e\tau \nabla p \right]$$

where heat conductivity $k = \theta \tau p C_v$. And there is an extra term $\nabla \cdot [(\theta - \gamma) e\tau \nabla p]$ which can be cancelled by setting $\theta = \gamma$. This means that Equ. (4.14g) should be exactly satisfied.

4.1.2 The circular function for viscous compressible flows

From the above derivation, we can see that θ only affects heat conductivity and the extra term in energy equation. And the transportation coefficients, such as the heat conductivity and the viscosity, are functions of p and τ.

In kinetic theory, pressure can be expressed as

$$p = \frac{1}{D} \int g_m (\boldsymbol{\xi} - \mathbf{u})^2 d\boldsymbol{\xi} = \frac{1}{D} \int g_m \mathbf{C}^2 d\mathbf{C}$$

where D is the spatial dimension, $\mathbf{C}$ is the peculiar velocity and g_m is the Maxwellian function. Define the mean peculiar velocity as

$$\bar{C} = \sqrt{\frac{1}{\rho} \int g_m \mathbf{C}^2 d\mathbf{C}}$$

Then we have $p = \frac{1}{D} \rho \bar{C}^2$. Thus μ and k can be expressed as

$$\begin{aligned} \mu &= \frac{1}{2} \tau \rho\, \bar{C}^2 \\ k &= \frac{1}{2} \tau \rho \gamma\, \bar{C}^2\, C_v \end{aligned}$$

So, viscosity and heat conductivity are effects of the peculiar velocity of gas molecules. Obviously, the bigger $\bar{C}$, the larger these two transportation coefficients. Set $r_m = \bar{C}$ and $r_t = \sqrt{\gamma}\bar{C}$, we can write viscosity and heat conductivity as

$$\begin{aligned} \mu &= \frac{1}{2}\tau\rho r_m^2 \\ k &= \frac{1}{2}\tau\rho r_t^2 C_v \end{aligned}$$

Here, we can regard r_m and r_t as the rates of diffusion of momentum and thermal (internal) energy, respectively.

For CF-ICF, $p = \frac{1}{D}\rho c^2$ where c is the radius of the circle. Thus,

$$\begin{aligned} \mu &= \frac{1}{2}\tau\rho c^2 \\ k &= \frac{1}{2}\tau\rho c^2 C_v \end{aligned}$$

This means that, for $\mathrm{CF-ICF}$, $r_t = r_m = c$, momentum and thermal energy diffuse at the same rate c, which contradicts the Maxwellian function. Thus, we should modify CF-ICF so that $r_t = \sqrt{\gamma} r_m$.

Intuitively, the reason why CF-ICF can not lead to the right energy dissipation term should be that both density and internal energy concentrate on the same circle of $r = c$, which makes momentum and thermal energy have the same rate of diffusion. So, if we can make density and thermal energy distribute differently, we might get different r_m and r_t.

We construct a modified circular function as

$$g(\xi, \lambda) = \begin{cases} \dfrac{\rho_c}{2\pi r} & \text{if } \|\boldsymbol{\xi} - \mathbf{u}\| = r \text{ and} \lambda = \lambda_c \\ \\ \rho_p & \text{if } \xi = \mathbf{u} \text{ and } \lambda = 0 \end{cases} \tag{4.12}$$

Different from CF-ICF, the density are divided into two parts. The first part, ρ_c, concen-

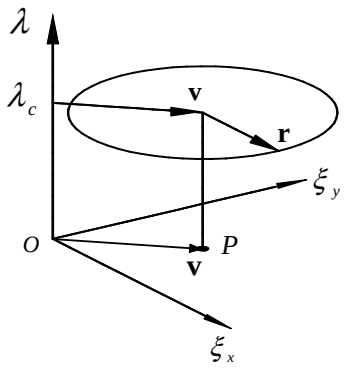

Figure 4.1: The modified circle function.

trates on a circle of radius r with rest energy λ_c. And the other part, ρ_p, is located at the projection of the circle center (point P) on the $\xi_x - \xi_y$ plane. So all internal energy is distributed on the circle which hosts only a fraction of density (Fig. 4.1). ρ_c, ρ_p, r and λ_c should satisfy

$$\rho = \rho_c + \rho_p \tag{4.13a}$$

$$\rho e = \rho_c(\frac{1}{2}r^2 + \lambda_c) \tag{4.13b}$$

$$p = (\gamma - 1)\rho e = \frac{1}{2}\rho_c r^2 \tag{4.13c}$$

which means that the density, internal energy and pressure remain invariant. Because ρ_c is uniformly distributed along the circle and the projection of the its center on the $\xi_x - \xi_y$ plane is the mean flow velocity $\mathbf{v}$, the momentum is invariant as long as $\rho = \rho_c + \rho_p$. From Equ. (4.13), we can obtain ρ_p, r and λ_c as functions of ρ_c

$$\rho_p = \rho - \rho_c, \qquad r = \sqrt{\frac{2(\gamma-1)\rho e}{\rho_c}} \qquad \text{and} \quad \lambda_c = \frac{(2-\gamma)\rho e}{\rho_c} \tag{4.14}$$

It is easy to get the moment integrals of this modified circular function as

$$\int g d\boldsymbol{\xi} d\lambda = \rho \tag{4.15a}$$

$$\int g \xi_\alpha d\boldsymbol{\xi} d\lambda = \rho u_\alpha \tag{4.15b}$$

$$\int g \left(\frac{1}{2}\xi_\chi^2 + \lambda\right) d\boldsymbol{\xi} d\lambda = \rho \left(\frac{1}{2}u_\chi^2 + \frac{b}{2}RT\right) \tag{4.15c}$$

$$\int g \xi_\alpha \xi_\beta d\boldsymbol{\xi} d\lambda = \rho u_\alpha u_\beta + p\delta_{\alpha\beta} \tag{4.15d}$$

$$\int g \left(\frac{1}{2}\xi_\chi^2 + \lambda\right) \xi_\alpha d\boldsymbol{\xi} d\lambda = \rho \left(\frac{1}{2}u_\chi^2 + \frac{b+2}{2}RT\right) u_\alpha \tag{4.15e}$$

$$\int g \xi_\alpha \xi_\beta \xi_\chi d\boldsymbol{\xi} d\lambda = \rho u_\alpha u_\beta u_\chi + \frac{1}{2}\rho c^2 \left(u_\alpha \delta_{\beta\chi} + u_\beta \delta_{\chi\alpha} + u_\chi \delta_{\alpha\beta}\right) \tag{4.15f}$$

$$\int \frac{1}{2} g \left(\xi_\chi^2 + 2\lambda\right) \xi_\alpha \xi_\beta d\boldsymbol{\xi} d\lambda = \rho RT \left\{ \underline{\frac{\rho}{\rho_c} bRT\delta_{\alpha\beta}} + \frac{1}{2}\left[(b+4)\, u_\alpha u_\beta + u_\chi^2 \delta_{\alpha\beta}\right] \right\}$$
$$+ \frac{1}{2}\rho u_\chi^2 u_\alpha u_\beta \tag{4.15g}$$

By setting $\rho_c = \dfrac{\rho}{\gamma}$, Equ. (4.15g) is the same as Equ. (4.1). Now, the seven moment integrals of the modified circular function are the same as those of the Maxwellian function. Thus they can make the BGK-Boltzmann equation recover to Navier-Stokes equations as

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{4.16a}$$

$$\partial_t (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v}\mathbf{v} + p\mathbf{I}) = \nabla \cdot \boldsymbol{\Pi} \tag{4.16b}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho E \mathbf{v} + p\mathbf{v}) = \nabla \cdot (\boldsymbol{\Pi} \cdot \mathbf{v}) + \nabla \cdot (k \nabla T) \tag{4.16c}$$

where

$$\begin{aligned}
\mathbf{\Pi} &= \left[\mu\left(\nabla\mathbf{v}+\nabla^T\mathbf{v}-\frac{2}{3}\nabla\cdot\mathbf{v}\mathbf{I}\right)+\mu_b\nabla\cdot\mathbf{v}\mathbf{I}\right] \\
\mu &= \tau p \\
\mu_b &= \left(\frac{5}{3}-\gamma\right)\mu \\
k &= \tau p C_p = \frac{\gamma}{\gamma-1}\tau p R
\end{aligned}$$

Now, we have the Circular Function for Viscous Compressible Flows (CF-VCF).

It should be noticed that the viscosity and the heat conductivity are not independent (making Prandtl number fixed as 1), also the bulk viscosity μ_b is non-zero. These two model errors are resulted from the BGK collision model since it has only one parameter τ to adjust all the three transport coefficients. Because the Prandtl number can only be unit, the model can not be applied to do thermal simulation of viscous flows with non-unit Prandtl number. But when Mach number is not too high (in this case, aerothermal effects are not important), aerodynamic forces are not sensitive on temperature. So we can hope to obtain reasonable aerodynamics forces when Mach number is not very high.

4.2 Assigning functions and lattice model

With the experience of developing LB models for inviscid compressible flows in the last chapter, constructing assigning functions and lattice model for viscous compressible flows is straightforward. Integrals in Equ. (4.1) are moments of g in $\boldsymbol{\xi}$ (up to the forth order) and λ (the first order). So, in order to conserve these moments, a Lagrangian interpolating stencil of the forth order in $\boldsymbol{\xi}$ and the first order in λ is needed.

A forth order 17-node Lagrangian interpolating stencil in velocity space was first constructed (shown in Fig. 4.2). The 17 velocity vectors are listed in Tab. 4.1. There is no need to set all components of lattice velocity as integer since we use FVM to solve DVBE. With the interpolating polynomials of above stencil $\phi_i(\boldsymbol{\xi})$, coupled with two energy levels,

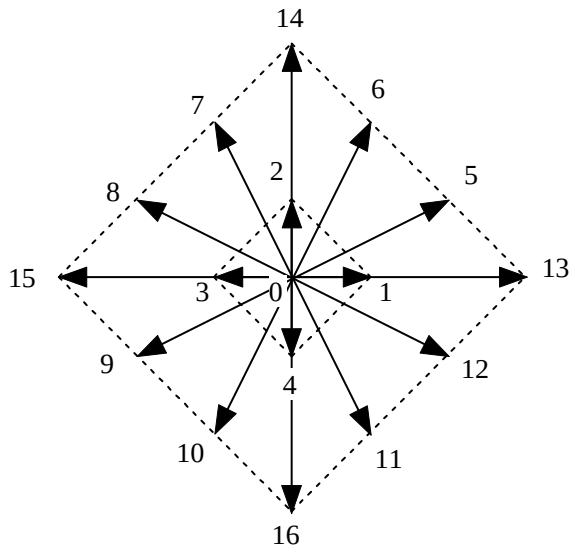

Figure 4.2: D2Q17 lattice.

$\lambda_1 = 0$ and $\lambda_2 > \lambda_c$, the new LB model for viscous compressible flows can be developed as

$$f_{i1}^{eq} = \int_0^{2\pi} \frac{\rho_c}{2\pi}\phi_i(u + r_c\cos\theta, v + r_c\sin\theta)\frac{\lambda_2 - \lambda_c}{\lambda_2}d\theta + \rho_p\phi_i(u,v) \tag{4.17a}$$

$$f_{i2}^{eq} = \int_0^{2\pi} \frac{\rho_c}{2\pi}\phi_i(u + r_c\cos\theta, v + r_c\sin\theta)\frac{\lambda_c}{\lambda_2}d\theta \tag{4.17b}$$

After scaling, $\breve{\lambda}_2 = 1$. Since the expressions of f_{iv}^{eq} are very tedious, they are not listed here. It is easy to run the Maple script in Appendix A.4 to obtain them.

$\mathbf{e}_0 = (0,0)$	for $i = 0$		
$\mathbf{e}_i = \frac{2}{3}(\cos\frac{i\pi}{2}, \sin\frac{i\pi}{2})$	for $i = 1\ldots4$		
$\mathbf{e}_i = (\cos\frac{i\pi}{2}, \sin\frac{i\pi}{2})$	for $i = 13\ldots16$		
$\mathbf{e}_5 = \left(\frac{4}{3}, \frac{2}{3}\right)$	$\mathbf{e}_7 = \left(-\frac{2}{3}, \frac{4}{3}\right)$	$\mathbf{e}_9 = \left(-\frac{4}{3}, -\frac{2}{3}\right)$	$\mathbf{e}_{11} = \left(\frac{2}{3}, -\frac{4}{3}\right)$
$\mathbf{e}_6 = \left(\frac{2}{3}, \frac{4}{3}\right)$	$\mathbf{e}_8 = \left(-\frac{4}{3}, \frac{2}{3}\right)$	$\mathbf{e}_{10} = \left(-\frac{2}{3}, -\frac{4}{3}\right)$	$\mathbf{e}_{12} = \left(\frac{4}{3}, -\frac{2}{3}\right)$

Table 4.1: Velocity vectors of D2Q17L2.

4.3 Boundary conditions

For viscous compressible flows, implementation of the non-slip wall boundary condition is the most important while other boundary conditions can be implemented in the same way as for inviscid flows described in the last chapter. So, in this section, only the non-slip wall boundary condition is discussed.

In the traditional LBM, the non-slip wall boundary condition is usually implemented with the bounce-back rule. It can guarantee the non-slip condition on a wall. For LBM on curvilinear grids or unstructured meshes, the bounce-back method is still feasible. But for multi-velocity thermal models, the bounce-back method can only be used to implement the non-slip and adiabatic wall condition. For other wall conditions such as walls with specified temperature or heat flux, special treatment is needed.

Recently, Guo *et al.* [37] proposed a method based on the extrapolation of the nonequilibrium part of distribution functions [36]. This method is of second-order accuracy. Assume that $\mathbf{x}_b$ is a boundary node, and $\mathbf{x}_f$ is the nearest neighboring fluid node of $\mathbf{x}_b$. Then, the distribution function at $\mathbf{x}_b$ is set to be

$$f_i(\mathbf{x}_b) = f_i^{eq}(\mathbf{x}_b) + [f_i(\mathbf{x}_f) - f_i^{eq}(\mathbf{x}_f)] \tag{4.18}$$

where the second part in the bracket on the right hand side is the nonequilibrium part of the distribution function at $\mathbf{x}_f$, which is used to approximate the value at node $\mathbf{x}_b$. $f_i^{eq}(\mathbf{x}_b)$ can be computed with the mean flow variables (ρ_b, $\mathbf{v}_b$ and e_b) which can be determined with those methods in the traditional CFD. In the kinetic theory, viscous stress, energy diffusion and dissipation come from the nonequlibrium part of distribution function. So, in this nonequilibrium extrapolation method, both of the mean flow variables and the microscopic f_i are consistent.

In our simulations, the mean flow variables on the wall (ρ_b and e_b) can be determined as follows

1. According to the pressure condition on the wall $\left.\dfrac{\partial p}{\partial n}\right|_b = 0$, extrapolate pressure p_b

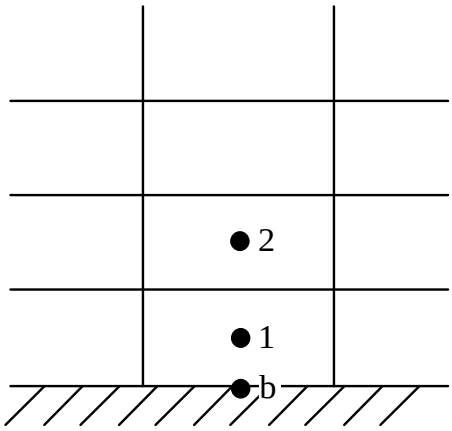

Figure 4.3: Schematic of wall boundary condition.

at the wall node from the fluid nodes near $\mathbf{x}_b$ with the second order or even higher order method. For structural grids, if the grid is orthogonal and smooth near the wall (Fig. 4.3), p_b can be written as

$$p_b = \frac{9p_1 - p_2}{8}$$

2. Determine the internal energy e_b at the wall node. If the wall is isothermal, e_b is a constant specified. If the heat flux q_b on the wall is specified, it means that the normal gradient of internal energy, $\left.\frac{\partial e}{\partial n}\right|_b$, is constant. Then e_b can also be extrapolated from the fluid nodes near $\mathbf{x}_b$ like computing p_b

$$e_b = \frac{9e_1 - e_2}{8} - \frac{3}{8}\left.\frac{\partial e}{\partial n}\right|_b$$

3. Compute ρ_b from p_b and e_b according to the equation of state $p = (\gamma - 1)\rho e$

4.4 Solution procedure and parallel computing

Because the governing equation is still DVBE and only the model is different, the solution procedure is almost the same as that for simulation of inviscid compressible flows introduced in the last chapter except for the following differences:

1. The grid should be fine enough to capture the shear layers in viscous flows.

2. Non-slip wall boundary condition. It has been discussed in the last section.

3. $\breve{\tau}$ is not constant any longer. It is determined according to Equ. (4.16).

4. $\Delta\breve{t}$ should be smaller enough than $\breve{\tau}$ to avoid obvious numerical dissipation.In our simulations, we set $\Delta\breve{t} = 0.1\breve{\tau}_m$, where $\breve{\tau}_m$ is the minimum value in the field.

For viscous flows, mesh spacings near non-slip walls are very small to resolve boundary layers. Usually, higher Reynolds number case needs smaller mesh spacing near the boundary to capture the thinner boundary layers. And the small mesh spacing limits the time step. So simulating viscous flows is very time consuming. It may take several days for our serial code to simulate the flow of Reynolds number 500 over a NACA0012 airfoil on an Itanium workstation.

In order to accelerate convergence of numerical simulation of viscous flows, parallel computing with shared memory is adopted. By means of domain decomposition, a structural grid is partitioned into several blocks and each block is assigned to a CPU. In order to keep simplicity of the code, every CPU holds a complete grid and arrays. Two levels of ghost cells serve as the internal boundary between the neighboring blocks (Fig. 4.4). At every time step, these internal boundaries are updated through MPI communicating subroutines. The flowchart of the algorithm is shown in Fig. 4.5.

4.5 Numerical tests

We will consider two test problems: Couette flow and laminar compressible flow around NACA0012 airfoil.

4.5.1 Simulation of Couette flow

Couette flow refers to the flow between two planes with constant separation between them, moving relative to one another. It is a good test of numerical methods for viscous heat dissipation [7, 13]. With the top wall moving at the speed of U and the bottom wall fixed,

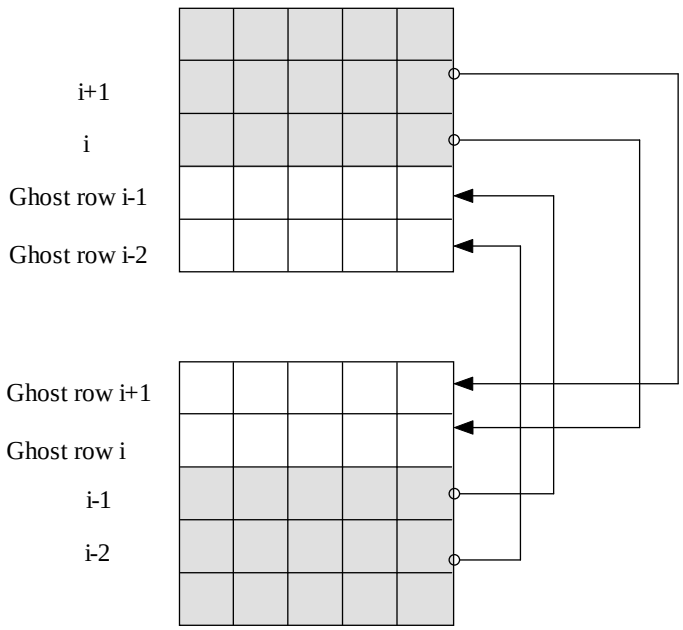

Figure 4.4: Update an internal boundary between two partitions.

the velocity profile is a straight line and the internal energy profile satisfies the following relations [7, 13]

$$\frac{e-e_0}{e_1-e_0} = \frac{y}{H} + \frac{E_c P_r}{2}\frac{y}{H}\left(1-\frac{y}{H}\right) \quad \text{for} \quad e_0 \neq e_1 \tag{4.19a}$$

$$e-e_0 = \frac{P_r U^2}{2\gamma}\frac{y}{H}\left(1-\frac{y}{H}\right) \quad \text{for} \quad e_0 \neq e_1 \tag{4.19b}$$

where e_0 and e_1 are the internal energies at the bottom and top walls, respectively, y is the distance from the bottom wall, H is the height of the channel, Pr is the Prandtl number and $Ec = U^2/\gamma(e_1 - e_0)$ is the Eckert number.

All cases were simulated on the same 6×50 grid. The non-slip wall boundary condition is applied on the bottom and top walls. At the left and right boundaries, periodic condition is implemented. The Reynolds number $Re = \dfrac{UH}{\nu}$ is set as 500.

We simulated several cases of different Ec number. We set $H = 1$, $\gamma = 1.4$, $e_0 = 1$, $U = 2$ and $Pr = 1$ which is fixed for our model. By adjusting e_1, Ec can be changed.

Figure 4.6 shows the internal energy profiles (solid dots) and the exact solutions (solid

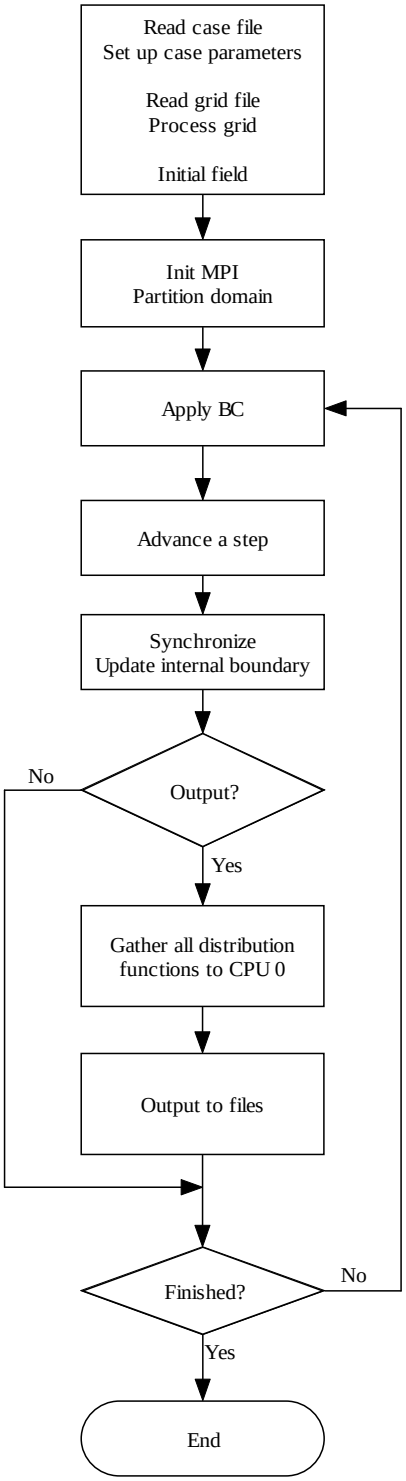

Figure 4.5: Flowchart of the parallel code.

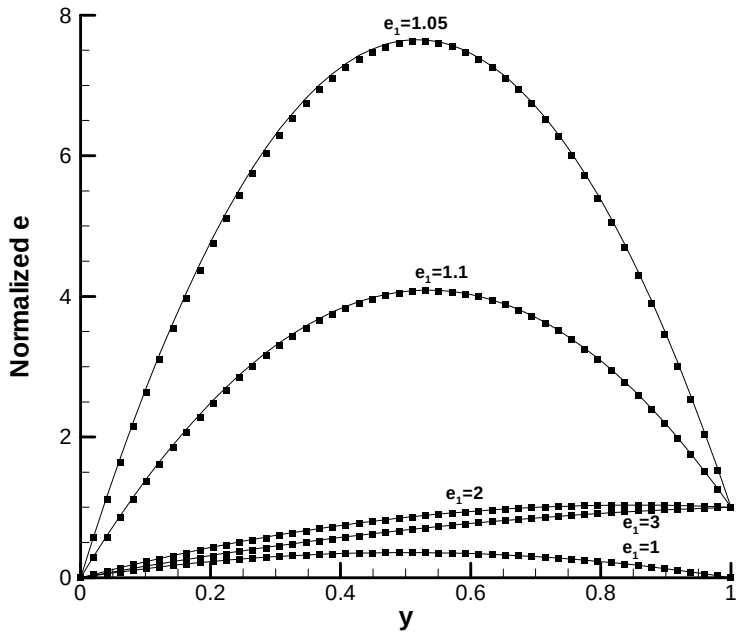

Figure 4.6: Internal energy profiles in Couette flow.

lines) for e_1 = 1, 1.05, 1.1, 2 ,3 which correspond to Ec = 1, 28.572 , 14.286, 1.4286, 0.7143. The Mach number of the sliding top wall for these cases are 2.673, 2.608, 2.548, 1.890 and 1.543, respectively. We can see that our numerical results agree excellently well with the exact solution. However, it should be pointed out that the extra bulk viscous stress vanishes in this simple parallel flow case with $\nabla \cdot \mathbf{v} = 0$, which makes our simulated results very accurate. For general compressible fows of $\nabla \cdot \mathbf{v} \neq 0$, the extra bulk viscous stress might affect the results.

4.5.2 Simulation of laminar flows over NACA0012 airfoil

Here, the results of laminar flows over NACA0012 by our D2Q17L2 model coupled with FVM are presented.

Fig. 4.7 shows the schematic of this problem. The chord length of the airfoil is 1. The distance between the leading edge and the outflow boundary is 21 chord lengths. And 17 chord lengths are used between the leading edge and the free stream boundary. Free-stream conditions are imposed at the outer boundaries (edge FGH), except for the

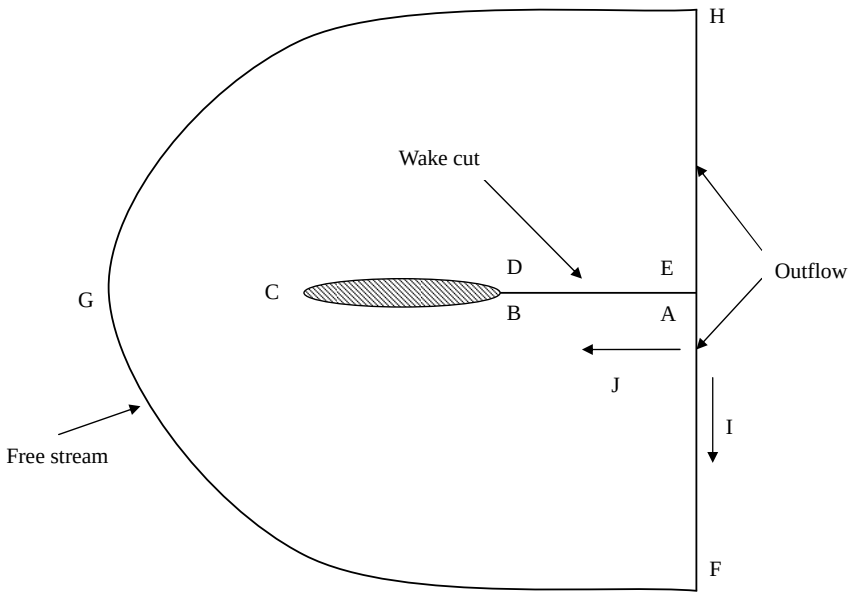

Figure 4.7: Schematic of flow over NACA0012.

two downstream edges, FA and EH where a first-order extrapolation is used for unknown distribution functions. The adiabatic non-slip wall condition is imposed on BCD. A 281×70 C-type grid (Fig. 4.8) was used for all these cases. The wake cute AB is divided into 70 segments. And the wall of the airfoil, BCD, has 140 segments. The minimum grid size normal to the wall boundary is taken as 4×10^{-4} in units of chord length.

Pressure coefficient C_p and skin friction coefficient C_f along the wall are compared with previous studies for quantitative comparison. They are defined as

$$\begin{aligned} C_p &= \frac{p - p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2} \\ C_f &= \frac{f_t}{\frac{1}{2}\rho_\infty U_\infty^2} \end{aligned}$$

Here, the skin friction $\mathbf{f}_t$ at a wall point is determined by projecting viscous stress tensor onto the tangent direction $\mathbf{n}_t$ of the wall point.

$$\mathbf{f}_t = \mathbf{\Pi} \cdot \mathbf{n_t} \tag{4.20}$$

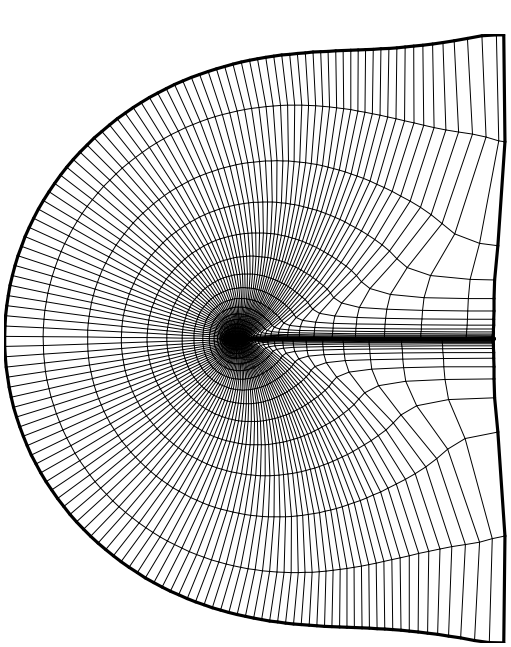

Figure 4.8: Schematic of the grid (shown every 2 grid points).

And for a LB model, the viscous stress tensor can be computed directly without using derivative

$$\mathbf{\Pi} = \sum_{iv} (f_{iv} - f_{iv}^{eq})\, \mathbf{e}_i\, \mathbf{e}_i \tag{4.21}$$

Case 1, $M_\infty = 0.8, Re = 500, \alpha = 10^\circ$

First, we simulate a case of $M_\infty = 0.8$, $Re = 500$, attack angle $\alpha = 10^\circ$. This is the test case A3 of GAMM workshop in 1985 [48]. The free-stream boundary conditions and the initial conditions are $\rho = 1$, $u = 0.8\cos 10^\circ$, $v = 0.8\sin 10^\circ$ and $p = 1/1.4$. And the reference internal energy is 2.5. Dimensionless time step $\Delta\breve{t}$ is 2.0×10^{-4}.

Fig. 4.9 shows the Mach number contours which agree satisfactorily with the result in [49]. In Figs. 4.10 and 4.11, the profile of the skin friction coefficient and pressure coefficient along the airfoil are shown. Also included in these figures are the data in [49]. The upper profiles are the coefficients along the upper surface of the airfoil, and vice versa. The skin friction coefficient C_f agrees with the reference data very well. However, the pressure coefficient C_p of the upper surface deviates the reference data significantly.

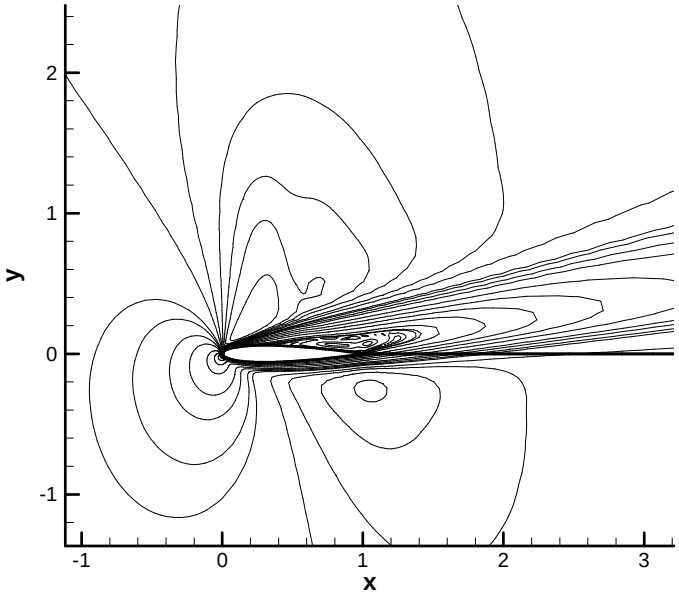

Figure 4.9: Mach number contours around NACA0012 ($\alpha = 10^\circ$, $M_\infty = 0.8$, $Re = 500$).

We think that the non-zero bulk viscosity might make this difference when vortices exist. Detailed discussion will be presented later.

Case 2, $M_\infty = 0.5, Re = 5000, \alpha = 0^\circ$

The second is a more difficult test case. It is the lamina subsonic flow at an angle of attack $\alpha = 0^\circ$, free-stream Mach number $M_\infty = 0.5$, and Reynolds number 5000. This case is near the upper limit of steady laminar flow. The flow separation occurs near the trailing edge which causes the formation of a small recirculation bubble.

Again, we set $e_0 = 2.5$. And initially, a bigger dimensionless time step $\Delta\breve{t} = 1.0 \times 10^{-4}$ was used. After some time steps, the simulation was shifted to a smaller time step $\Delta\breve{t} = 1.0 \times 10^{-5}$ for more accurate computation.

Fig. 4.12 shows the Mach number contour around the airfoil. Although agreeing qualitatively well with the contour in [50], there is a bump after the trailing edge and the wake there is thicker in Fig. 4.12. Shown in Figs.4.14 and 4.13 are the chord-wise distribution of the pressure and skin friction coefficients from the present computations

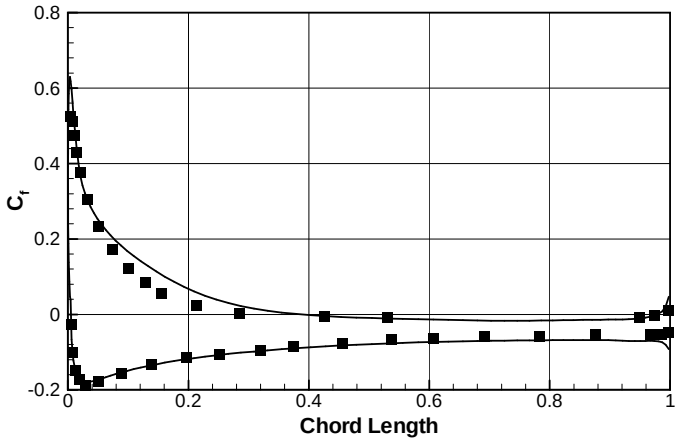

Figure 4.10: Skin friction coefficient for NACA0012 ($\alpha = 10^\circ$, $M_\infty = 0.8$, $Re = 500$).

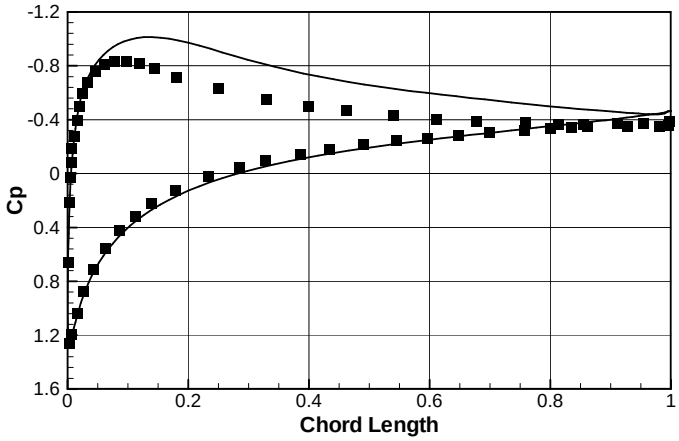

Figure 4.11: Pressure coefficient for NACA0012 ($\alpha = 10^\circ$, $M_\infty = 0.8$, $Re = 500$.

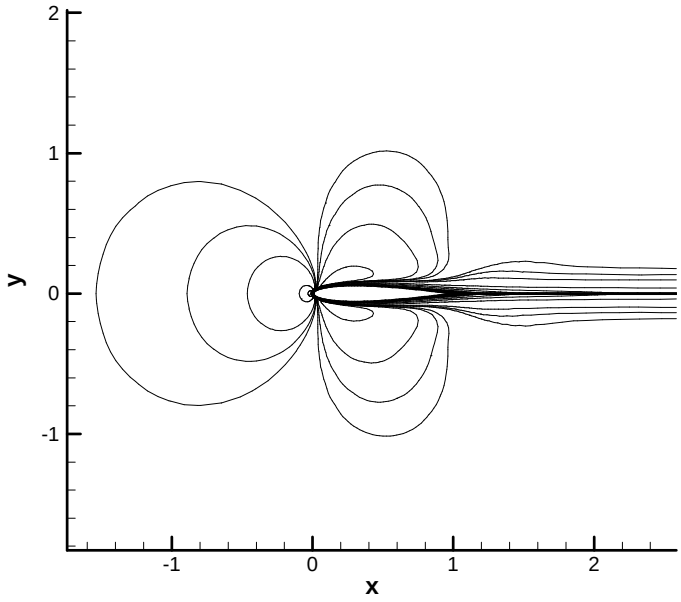

Figure 4.12: Mach number contours around NACA0012 ($\alpha = 0^\circ$, $M_\infty = 0.5$, $Re = 5000$).

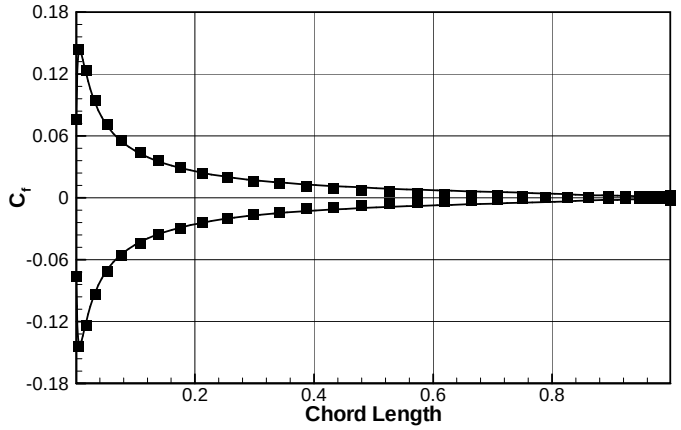

Figure 4.13: Skin friction coefficient for NACA0012 ($\alpha = 0^\circ$, $M_\infty = 0.5$, $Re = 5000$).

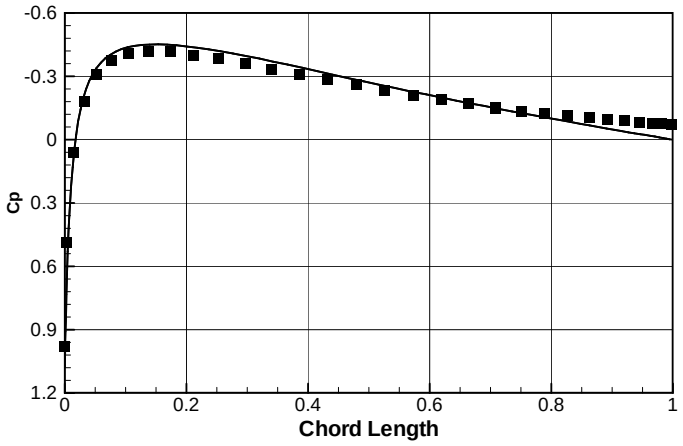

Figure 4.14: Pressure coefficient for NACA0012 ($\alpha = 0^\circ$, $M_\infty = 0.5$, $Re = 5000$).

(shown as lines) and the reference data from [50] (shown as solid squares). Comparing with the profiles in the last case, our two profiles approximate the reference data much better. The skin friction coefficient profile agrees very well with the results computed with discontinuous Galerkin method by Bassi and Rebay in [50]. However, the location of the zero friction point, corresponding to the separation point, is 0.97 which deviates the range $0.810 \sim 0.824$ in [51]. The C_p profile is not as good as the C_f profile. Comparing with the reference data, in the zone near the trailing edge, the pressure is higher, which should be relative to the error of the skin friction coefficient near the trailing edge.

Case 3, $M_\infty = 0.85, Re = 2000, \alpha = 0^\circ$

The third case is the test case A6 from GAMM workshop [48]. The free stream Mach number is 0.85 and the Reynolds number is 2000.

The Mach number contour is shown in Fig. 4.15 which agrees very well with that of Forsyth and Jiang in [52]. Skin friction coefficient profile (the solid line in Fig. 4.16) agrees perfectly with the reference data (the solid squares) in [52]. And the separation point is 0.66, close to 0.70 in [52]. Similar to that of the second case ($M_\infty = 0.5$, $Re = 5000$), the pressure coefficient profile shown in Fig. 4.17 presents higher value near the trailing edge.

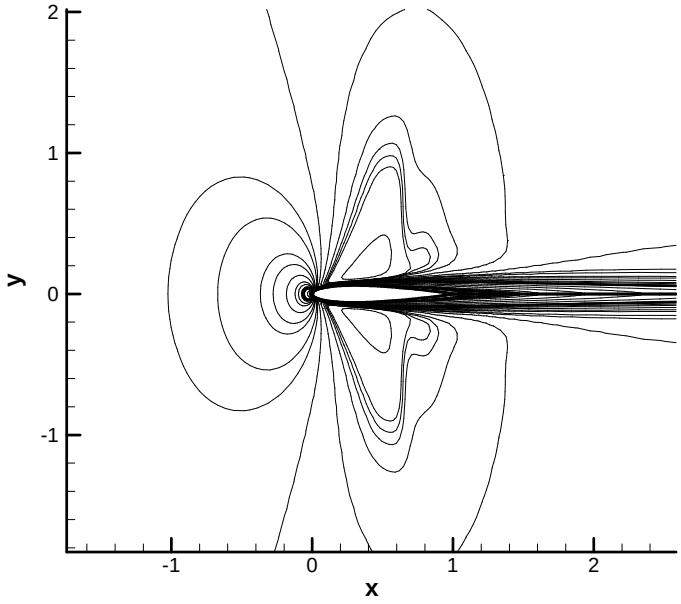

Figure 4.15: Mach number contours around NACA0012 ($\alpha = 0^\circ$, $M_\infty = 0.85$, $Re = 2000$).

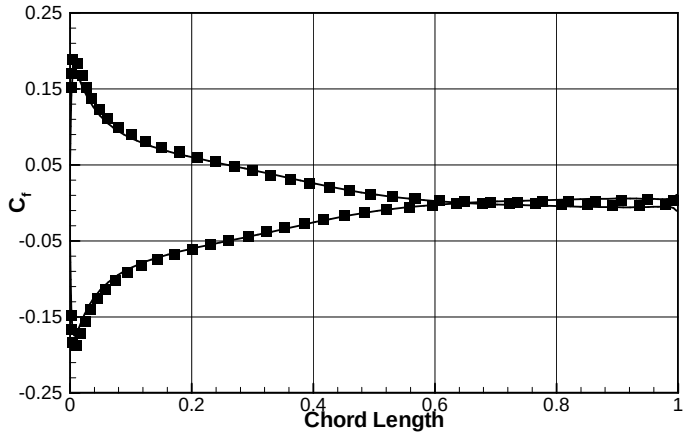

Figure 4.16: Skin friction coefficient for NACA0012 ($\alpha = 0^\circ$, $M_\infty = 0.85$, $Re = 2000$).

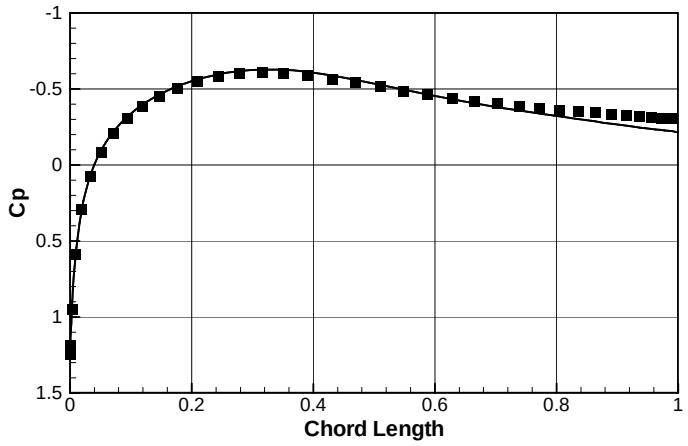

Figure 4.17: Pressure coefficient for NACA0012 ($\alpha = 0^\circ$, $M_\infty = 0.85$, $Re = 2000$).

Case 4, $M_\infty = 2, Re = 1000, \alpha = 10^\circ$

The last case, test case A4 from GAMM workshop [48], is a laminar supersonic flow problem with the angle of attack $\alpha = 10^\circ$, free-stream Mach number of 2.0, Reynolds number of 1000. Different from the previous three cases, there is no vortex in this case.

In this case, e_0 is set as 4.0 and the dimensionless time step $\Delta\breve{t} = 8.0 \times 10^{-5}$. In fact, in order to accelerate the convergence, a bigger time step $\Delta\breve{t} = 2.0 \times 10^{-4}$ was used first. After the residual is small enough, we then use the smaller time step and continue iteration.

The Mach number contour is shown in Fig. 4.18. And Figs. 4.19 and 4.20 show the skin friction and pressure coefficient distribution along the airfoil. It is interesting to see that our results agree excellently with the results computed with NS solver by Forsyth and Jiang in [52].

Observing all the above four simulations of flows over NACA0012, we can see that the results are very good and some of them are excellent. The pressure field and separation point have some deviations from reference data. It is believed that such deviations might be the result of the nonzero bulk viscosity in the recovered Navier-Stokes (4.16) from

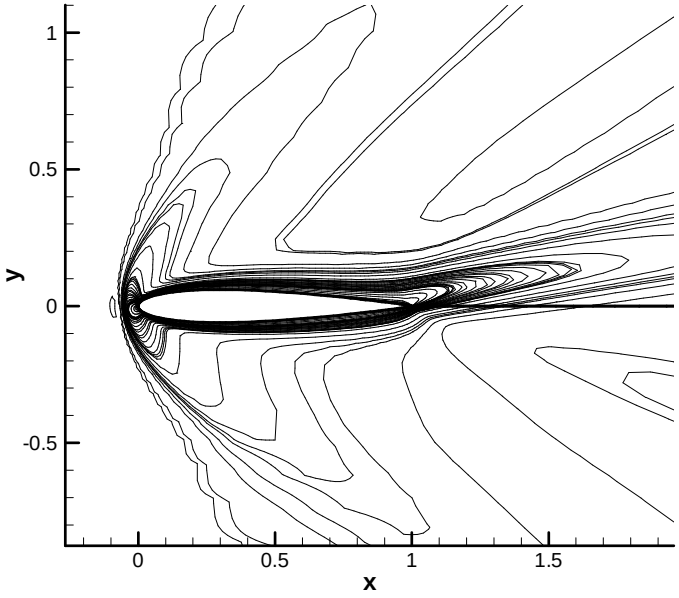

Figure 4.18: Mach number contours around NACA0012 ($\alpha = 10^\circ$, $M_\infty = 2$, $Re = 1000$).

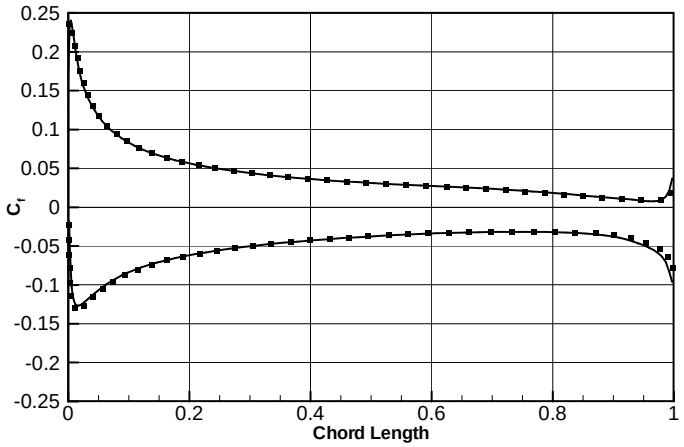

Figure 4.19: Skin friction coefficient for NACA0012 ($\alpha = 10^\circ$, $M_\infty = 2$, $Re = 1000$).

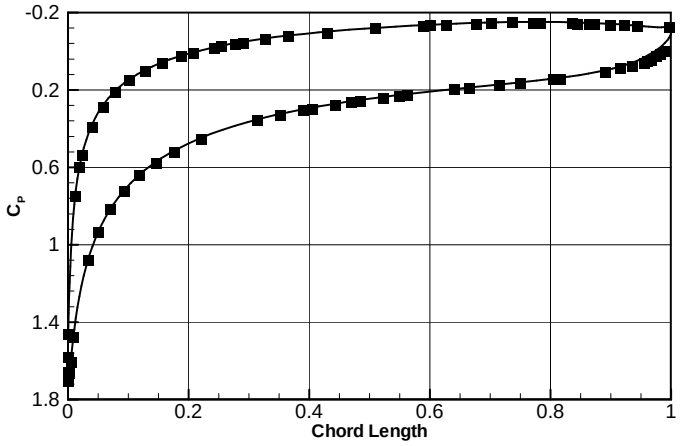

Figure 4.20: Pressure coefficient for NACA0012 ($\alpha = 10^\circ$, $M_\infty = 2$, $Re = 1000$).

DBVE. Consider the continuity equation of the following form

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \mathbf{v} \tag{4.22}$$

Here, $\frac{D\rho}{Dt}$ is the material derivative of density along a path line which is coincident with streamline in case of steady flows. From Equ. (4.22), we can see that along a path line, when the gas is expanding ($\frac{D\rho}{Dt} < 0$), the divergence of the velocity, $\nabla \cdot \mathbf{v} > 0$, and vise versa. For compressible flows, fluid severely expands or is compressed, especially near shock waves. Thus the term of bulk viscosity, $\mu_b \mathbf{I} \nabla \cdot \mathbf{v}$, plays a big role in the compressible flow field.

Since $\mu_b \mathbf{I} \nabla \cdot \mathbf{v}$ is a normal stress tensor, it is like a pseudo-pressure tensor. So it might directly affect the pressure field. That could be why the error of the pressure field is more obvious. The shear stress could feel the influence of the bulk viscous stress indirectly through pressure change. So, the skin friction is not very sensitive to the bulk viscous stress. As a result, it could agree well with the results by solving Navier-Stokes equation without the bulk viscous stress. In compression or expansion zones where $\nabla \cdot \mathbf{v}$ is big, the effects of the bulk viscosity stress might be big. However, if inertia is high enough, the effect of bulk viscosity might not be obvious to impact the flow field. When the inertia of fluid is not large enough to suppress the bulk viscosity stress, for instance, in a vortex

attached to the suction wall of a airfoil where gas is expanding and velocity is small, the bulk viscous stress might impact the flow field even if the derivative of $\nabla \cdot \mathbf{v}$ is not very big. The flow fields near vortices have a big error. In our simulations, C_p profiles deviate bigger at the place where vortices are produced. For Case 1, there is a big vortex on the upper surface. Thus the error of C_p is big. In Case 2 and 3, the vortices are small, so pressure coefficient profiles are much more accurate than that of Case 1. For Case 4, still $\alpha = 10°$, but there is no separation. So both C_p and C_f profiles are very accurate.

We can see that the bulk viscosity makes obvious different. In order to make more accurate simulations with LBM, eliminating the bulk viscous stress is necessary. MRT collision model [4, 5] might be a choice, since it provides more adjustable parameters to tune the transportation properties.

4.6 Concluding remarks

In this chapter, the new derivation method proposed in the last chapter is further extended to develop LB models for viscous compressible flows. A simple function, CF-VCF , is constructed to replace the Maxwellian function as the equilibrium density distribution function by modifying CF-ICF which is proposed for inviscid flows in Chapter 3. CF-VCF satisfies all statistical relations needed to recover NS equations. And CF-VCF is assigned onto the lattice velocity directions by the Lagrangian interpolating stencil to obtain the equilibrium distribution functions and the associated lattice velocity model. In the derivation, our new method presented its generality and simplicity. Except the simple function, which is used at the beginning of derivation, is different, all the other steps are same. And the whole procedure can be automatically implemented with Maple or Mathematica software.

Numerical experiments showed that compressible viscous flows can be well simulated by the present model. The numerical method is the same as that described in the previous chapter. For flows without vortices, such as Couette flow and flow around a NACA0012 with $M_\infty = 2$, $Re = 1000$ and 10 degree attack-angle, the results agree excellently with

previous studies. However when vortices are available, the simulations can not give accurate results. The non-zero bulk viscosity might be the reason. The bulk viscosity stress could make difference in the zone where gas is compressed or expanded suddenly, such as the vortices at the sucking face of a body. To overcome this problem, MRT collision model might be useful because more relaxation parameters [4, 5] could be used to adjust transportation coefficients.

Still, the computing is very time consuming since only full explicit scheme without any acceleration method was implemented. An case of NACA0012 of $Re = 5000$ could take 1 or 2 days. When simulating flows of higher Reynodls number, mesh size will be larger, making computing time longer. So, applying implicit scheme with some kind of acceleration methods is necessary.

Also, for viscous flows of higher Reynolds number, simulation of turbulent flows is a important topic. For LBM simulation of turbulent flows, turbulence models are still needed, like solving NS equations to simulate turbulent flows with RANS or LES.

Chapter 5

LBM-based Flux Solver

In this chapter, a new application of LBM in simulation of compressible flows, LBM-based flux solver, is briefly described. Different from solving DVBE in the previous chapters, Euler equations are solved in this approach, and the flux at the interface is evaluated by a compressible lattice Boltzmann model developed in the previous chapters. This application combines LBM and FVM for solving Euler/NS equations. The proposed approach is validated by its application to solve one-dimensional and three-dimensional test problems.

5.1 Finite volume method and flux evaluation for compressible Euler equations

In FVM, a domain is discretized into many small control volumes which are usually called cells. The variation of flow states in every cell is expressed as the sum of flux across all the interfaces of the cell.

$$\frac{\partial \bar{\mathbf{Q}} V}{\partial t} = -\sum_{k}^{K} \left(\mathbf{F}_k + \mathbf{D}_k\right) s_k$$

where $\bar{\mathbf{Q}}$ is the cell averaged flow state (density ρ , moment $\mathbf{v}$ and energy E), V is the volume, K is the number of interfaces of the control volume, $\mathbf{F}_k$ and $\mathbf{D}_k$ are the inviscid flux and viscous flux (only for viscous flows) across the k-th interface, and s_k is the area

the interface. When FVM is applied to solve compressible Euler/NS equations, one key issue is to evaluate the inviscid flux $\mathbf{F}_k$ at the interface. Among various FVM schemes, generalized Godunov schemes are widely used. In these schemes, the smooth profiles of variables in each cell are reconstructed according to the cell averaged quantities of the cell and its neighbors. Across the interfaces of the cells, the profiles may not be continuous. The great feature of generalized Godunov schemes is that the interaction between every two neighboring cells is considered with a flux solver $\hat{\mathbf{F}}$.

$$\mathbf{F} \approx \hat{\mathbf{F}}\left(\mathbf{Q}_l, \mathbf{Q}_r\right)$$

where $\mathbf{Q}_l$ and $\mathbf{Q}_r$ are flow states on the two sides of the interface. The task of a flux solver is to compute the inviscid flux of mass, momentum and energy through a discontinuous interface (Fig. 5.1). For the original Godunov scheme [53], the piecewise constant profile inside a cell was assumed. So $\mathbf{Q}_l$ and $\mathbf{Q}_r$ are the same as cell average values on each side of the interface

$$\mathbf{Q}_l = \bar{\mathbf{Q}}_l$$
$$\mathbf{Q}_r = \bar{\mathbf{Q}}_r$$

where $\bar{\mathbf{Q}}_l$ and $\bar{\mathbf{Q}}_r$ are cell average values. The second order generalized Godunov schemes, such as MUSCL schemes, take linear profiles. Thus, $\bar{\mathbf{Q}}_l$ is expressed as

$$\mathbf{Q}_l = \bar{\mathbf{Q}}_l + (\mathbf{x}_l - \mathbf{x}_c) \cdot \nabla \mathbf{Q}$$

where $\mathbf{x}_l$ is the location of the interface center, $\mathbf{x}_c$ is the center of the control volume. For higher order generalized Godunov schemes like ENO/WENO, higher order polynomial profiles are used.

Originally, the exact Riemann flux solver was used by Godunov [53]. That is the exact solution of a Riemann problem. A Riemann problem is a Cauchy problem of Euler

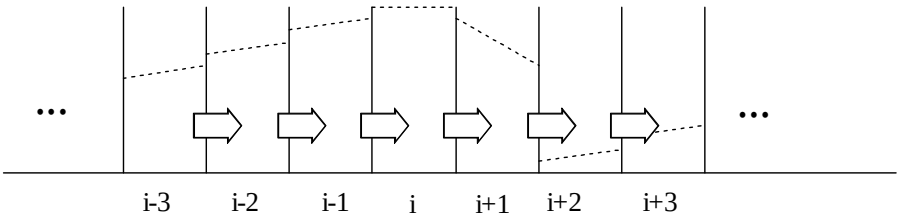

Figure 5.1: The schematic view of one dimensional Godunov scheme. The domain is divided into some finite volumes, $(\ldots i-3, \ldots, i+3, \ldots)$. The profiles of variables are assumed linear in every volume. The white arrows are interfaces between neighboring cells

equations with initial data that is discontinuous along a plane, i.e

$$\mathbf{q}_0(x) = \begin{cases} \mathbf{q}_L & x < 0 \\ \mathbf{q}_R & x > 0 \end{cases}$$

in one dimensional space. Later, several approximate one dimensional Riemann flux solvers (such as Roe [54], HLL [55] and Osher [56]) were developed.

At the same time, flux vector splitting (FVS) methods [57, 58] are also widely used as flux solvers to compute flux across the cell interface. The idea of FVS is to divide the flux vector $[\rho u, \rho u u + p, (\rho E + p) u]$ into several sub-vectors along different directions and determine the flux crossing an interface according to the directions of sub-vectors on the two sides of the interface. Fig. 5.2 illustrates the idea of FVS. $\mathbf{F}(\mathbf{Q}_l)$ at the left side of the interface is divided into two parts, $F^-(\mathbf{Q}_l)$ and $F^+(\mathbf{Q}_l)$. And $F^+(\mathbf{Q}_l)$ contributes to the interface flux because it crosses the interface. Also, $\mathbf{F}(\mathbf{Q}_r)$ is divided and its contribution is determined in the same way. Thus the interface flux $\mathbf{F} = \mathbf{F}^-(\mathbf{Q}_r) + \mathbf{F}^+(\mathbf{Q}_l)$.

In addition, based on the gas kinetic theory, Xu [23] proposed a BGK gas kinetic scheme which was derived from the BGK Boltzmann equation. The idea of the kinetic scheme is to compute flux from the distribution functions on both sides of an interface. For a distribution function, the effective part which crosses the interface contributes to the interface flux. The two effective parts of the distribution functions on the two sides of interface are integrated in time and phase space to evaluate the interface flux. Moreover,

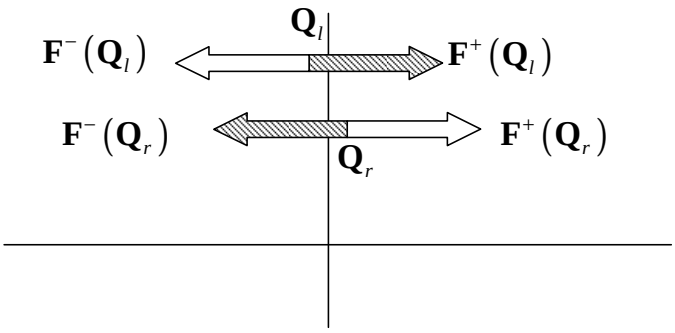

Figure 5.2: Schematic view of FVS. The shaded vectors cross the interface and contribute to the interface flux.

instead of using equilibrium distribution function, Xu [23] took account the effect of non-equilibrium due to the gradient of the gas inside every volumes to achieve higher accuracy. Different from the approximate Riemann flux solvers which can only compute inviscid flux, Xu's BGK kinetic scheme can compute viscous flux at the same time. Also, he claimed that this scheme is more stable than approximate Riemann flux solvers. The distribution function is used only during derivation of the flux solver, while the flux solver can be expressed in form of the macroscopic variables. However, Xu's kinetic scheme is a little complicated because some integrals of the Maxwellian function are involved during the derivation process, and taking non-equilibrium effect into consideration introduces more complexity.

Nevertheless, using distribution function on an interface to compute flux provides a good effective way. This idea can alos be applied in LBM. Since summations replace integrals in LBM, the kinetic scheme of LBM version should be simpler and its application is much easier.

5.2 LBM-based flux solver

In previous chapters, DVBE was solved with FVM. Due to the linear advection term in DVBE, the flux of the distribution functions through the interfaces could be computed easily with the exact Riemann solver. Since macroscopic variables (density, momentum

and energy) and their fluxes can be derived from the distribution functions, the computation of the flux from distribution functions could be regarded as a special flux solver which might be applied in solving Euler/NS equations.

When DVBE is solved, the flux through an interface can be determined as follows (Fig. 5.3).First, extrapolate the distribution functions on each side of an interface, $\mathbf{f}^L$ and $\mathbf{f}^R$, by means of MUSCL scheme. If the discrete velocity ($\mathbf{e}_k$) of a component of the distribution function, f_k , crosses the interface, this component contributes to the interface flux. Then, the flux vector can be determined. At this point, we can see that the flux of macroscopic variables can be determined by $\mathbf{f}^L$ and $\mathbf{f}^R$. Because the $[\rho u, \rho uu + p, (\rho E + p)\, u]$ contributed by the component f_k

$$\mathbf{F}_k = \left[\begin{array}{l} \rho \mathbf{v} \\ \rho \mathbf{v}\mathbf{v} + p\mathbf{I} \\ (\rho E + p)\, \mathbf{v} \end{array} \right]_k \cdot \mathbf{n}A = \left[\begin{array}{l} f_k \mathbf{e}_k \\ f_k \mathbf{e}_k \mathbf{e}_k \\ \left(\frac{1}{2} f_k \mathbf{e}_k \cdot \mathbf{e}_k + f_k \lambda\right) \mathbf{e}_k \end{array} \right] \cdot \mathbf{n}A$$

where A is the area of the interface and $\mathbf{n}$ is its unit normal vector, the total flux through the interface is obtained by summing up all the contributions of those components across the interface.

In one dimensional space, assuming that two-velocity model is applied for simplicity, the component f_1^{L} of $\mathbf{f}^L$ and f_2^R of $\mathbf{f}^R$ are across the interface (Fig. 5.3). Thus the flux of macroscopic variables is the summation of the fluxes of f_1^{L} and f_2^R.

$$\begin{aligned}
&\rho \mathbf{v} = f_1^L \mathbf{e}_1 + f_2^R \mathbf{e}_2 = \sum_k f_k^E \mathbf{e}_k \\
&\rho \mathbf{v}\mathbf{v} + p\mathbf{I} = f_1^L \mathbf{e}_1 \mathbf{e}_1 + f_2^R \mathbf{e}_2 \mathbf{e}_2 = \sum_k f_k^E \mathbf{e}_k \mathbf{e}_k \\
&(\rho E + p)\, \mathbf{v} = \mathbf{e}_1 f_1^L \left(\tfrac{1}{2} \mathbf{e}_1 \cdot \mathbf{e}_1 + \lambda_1\right) + \mathbf{e}_2 f_2^R \left(\tfrac{1}{2} \mathbf{e}_2 \cdot \mathbf{e}_2 + \lambda_2\right) = \sum_k \mathbf{e}_k f_k^E \left(\tfrac{1}{2} \mathbf{e}_k \cdot \mathbf{e}_k + \lambda_k\right)
\end{aligned}$$

This means that we could regard f_1^{L} and f_2^R as a set of effective distribution functions $\mathbf{f}^E$ on the interface. Then the flow states on the interface can be determined directly from $\mathbf{f}^E$. As a consequence, the flux can be computed.

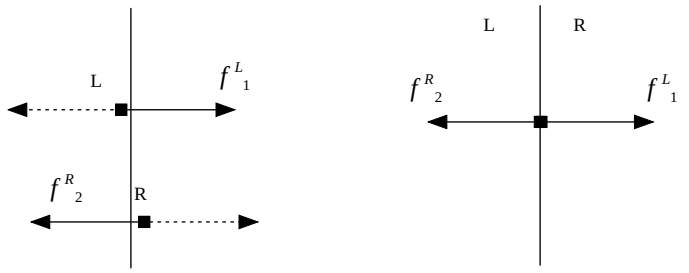

Figure 5.3: The effective distribution functions on an interface.

As shown above, a new flux solver might be developed based on our LB models for compressible flows. That is to compute distribution functions, $\mathbf{f}^L$ and $\mathbf{f}^R$, on each side of an interface and then apply the above method to construct effective distribution function on the interface. From the distribution function, flow state and flux on the interface can be computed. The easiest way is to assume equilibrium states on the interface. This means that $\mathbf{f}^L$ and $\mathbf{f}^R$ are computed by equilibrium functions according to macroscopic variables $\mathbf{Q}^L$ and $\mathbf{Q}^R$ obtained with MUSCL extrapolation or other reconstruction method. With this algorithm, CFL number can be determined by the convection term of Euler equations, making time step much larger than that of DVBE simulations by several orders. So this algorithm is much faster than direct solving DVBE with the full explicit method used in previous chapters. However, during implementation, it was found that this procedure is not very efficient since it needs the flow states on the two sides of the interface to be scaled to the lattice unit when $\mathbf{f}^L$ and $\mathbf{f}^R$ are computed, and be transferred back after $\mathbf{f}^E$ is constructed.

In fact, this flux solver based on LBM can also be interpreted and implemented as a special FVS scheme. The procedure can be considered as the following LBM flux vector splitting (FVS) scheme, which can be more conveniently implemented than computing flux by constructing the effective distribution function $\mathbf{f}^E$ stated above. The procedure can be described as following (Fig. 5.4):

1. Reconstruct profiles of macro variables in cell i and get $\mathbf{Q}^R_{i-1/2}$ and $\mathbf{Q}^L_{i+1/2}$ on the two interfaces of the cell.

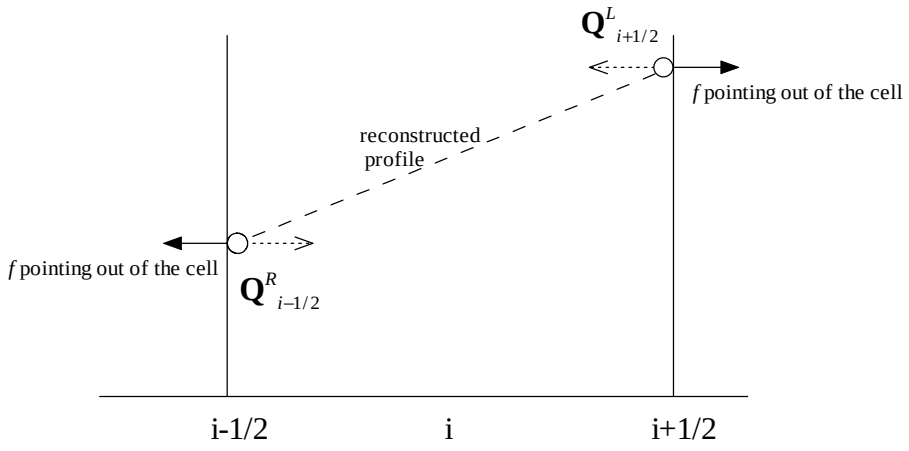

Figure 5.4: Flux vector splitting implementation of LBM based flux solver.

2. For each of $\mathbf{Q}^R_{i-1/2}$ and $\mathbf{Q}^L_{i+1/2}$, determine discrete velocity stencil according to the total energy max $\max(\|\mathbf{e}_k\|) = \sqrt{e^*}$, which means that there is no need to scale the macroscopic variables and transfer back the distribution function. Compute discrete equilibrium function f^{eq} and obtain flux across the interface from those components of crossing the interface. Thus the flux of macroscopic variables contributed by f^{eq} can be obtained.

3. After all the cells are accessed and their contributions are computed, the fluxes between every couple of neighbor cells are determined.

5.3 Numerical validation for one dimensional FV-LBM scheme

In this section, the one dimensional Sod shock tube case was presented to test the FV-LBM. In the simulation, the 3rd order MUSCL scheme and Van Albada slope limiter [39] were used for reconstruction. The 4-stage Runge-Kutta method was applied for time integration. The domain is equally divided into 100 small cells.

The LB model used here is D1Q4L2, of which 4 velocities are d_1, $-d_1$, d_2, $-d_2$. This model has no static particle which could make trouble when identifying whether it will be across an interface. Here, d_1 and d_2 are left as variables which can be determined

according to local total energy e^* . In our simulation, $d_1 = d_2/4$. The expression of D1Q4L2 is listed in Equ. (5.1a).

$$\begin{cases} \rho_1 = \dfrac{\rho\left(-d_1\, {d_2}^2 - {d_2}^2\, u + d_1 u^2 + d_1 c^2 + u^3 + 3\, u c^2\right)}{2d_1\left({d_1}^2 - {d_2}^2\right)} \\ \rho_2 = \dfrac{\rho\left(-d_1 d_2^2 + {d_2}^2\, u + d_1 u^2 + \ d_1 c^2 - u^3 - 3\, u c^2\right)}{2d_1\left({d_1}^2 - {d_2}^2\right)} \\ \rho_3 = \dfrac{\rho\left({d_1}^2\, d_2 + {d_1}^2\, u - d_2 u^2 - \ d_2 c^2 - u^3 - 3\, u c^2\right)}{2d_1\left({d_1}^2 - {d_2}^2\right)} \\ \rho_4 = \dfrac{\rho\left({d_1}^2\, d_2 - {d_1}^2\, u - d_2 u^2 - \ d_2 c^2 + u^3 + 3\, u c^2\right)}{2d_1\left({d_1}^2 - {d_2}^2\right)} \end{cases} \tag{5.1a}$$

$$\begin{cases} f_{i0}^{eq} = \rho_i\,(1 - e_p) \\ f_{i1}^{eq} = \rho_i e_p \end{cases} \tag{5.1b}$$

where $e_p = \dfrac{3}{2}(\gamma - 1)\, e$. The result is presented in Fig. 5.5. The lines are exact solution, while the solid symbols are computed with FV-LBM. It can be seen clearly that our new flux solver provides very good results.

As stated above, the time step is determined by the CFL condition, which is not limited by the stiff collision term. And only three equations (continuity, momentum and energy equations) rather than eight equations of distribution functions. So it is much faster than directly solving DVBE. Although accuracy analysis was not made, the fact that such solution profiles are obtained with grid size of 100 may suggest the second order of accuracy for the solution. Later numerical test on the 3D case also confirms this point.

5.4 Multi-dimensional application of FV-LBM

Although multi-dimensional FV-LBM can be developed based on the multidimensional LB models developed in the previous chapters, this way is not efficient since one dimensional FV-LBM can be well applied to simulate multi-dimensional problems, just like applying one dimensional flux solvers in multi-dimensional problems.

Usually, in a multi-dimensional simulation, the one dimensional flux solver is applied along the normal direction of an interface. So, mass flux, normal momentum flux and

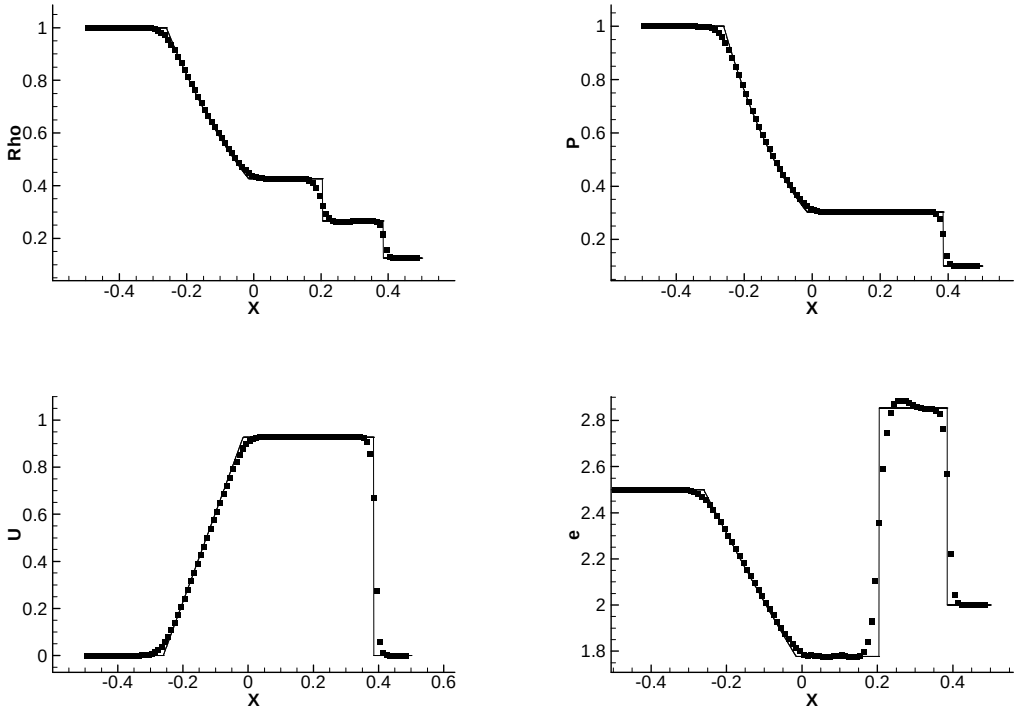

Figure 5.5: Sod shock tube simulation by solving Euler equations with LBM-based Flux Vector Splitting scheme. The grid size is 100.

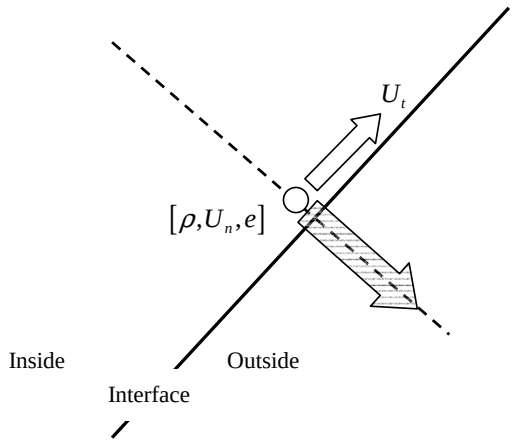

Figure 5.6: Schematic view of applying 1D LBM FVS in multi-dimensional problems. The flux solver is operated along the normal direction (dash-dot line) of the interface. $\mathbf{U}_n$ is used as velocity to compute 1D normal flux, while the momentum and kinetic energy of tangent velocity $\mathbf{U}_t$ are passively transported by the mass flux.

normal energy flux can be obtained. The tangent momentum flux and tangent kinetic energy can be regarded as passive quantities transported by mass flux (shown in Fig. 5.6).

So, the procedure of applying FV-LBM for 3D problems can be described as follows:

1. Reconstruct profiles of macro variables in cell and extrapolate to obtain $\mathbf{Q}^{in} = [\rho, u, v, w, e]^T$ on the inner sides of all the interfaces

2. For a $\mathbf{Q}^{in}$ at an interface $\mathbf{A} = A\,[n_x, n_y, n_z]^T$ ($[n_x, n_y, n_z]^T$ is the unit normal vector of the surface, A is the area), compute total energy e^*, the normal velocity $\mathbf{U}_n = U_n\,[n_x, n_y, n_z]^T$ and the tangent velocity $\mathbf{U}_t = [U_{tx}, U_{ty}, U_{tz}]^T$. e^* will be used to determine the lattice, U_n is used as the velocity in the one dimensional flux solver, while U_t will be used to compute passively transported moment and kinetic energy.

3. Determine discrete velocities ($d_2 = \sqrt{e^*}$ and $d_1 = d_2/4$) according to e^*. Based on these velocities, discrete equilibrium distribution functions $\mathbf{f}^{eq}$ can be computed according to $[\rho, U_n, e]$. From these distribution functions, select those functions streaming out of the cell, $\mathbf{f}^{eq}_{out}$

4. Compute the mass flux, momentum flux and entropy flux normal to the interface

from the cell across the interface A according to the 1D LBM FVS stated in previous section

$$\rho U_n A = A \sum_i f^{eq}_{out,i} e_i \tag{5.2}$$

$$(\rho U_n U_n + p)\,\mathbf{A} = A \sum_i f^{eq}_{out,i} e_i e_i \left[n_x, n_y, n_z\right]^T \tag{5.3}$$

$$\left[\rho\left(e + \frac{1}{2}U_n^2\right) + p\right]\mathbf{U}_n \cdot \mathbf{A} = A \sum_i e_i f^{eq}_{out,i}\left(\frac{1}{2}e_i e_i + \lambda_i\right) \tag{5.4}$$

5. Compute the passively transported tangent momentum flux as

$$\rho U_n A \left[U_{tx}, U_{ty}, U_{tz}\right] \tag{5.5}$$

where $\rho U_n A$ is obtained from Equ. (5.2). Thus the total moment flux can be obtained as

$$A \sum_i f^{eq}_{out,i} e_i e_i \left[n_x, n_y, n_z\right]^T + \rho U_n A \left[U_{tx}, U_{ty}, U_{tz}\right]^T \tag{5.6}$$

6. Passively transported tangent kinetic energy is

$$\frac{1}{2}\rho U_n A U_t^2 \tag{5.7}$$

So the total flux of total enthopy is

$$A \sum_i e_i f^{eq}_{out,i}\left(\frac{1}{2}e_i e_i + \lambda_i\right) + \frac{1}{2}\rho U_n A U_t^2 \tag{5.8}$$

Thus the flux vector from this cell across the interface $\mathbf{F}_A^+$ is obtained. With the same method, the flux vector from the neighboring cell to this cell, $\mathbf{F}_A^-$, can also be computed. Then the total flux across the interface is $\mathbf{F}_A = \mathbf{F}_A^+ + \mathbf{F}_A^-$.

This flux solver can be implemented in the form of FVS and easily incorporated in traditional FVM Euler/NS solver. In a 3D multi-block structured grid solver, ADIMB developed by the CFD lab in ADI Corp., this new FV-LBM was implemented without

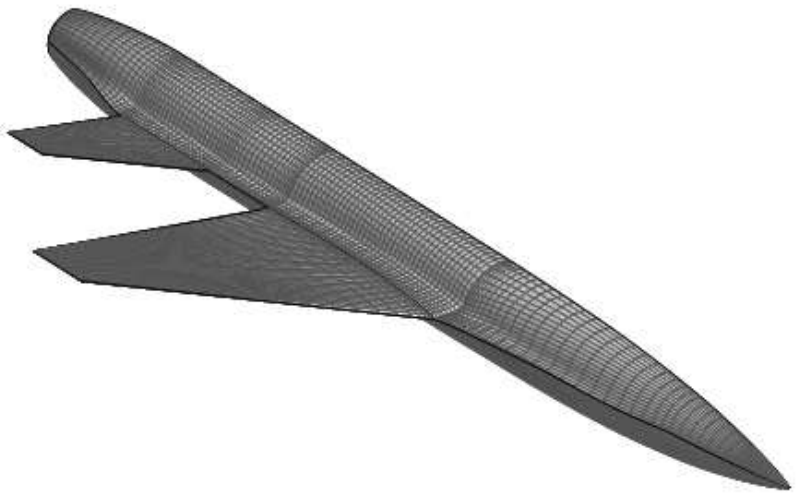

Figure 5.7: Srface grid of AFA model.

modifying other parts. LU-SGS implicit time advancing scheme is used with a large CFL number (up to 20).

With this solver, AFA model [59] was computed with Jameson's central scheme, Van Leer FVS scheme and present scheme. AFA model is a half-model of a simplified F16 fighter, used for aeroelastic analysis. The flow condition is defined as inviscid flow, Mach number 0.8 and 2° angle of attack. The grid contains four blocks. And about $310,000$ mesh points are used.

Fig. 5.7 shows the grid of the model. The grid consists of four blocks. And H-type grid is used to carve the wings. The pressure coefficient contours computed with the three schemes are presented in Fig. 5.8. We can see that very good agreement is achieved, especially between Van Leer FVS and LBM FVS. Tab. 5.1 shows coefficients of lift, drag and moment computed with the three schemes, which also shows the agreement between these three results. Obviously, the present results are in excellent agreement with those of Van Leer FVS scheme. Fig. 5.9 presents the residual histories of the three schemes, from which we can see that the convergence of present scheme is almost the same as Van Leer FVS. The time used by LBM FVS is about 25% more than that of Van Leer scheme. Considering that the current implementation of LBM FVS is not optimized at all, this efficiency could be satisfactory. In the current implementation, the equilibrium functions

	C_l	C_d	C_M
Jameson's central scheme	0.2303	0.0103	-0.0225
Van Leer FVS	0.2384	0.0124	-0.0336
LB-FVS	0.2375	0.0123	-0.0345

Table 5.1: Coefficients of lift, drag and moment computed with Jameson's central scheme, Van Leer FVS scheme and our LBM-FVS scheme

are actually computed. In fact, it is not necessary to compute them. And the flux vectors can be directly expressed as functions of those macroscopic flow variables. In this way, the efficiency can be improved.

5.5 Concluding remarks

In this chapter, an idea of developing LBM-based flux solver was presented, which can be directly implemented in the conventional finite volume Euler solver. In this flux solver, with equilibrium distribution functions on two sides of a cell interface, the distribution functions streaming across the interface contribute to the flux through the interface. With this idea, a flux vector splitting (FVS) scheme based on LBM was developed. The proposed approach was first validated by its application to solve the one dimensional Sod shock tube problem, and then applied to simulate the three-dimensional flow around an aircraft by using a multi-block 3D CFD solver. The obtained numerical results are compared excellently well with those of Van Leer FVS scheme.

So far, it is not clear what advantages this LBM FVS has. At least, LBM based FVS could provide an new view of flux solver. From this view, people might obtain useful hints. Due to time constraint, only preliminary applications of FV-LBM were demonstrated in this chapter. More analysis, tests and validations will be needed in the future work to find out the advantages/disadvantages, accuracy and efficiency of the proposed FV-LBM.

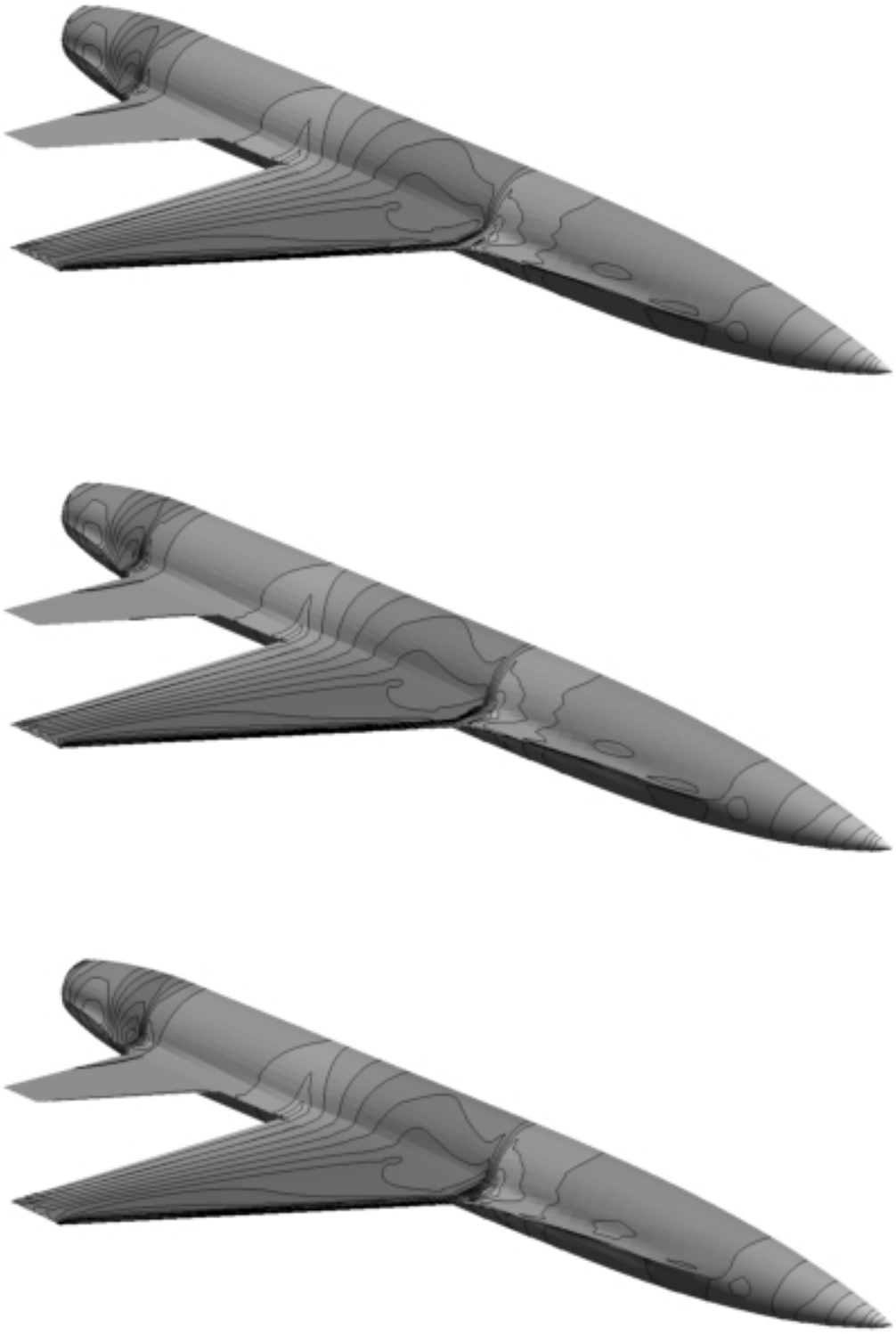

Figure 5.8: Surface C_p contours computed with FV-LBM (top), Van Leer FVS (middle) and Jameson's central scheme (bottom). Contour levels are from -0.6 to 0.5 with step as 0.05.

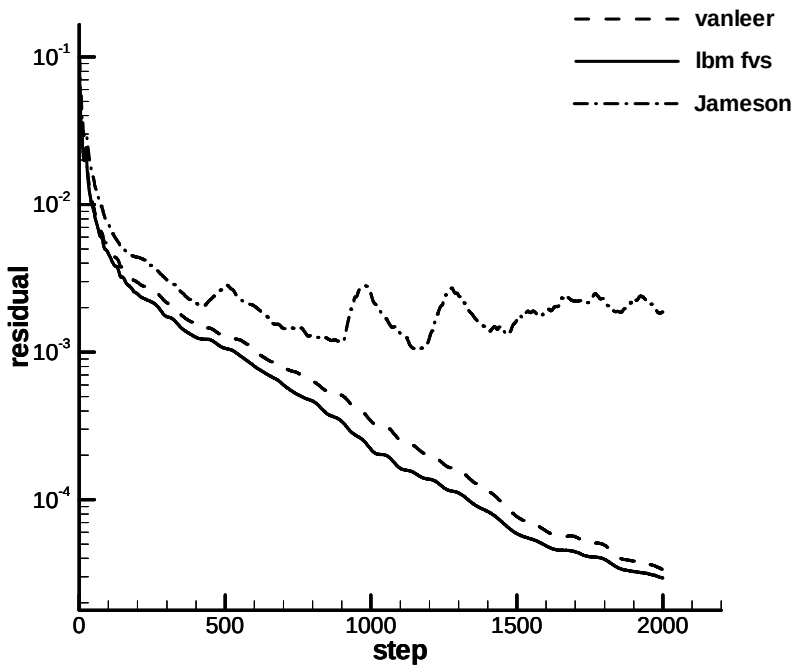

Figure 5.9: Comparison of convergence history.

Chapter 6

Conclusion and Outlook

6.1 Conclusion

In this thesis, a novel and convenient derivation method for constructing lattice Boltzmann models have been described. In this method, the Maxwellian distribution function is replaced by a circular function which is very simple and satisfies all needed statistical relations to recover the compressible Euler/NS equations. By the Lagrangian interpolation, the circular function is then discretized onto the lattice velocity directions with all the needed statistical relations exactly satisfied. In this frame, the equilibrium distribution functions and the associated lattice velocity model can be derived naturally without assuming specific forms.

This method was first applied to develop a D2Q13 model for isothermal incompressible flows. A circular function in velocity space was proposed and discretized with a two-dimensional Lagrangian polynomial of the third degree to construct the D2Q13 model. The new D2Q13 model was tested by simulating the lid driven cavity flow with the standard streaming-collision procedure and the finite difference method. Excellent results were obtained, which proved the validity and feasibility of the new deriving method. The new deriving method provides some interesting suggestions about LBM. First, in this deriving method, there is no need to assume a formula beforehand. In fact, it can be derived step by step naturally. Also the configuration of the lattice can be obtained with

some knowledge of polynomial and linear theory. The idea of this deriving method is very natural and clear. Second, we do not need the small Mach number assumption and isothermal assumption as mathematical requirements during derivation. Although the D2Q13 is derived for incompressible and isothermal flows, these two conditions are imposed only during simulations. So this method has fewer limits. Third, it is interesting to note that using a symmetric stencil might not be necessary for recovering Navier-Stokes equations. Since numerical methods such as FVM and FEM can handle irregular gird and lattice and the wall boundary conditions can be implemented with non-equilibrium extrapolation method, LB models based on an asymmetric lattice might be feasible.

Lattice Boltzmann models for inviscid compressible flows were developed with this deriving method. A new circular function was constructed in a velocity-energy space. This circular function satisfies all statistical relations needed to recover the compressible Euler equations. The idea of discretization with Lagrangian interpolating stencil was extended to the velocity-energy space. D1Q5L2 model and D2Q13L2 model were developed. And with MUSCL FVM scheme, the models are used to simulate several cases with shock waves. Again, the excellent results show that the deriving method is right and the derived models are feasible for simulation of high Mach number inviscid flows.

In order to develop lattice Boltzmann models for viscous compressible flows, the second version of circular function was modified to cancel the extra heat source in the energy equation, and a D2Q17L2 model was derived naturally. It should be indicated that the recovered Navier-Stokes equation contains a nonzero bulk viscous stress and the Prandtl number is fixed as due to the limit of the BGK collision model. Nevertheless satisfactory numerical results of laminar flow over NACA0012 airfoil were obtained and some of them agree perfectly with simulation results by a Navier-Stoke solver.

LB models for compressible flows can also be applied in solving Euler equations. An idea of developing LBM-based flux solver was presented, which can be directly implemented in the conventional finite volume Euler solver. In this flux solver, with equilibrium distribution functions on two sides of a cell interface, the distribution functions streaming across the interface contribute to the flux through the interface. With this idea, a flux

vector splitting (FVS) scheme based on LBM was developed. The proposed approach was first validated by its application to solve the one dimensional Sod shock tube problem, and then applied to simulate the three-dimensional flow around an aircraft by using a multi-block 3D CFD solver. The obtained numerical results are compared excellently well with those of the Van Leer FVS scheme.

6.2 Recommendation for future work

The work presented in this thesis allows several possible extensions. Although several LB models for compressible flows were developed in this work, more work has to be done to make them perfect. In our models, the number of discrete velocities might be too many. It is possible to decrease them by optimizing the Lagrangian interpolating stencil. For the viscous model, the nonzero bulk viscosity and the fixed Prandtl number impede its application. Adjustable bulk viscosity and Prandtl number might be achieved by applying the multi-relaxation-time collision [3, 5].

The temporal scheme used in this work is the Euler forward scheme. However the time step is not determined according to the CFL condition but severely limited by the collision term which is a stiff source term. Thus the time step used in this work is much smaller than that determined from the CFL condition, especially for simulations of inviscid flows in which grids are much coarser than those used in simulations of viscous flows. So applying implicit scheme to the collision term should be a good choice. In fact, Implicit-Explicit (IMEX) algorithm has been studied to solve hyperbolic system with stiff sources [60–64]. For BKG equation, because the collision term dosen't change mean flow variables (density, velocity and internal energy), applying IMEX can be very easy and cheap [65]. But simulation of viscous flow with this BGK simplified IMEX haven't been released.

Furthermore, since we abandoned the standard streaming-collision procedure and finite volume method was used in this work, many related numerical methods in the traditional CFD can be applied. For instance, local time step, multigrid method, implicit

residual smooth, linearized implicit algorithm ..., can be applied to accelerate convergence. Automatic mesh refine (AMR), anisotropic mesh refine, higher order FVM schemes (PPM, ENO/WENO), or high order finite element methods (spectral element method, discontinuous Galerkin method) ..., can be used to obtain more accurate results.

Another interesting study is the flux solver based on LBM. In Chapter 5, only preliminary applications of FV-LBM were demonstrated. More analysis, tests and validations have to be done in the future to find out the advantages/disadvantages, accuracy and efficiency of the proposed FV-LBM. Although the current LB based flux solver is much simpler than the kinetic scheme of Xu Kun, it is not as powerful and stable as the kinetic scheme. It is possible to integrate in time to incorporate viscous effects like the kinetic scheme. Also, LB based flux solver might be tried to be applied in more fields.

Bibliography

[1] U. Frish, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the Navier-Stokes equation. *Physical Review Letters*, 56:1505, 1986.

[2] L. Bhatnagar, E. P. Gross, and K. Krook. A model for collision processes in gases. *Physical Review*, 94:511, 1954.

[3] M. Soe, G. Vahala, P. Pavlo, L. Vahala, and H. Chen. Thermal lattice Boltzmann simulations of variable Prandtl number turbulent flows. *Physical Review E*, 57:4227, 1998.

[4] P. Lallemand and L.-S. Luo. Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, Galilean invariance, and stability. *Physical Review E*, 61:6546, 2000.

[5] M. Krafczyk, P. Lallemand, D. d'Humires, I. Ginzburg, and L.-S. Luo. Multiple-relaxation-time lattice Boltzmann models in three-dimensions. *Philosophical Transactions of Royal Society of London A*, 360:437, 2002.

[6] S. Hou, Q. Zou, S. Chen, G. Dollen, and A. C. Cogley. Simulation of cavity flow by the lattice Boltzmann method. *Journal of Computational Physics*, 118:329, 1995.

[7] F. J. Alexander, S. Chen, and J. D. Sterling. Lattice Boltzmann thermohydrodynamics. *Physical Review E*, 47:R2249, 1993.

[8] XY. He and Li-Shi Luo. A priori derivation of the lattice Boltzmann equation. *Physical Review E*, 55:R6333, 1997.

[9] X. He and L.-S. Luo. Theory of the lattice Boltzmann method: from Boltzmann equation to the lattice Boltzmann equation. *Physical Review E*, 56:6811, 1997.

[10] X. Shan and X. He. Discretization of the velocity space in the solution of the Boltzmann equation. *Physical Review Letters*, 80:65, 1998.

[11] X. Shan, H. Chen, and X. Yuan. Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation. *Journal of Fluid Mechanics*, 55:413, 2006.

[12] Y. Chen, H. Ohashi, and M. Akiyama. Thermal lattice Bhatnagar-Gross-Krook model without nonlinear deviations in macrodynamic equations. *Physical Revew E*, 50:2776, 1994.

[13] Y. Chen, H. Ohashi, and M. Akiyama. Heat-transfer in lattice BGK modeled fluid. *Journal of Statistical Physics*, 81:71, 1995.

[14] X. Shan. Simulation of Rayleigh-Benard convection using a lattice Boltzmann method. *Physical Revew E*, 55:2780, 1997.

[15] X. He, S. Chen, , and G. D. Doolen. A novel thermal model for the lattice Boltzmann method in incompressible limit. *Journal of Computational Physics*, 146:282, 1998.

[16] Y. Peng Y, C. Shu, and Y. T. Chew. Simplified thermal lattice Boltzmann model for incompressible thermal flows. *Physical Revew E*, 68:026701, 2003.

[17] O. Filippova and D. Hanel. Lattice-BGK model for low Mach number combustion. *International Journal of Modern Physics C*, 9:1439, 1999.

[18] O. Filippova and D. Hanel. A novel lattice BGK approach for low Mach number combustion. *Journal of Computational Physics*, 158:139, 2000.

[19] P. Lallemand and L.-S. Luo. Hybrid finite-difference thermal lattice Boltzmann equation. *International Journal of Modern Physics B*, 17:41, 2003.

[20] M. Watari and M. Tsutahara. Two-dimensional thermal model of the finite-difference lattice Boltzmann method with high spatial isotropy. *Physical Review E*, 67:036306, 2003.

[21] M. Watari. Possibility of constructing a multispeed Bhatnagar-Gross-Krook thermal model of the lattice Boltzmann method. *Physical Review E*, 70:016703, 2004.

[22] M. Watari and M. Tsutahara. Supersonic flow simulations by a three-dimensional multispeed thermal model of the finite difference lattice Boltzmann method. *Physica A*, 364:129, 2003.

[23] K. Xu. A gas-kinetic BGK scheme for the Navier-Stokes equations, and its connection with artificial dissipation and godunov method. *Journal of Computational Physics*, 171:289, 2001.

[24] G. Yan, Y. Chen, and S. Hu. Simple lattice Boltzmann model for simulating flows with shock wave. *Physical review E*, 59:454, 1999.

[25] W. Shi, W. Shyy, and R. Mei. Finite-difference-based lattice Boltzmann method for inviscid compressible flows. *Numerical heat transfer, Part B*, 40:1, 2001.

[26] A. Harten. High resolution schemes using flux limiters for hyperbolic conservation laws. *Journal of Computational Physics*, 49:357, 1983.

[27] T. Kataoka and M. Tsutahara. Lattice Boltzmann method for the compressible Euler equations. *Physical review E*, 69:056702–1, 2004.

[28] T. Kataoka and M. Tsutahara. Lattice Boltzmann method for the compressible Navier-Stokes equations with flexible specific-heat ratio. *Physical review E*, 69:035701–1, 2004.

[29] C. Sun. Lattice-Boltzmann method for high speed flows. *Physical review E*, 58:7283, 1998.

[30] C. Sun. Adaptive lattice Boltzmann model for compressible flows: viscous and conductive properties. *Physical review E*, 61:2645, 2000.

[31] C. Sun. Simulations of compressible flows with strong shcoks by an adaptive lattice Boltzmann model. *Journal of Computational Physics*, 161:70, 2000.

[32] C. Sun and A. T. Hsu. Three-dimensional lattice Boltzmann model for compressible flows. *Physical review E*, 68:016303, 2003.

[33] C. Sun and A. T. Hsu. Multi-level lattice Boltzmann model on square lattice for compressible flows. *Computers and Fluids*, 33:1363, 2004.

[34] G.-H. Cottet and P. Koumoutsakos. *Vortex methods: theory and practice*. Cambridge university press, 2000.

[35] U. Ghia, K. N. Ghia, and C. T. Shin. High-resolutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48:387, 1982.

[36] Z. Guo and T. S. Zhao. Explicit finite-difference lattice Boltzmann method for curvilinear coordinates. *Physical Review E*, 67:066709, 2003.

[37] Z. Guo and C. Zheng. An extrapolation method for boundary conditions in lattice Boltzmann method. *Physics of Fluids*, 14:2002, 2002.

[38] B. Van Leer. Towards the ultimate conservative difference scheme, V. A second order sequel to Godunov's method. *Journal of Computational Physics*, 32:101, 1979.

[39] G. D. Van. Albada, B. Van Leer, and W. W. Roberts. A comparative study of computational methods in cosmic gas dynamics. *Astronomy and Astrophysics*, 108:76, 1982.

[40] P. Colella. Multidimensional upwind methods for hyperbolic conserva- tion laws. *Journal of Computational Physic*, 87:171, 1990.

[41] J. Fürst. A weighted least square scheme for compressible flows. *Flow Turbulence Combust*, 76:331, 2006.

[42] A. Meister. Transonic flow past a RAE 2822 profile (2d). `http://www-ian.math.uni-magdeburg.de/anume/testcase/euler/euler1_e.html`.

[43] P.H. Cook, M.A. McDonald, and M.C.P. Firmin. Aerofoil rae 2822 - pressure distributions, and boundary layer and wake measurements. Technical report, AGARD Report AR 138, 1979.

[44] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202, 1996.

[45] M. R. Visbal and D. V. Gaitonde. Shock capturing using compact-differencing-based methods. In *In AIAA paper 2005-1265, 43rd AIAA Aerospace Sciences Meeting and Exhibit, 10-13, January 2005, Reno, Nevada, Aerospace Center, 370 L'Enfant Promenade, SW, Washington, DC 20024-2518*, page 1265, 2005.

[46] D. Gaitonde and J. Shang. Accuracy of flux-split algorithms in high-speed viscous flows. *AIAA Journal*, 31:1215, 1993.

[47] W. Hayes and R. Probstein. *Hypersonic Flow Theory*, volume 1. Academic Press, New York, 1959.

[48] M.O. Bristeau, R. Glowinski, J. Periaux, and H. Viviand, editors. *Proceedings of the GAMM workshop on the Numerical simulation of compressible Navier-Stokes flows*, Sophia-Antipolis (France), Dec, 1985 1986. INRIA, Vieweg-Verlag.

[49] F. Casalini and A. Dadone. Computations of viscous flows using a multigrid finite volume lambda formulation. *Engineering Computation*, 16:767, 1999.

[50] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131:267, 1997.

[51] V. Venkatakrishnan. Viscous computations using a direct solver. *Computers and Fluids*, 18:191, 1990.

[52] P.A. Forsyth and H. Jiang. Nonlinear iteration methods for high speed laminar compressible Navier-Stokes equations. *Computers and Fluids*, 26:249, 1997.

[53] S. K. Godunov. A difference scheme for numerical solution of discontinuos solution of hydrodynamic equations. *Math. Sbornik*, 47:271, 1959.

[54] P.L. Roe. Approximate riemann solvers, parameter vectors and di?er- ence schemes. *Journal of Computational Physics*, 43:357, 1981.

[55] A. Harten, P.D. Lax, and B. van Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25:35, 1983.

[56] S. R. Chakravarthy and S. Osher. High resolution applications of the Osher upwind scheme for the Euler equations. *In Computational Fluid Dynamics Conference, 6th, Danvers, MA, July 13-15, 1983, Collection of Technical Papers (A83-39351 18-02). New York, AIAA*, 1983:363, 1983.

[57] B. Van Leer. Flux-vector splitting for the Euler equations. *Lecture Notes in Physics*, 170:507, 1982.

[58] M.-S. Liou and C. Steffen. A new flux splitting scheme. *Journal of Computational Physics*, 107:23, 1993.

[59] D. E. Raveh. Maneuver load analysis of over-determined trim systems. In *in 48th AIAA/ASME/ASCE/AHS/ASC Structure, Structural Dynamics, and Material Conference, 23-26 April 2007, Honolulu, Hawaii, AIAA Paper 2007-1985*, page 1985, 2007.

[60] L. Pareschi and G. Russo. Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. *Journal of Scientic Computing*, 25:129, 2005.

[61] G. Naldi and L. Pareschi. Numerical schemes for hyperbolic systems of conservation laws with stiff diffusive relaxation. *SIAM Journal of Numerical Analysis*, 37:1246, 2000.

[62] L. Pareschi. Central differencing based numerical schemes for hyperbolic conservation laws with relaxation terms. *SIAM Journal of Numerical Analysis*, 39:1395, 2001.

[63] R. B. Pember. Numerical methods for hyperbolic conservation laws with stiff relaxation. I. Spurious solutions. *SIAM Journal on Applied Mathematics*, 53:1293, 1993.

[64] R. B. Pember. Numerical methods for hyperbolic conservation laws with stiff relaxation. II. Higher-order Godunov methods. *SIAM Journal on Scientific Computing*, 14:824, 1993.

[65] S. Pieraccini and G. Puppo. Implicit-explicit schemes for BGK kinetic equations. *Journal of Scientific Computing*, 32:1, 2007.

Appendix A

Maple Scripts to Generate f^{eq}

A.1 D2Q13 for isothermal incompressible flows

```
> restart:with(linalg):
> f:=(x,y)->[1,x,y,x^2,y*x,y^2,x^3,x^2*y,x*y^2,y^3,x^4,x^2*y^2,y^4]:
> dc:=array(1..13,1..2,
[[0,0],[1,0],[0,1],[-1,0],[0,-1],[1,1],[-1,1],
[-1,-1],[1,-1],[2,0],[0,2],[-2,0],[0,-2]]):
> A:=Matrix([f(0,0),f(1,0),f(0,1),f(-1,0),f(0,-1),f(1,1),f(-1,1),
f(-1,-1),f(1,-1),f(2,0),f(0,2),f(-2,0),f(0,-2)]):
> B:=inverse(A):
> base_f:=evalm(transpose(B) &*
matrix(13,1,[1,x,y,x^2,y*x,y^2,x^3,x^2*y,x*y^2,y^3,x^4,x^2*y^2,y^4]));
> feq:=array(1..13):
> for i from 1 to 13 do
> feq[i]:=int(rho*subs(x=u+c*cos(alpha),y=v+c*sin(alpha),
base_f[i,1])/(2*Pi), alpha=0..2*Pi):
> end do;
> with(codegen):cost(feq);cost(optimize(feq));
> C(feq,optimized);
```

In our simulation, c was set as 0.85 and $\|v\| = 0.1$.

A.2 D2Q13L2 for inviscid compressible flows

```
> restart:with(linalg):
> f:=(x,y)->[1,x,y,x^2,y*x,y^2,x^3,x^2*y,x*y^2,y^3,x^4,x^2*y^2,y^4]:
> dc:=array(1..13,1..2,
[[0,0],[1,0],[0,1],[-1,0],[0,-1],[1,1],[-1,1],
[-1,-1],[1,-1],[2,0],[0,2],[-2,0],[0,-2]]):
> A:=Matrix([f(0,0),f(1,0),f(0,1),f(-1,0),f(0,-1),f(1,1),f(-1,1),
f(-1,-1),f(1,-1),f(2,0),f(0,2),f(-2,0),f(0,-2)]):
> B:=inverse(A):
> base_f:=evalm(transpose(B) &*
matrix(13,1,[1,x,y,x^2,y*x,y^2,x^3,x^2*y,x*y^2,y^3,x^4,x^2*y^2,y^4]));
> feq:=array(1..13,1..2):eta:=array(1..2):
> for i from 1 to 13 do
> tfeq:=int(rho*subs(x=u+c*cos(alpha),y=v+c*sin(alpha),
base_f[i,1])/(2*Pi), alpha=0..2*Pi):
> feq[i,1]:=tfeq*(1-ep);
> feq[i,2]:=tfeq*ep;
> end do;
> with(codegen):cost(feq);cost(optimize(feq));
> C(feq,optimized);
```

Here, $ep = (2-\gamma)e$ and $c = \sqrt{2(\gamma-1)e}$.

A.3 D1Q5L2 for inviscid compressible flows

```
> restart:
> with(linalg):
```

```
> f:=(x)->[1,x,x^2,x^3,x^4]:
> dc:=array(1..5,[0,d[1],-d[1],d[2],-d[2]]):
> A:=Matrix([f(dc[1]),f(dc[2]),f(dc[3]),f(dc[4]),f(dc[5])]):
> B:=inverse(A):
> base_f:=evalm(transpose(B) &* matrix(5,1,[1,x,x^2,x^3,x^4])):
> feq:=array(1..10):
> for i from 1 to 5 do
tfeq:=simplify((subs(x=u+c,base_f[i,1])+subs(x=u-c,base_f[i,1]))*rho/2):
feq[i+i-1]:=tfeq*(1-ep):
feq[i+i]:=tfeq*ep:
end do:
> with(codegen): C(feq,optimized);
```

Here, $ep = \frac{1}{2}(3-\gamma)e$ and $c = \sqrt{(\gamma-1)e}$. In our simulation, $d[1] = 1$ and $d[2] = 2$.

A.4 D2Q17L2 for viscous compressible flows

```
> restart:with(linalg):
> f:=(x,y)->[1,x,y,x^2,x*y,y^2,x^3,x^2*y,x*y^2,y^3,
x^4,x^3*y,x^2*y^2,x*y^3,y^4,x^5,y^5]:
> dc:=array(1..17,1..2,
[
[0*2/3,0*2/3],
[1*2/3,1*2/3], [-1*2/3,1*2/3], [-1*2/3,-1*2/3], [1*2/3,-1*2/3],
[1*2/3,2*2/3], [2*2/3,1*2/3], [-1*2/3,2*2/3], [-2*2/3,1*2/3],
[-1*2/3,-2*2/3], [-2*2/3,-1*2/3], [1*2/3,-2*2/3], [2*2/3,-1*2/3],
[3*2/3,0*2/3], [0*2/3,3*2/3], [-3*2/3,0*2/3], [0*2/3,-3*2/3]
]);
> A:=Matrix([
f(dc[1,1],dc[1,2]),
```

```
f(dc[2,1],dc[2,2]),
f(dc[3,1],dc[3,2]),
f(dc[4,1],dc[4,2]),
f(dc[5,1],dc[5,2]),
f(dc[6,1],dc[6,2]),
f(dc[7,1],dc[7,2]),
f(dc[8,1],dc[8,2]),
f(dc[9,1],dc[9,2]),
f(dc[10,1],dc[10,2]),
f(dc[11,1],dc[11,2]),
f(dc[12,1],dc[12,2]),
f(dc[13,1],dc[13,2]),
f(dc[14,1],dc[14,2]),
f(dc[15,1],dc[15,2]),
f(dc[16,1],dc[16,2]),
f(dc[17,1],dc[17,2])]);
> B:=inverse(A):
> base_f:=evalm(transpose(B) &*
matrix(17,1,[1,x,y,x^2,x*y,y^2,x^3,x^2*y,x*y^2,y^3,
x^4,x^3*y,x^2*y^2,x*y^3,y^4,x^5,y^5])):

> feq:=array(1..34):
> for i from 1 to 17 do
tfeq:=int((rho/theta)*subs(x=u+c*cos(alpha),y=v+c*sin(alpha),
base_f[i,1])/(2*Pi),alpha=0..2*Pi):
feq[i*2-1]:=tfeq*(1-gamma*ep)+(rho*(gamma-1)/gamma)*subs(x=u,y=v,
base_f[i,1]):
feq[i*2]:=tfeq*gamma*ep:
end do:
```

```
> c:=(gamma*C^2)^(1/2):

> with(codegen): C(feq,optimized);
```

Here, $ep = (2 - \gamma)e$ and $c = \sqrt{2(\gamma - 1)e}$.