

Implicit Schemes for the Lattice Boltzmann Equation

Maksat Ashyraliyev

In partial fulfillment of requirements for the Master of
Computational Science degree

Faculty of Science
Section Computational Science
Universiteit van Amsterdam

Supervised by Dr. Alfons Hoekstra

August 2004

Contents

1	Introduction	1
1.1	From macro to micro	1
1.2	Motivation of project	4
1.3	Literature overview	5
1.4	Outline of contents	9
2	Collision implicit scheme	11
2.1	Theory	11
2.2	Implementation aspects	13
2.2.1	Algorithms	13
2.2.2	Boundary conditions	15
2.2.3	Iterative method	18
2.3	Numerical Experiments	18
2.3.1	2D Poiseuille Channel Flow	18
2.3.2	Backward-Facing Step	22
2.3.3	2D Pulsatile Channel Flow	26
2.4	Conclusions and discussion	28
3	Advection implicit scheme	29
3.1	Theory	29
3.2	Implementation aspects	31
3.3	Numerical Experiments	32
3.4	Conclusions and discussion	33
4	Summary and Future Plans	35
4.1	Summary	35
4.2	Future Plans	35

5	Appendixes	37
5.1	Appendix A	37
5.2	Appendix B	43

Preface

I remember, some time ago, I was sitting on the lecture in Numerical Math and desperately trying to keep myself awake. The professor was giving a lecture 'Implicit schemes for advection equation' and trying to explain how and why an implicit scheme improves the explicit one. I asked the question: whether there is any general statement saying that any implicit scheme gives some advantages over an explicit scheme. I couldn't get an 'explicit' answer to my question, only an 'implicit' one. Since that time the question was always in my mind and when I was offered the project 'Implicit schemes for the Lattice Boltzmann Equation' a year ago I accepted it without any doubts.

Now, before jumping into the world, which combines Mathematics, Physics and Computer Science, called Computational Science, I would like to thank some people.

Dr. Alfons Hoekstra, my scientific adviser. Thank you for all the knowledge and experience you have offered me. Thanks for being patient and not being angry with me, especially when I dropped in your office every now and then without making any appointments.

I would like to thank all members of SCS group with prof. Peter Sloot. Friday SCS meetings were really helpful for me to enlarge the range of my interests.

Special thanks to Drona and Artoli for continuous help during my project. It was always nice to talk to you. You are unlimited sources of information about Lattice Boltzmann world.

These two years of my Master study in Amsterdam were completely supported by Nuffic and Dutch government. Without that I probably would have never been able to study abroad. I would like to thank all people in Dutch consulate in Turkmenistan and people in Nuffic for giving me this chance. Honestly, I am still sometimes thinking that I am dreaming.

Finally, I would like to thank my big family, especially my parents. Mom and dad, I love you!

Chapter 1

Introduction

1.1 From macro to micro

It is well-known that incompressible viscous fluid flow can be described by the Navier-Stokes (NS) equations:

$$\nabla \cdot \mathbf{u} = 0, \quad (1.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u}, \quad (1.2)$$

where \mathbf{u} is the velocity, ρ is the density, p is the pressure and ν is the kinematic viscosity of the fluid.

Indeed, in most of the cases it is difficult to find an exact solution of second order nonlinear partial differential equations (1.1)-(1.2). It is mainly so due to the fact that the required solution cannot be expressed by using only known elementary functions. For this reason a lot of efforts were put to construct different numerical techniques to resolve (1.1)-(1.2), especially with appearance of high-efficiently computers.

Instead of dealing with NS equations, one can go to the kinematic level and look at the Boltzmann equation. Bhatnagar, Gross and Krook in 1954 (see [1]) derived so called Boltzmann BGK equation:

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla f = -\frac{1}{\lambda} (f - f^{(eq)}), \quad (1.3)$$

where $f \equiv f(\mathbf{x}, \xi, t)$ is the single-particle distribution function at position \mathbf{x} in the space with the microscopic velocity ξ at time t , λ is the relaxation time due

to collision and $f^{(eq)}$ is the Maxwell-Boltzmann distribution function given by formula:

$$f^{(eq)} = \frac{\rho}{(2\pi RT)^{D/2}} \exp \left[-\frac{(\boldsymbol{\xi} - \mathbf{u})^2}{2RT} \right], \quad (1.4)$$

in which R is the ideal gas constant, D is the dimension of space, ρ is the macroscopic density, \mathbf{u} is the macroscopic velocity and T is the macroscopic temperature. For an isothermal model temperature T has no physical significance. Macroscopic variables can be calculated as moments of the distribution function f , namely:

$$\rho = \int f d\xi = \int f^{(eq)} d\xi, \quad \rho \mathbf{u} = \int \boldsymbol{\xi} f d\xi = \int \boldsymbol{\xi} f^{(eq)} d\xi \quad (1.5)$$

Moreover, macroscopic variables satisfy NS equations (1.1)-(1.2) in the limit of low Mach number.

The first advantage of going from *macro* level to *micro* level is that now we have to solve ordinary differential equation (1.3) instead of partial differential equations (1.1)-(1.2). However, solving equation (1.3) exactly is also far from trivial due to nonlinear terms. Fortunately, there is another advantage.

By introducing the Lagrangian derivative $D_t \equiv \partial_t + \boldsymbol{\xi} \cdot \nabla$, we can rewrite the continuous Boltzmann BGK equation (1.3) in the form of an ordinary differential equation:

$$D_t f = -\frac{1}{\lambda} (f - f^{(eq)}) \quad (1.6)$$

By formally integrating (1.6) over time t from 0 to Δt , we get:

$$\begin{aligned} & f(\mathbf{x} + \boldsymbol{\xi} \Delta t, \boldsymbol{\xi}, t + \Delta t) - f(\mathbf{x}, \boldsymbol{\xi}, t) \\ &= -\frac{1}{\lambda} \int_0^{\Delta t} \left(f(\mathbf{x} + \boldsymbol{\xi} t', \boldsymbol{\xi}, t + t') - f^{(eq)}(\mathbf{x} + \boldsymbol{\xi} t', \boldsymbol{\xi}, t + t') \right) dt' \end{aligned} \quad (1.7)$$

Assuming that Δt is small enough, $f^{(eq)}$ is smooth enough locally and evaluating the collision integral on the right-hand side of (1.7) explicitly, namely at $t' = 0$, we get

$$f(\mathbf{x} + \boldsymbol{\xi} \Delta t, \boldsymbol{\xi}, t + \Delta t) - f(\mathbf{x}, \boldsymbol{\xi}, t) = -\frac{1}{\tau} \left(f(\mathbf{x}, \boldsymbol{\xi}, t) - f^{(eq)}(\mathbf{x}, \boldsymbol{\xi}, t) \right) \quad (1.8)$$

where $\tau = \frac{\lambda}{\Delta t}$ is the dimensionless relaxation time. By discretizing equation (1.8) in phase space, He and Luo (see [2]) derived the following well-known D2Q9

Lattice BGK model (LBM)

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = -\frac{1}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t) \right), \quad i = 0, \dots, 8 \quad (1.9)$$

where $\mathbf{e}_i (i = 0, \dots, 8)$ are the lattice velocities shown in Figure 1.1.

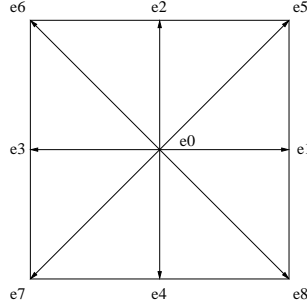


Figure 1.1: Nine velocities for the D2Q9 model.

Accordingly, the hydrodynamic moments of Eqs.(1.5) can be computed by:

$$\rho = \sum_{i=0}^8 f_i = \sum_{i=0}^8 f_i^{(eq)}, \quad \rho \mathbf{u} = \sum_{i=0}^8 \mathbf{e}_i f_i = \sum_{i=0}^8 \mathbf{e}_i f_i^{(eq)}, \quad (1.10)$$

The equilibrium distributions $f_i^{(eq)}$ only depend on local density and velocity:

$$f_i^{(eq)} = w_i \rho \left(1 + 3(\mathbf{e}_i \cdot \mathbf{u}) + \frac{9(\mathbf{e}_i \cdot \mathbf{u})^2}{2} - \frac{3\mathbf{u}^2}{2} \right), \quad i = 0, \dots, 8 \quad (1.11)$$

where weights are:

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

Equation (1.9) is the time evolution of the LBM. It is much more easier to implement such a model than any other numerical method. Despite the simplicity, LBM recovers NS equations (1.1)-(1.2) (see [3]) in the limit of low Mach number with the kinematic viscosity $\nu = (\tau - 0.5)c_s^2 \Delta t$, where the speed of sound for D2Q9 model $c_s = \frac{1}{\sqrt{3}}$. This was shown by applying a Chapman-Enskog analysis. Moreover, there is a big opportunity for parallel computing due to the locality of collision and streaming operations.

By now we briefly introduced three main objects we have to deal with: NS equations, LBM and Boltzmann BGK equation. We summarize the relations between them in Figure 1.2.

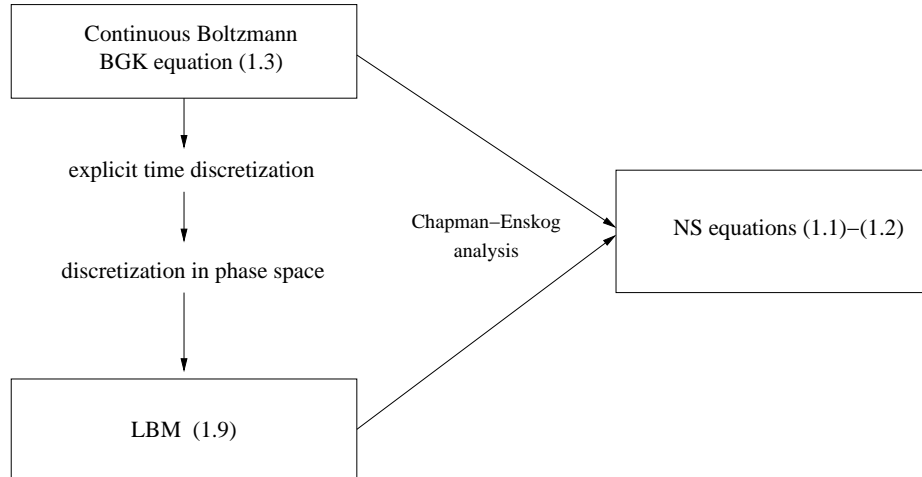


Figure 1.2: NS equations, LBM and Boltzmann BGK equation.

1.2 Motivation of project

In this project we study implicit schemes for Boltzmann equation (1.3). First of all, let us briefly outline the main drawbacks of LBM. It is well-known that the explicit scheme (1.9) becomes numerically unstable as $\tau \rightarrow \frac{1}{2}$ (see [4]), which basically means low viscosity. We also know that the simulation time step Δt in LBM is severely restricted. It is due to the CFL restriction. Since the LBM is the explicit discretization of the equation (1.3), the CFL number for stability reasons should be equal to 1. This couples the simulation time step and grid size.

So, by using the LBM a large number of iterations is required for convergence. It makes the simulations computationally expensive. This is really a crucial point, especially in simulations of time dependent flows. So, the motivation of the project is to find implicit schemes to solve (1.3) which allow us to

- expand the parameter space of LBM, i.e. simulate low-viscosity flows;
- increase the simulation time step Δt .

There are two terms in the continuous Boltzmann equation (1.3) : advection and collision. Depending on how we treat this terms, explicitly or implicitly, we can get a variety of semi-implicit and implicit schemes. By discretising the collision term implicitly, we have collision semi-implicit schemes and by discretising the advection term implicitly, we have advection semi-implicit schemes. By discretising both terms terms implicitly, we obtain fully implicit schemes.

Since there is a relation between the relaxation time τ and viscosity ν and the relaxation time is present only in the collision term, we dare to make our first hypothesis: collision-implicit scheme may allow us to expand the parameter space of LBM, i.e. to simulate low-viscosity flows.

Another important issue is that the time for advection is much larger than the time for collision. So, we make the second hypothesis: an advection-implicit scheme may allow us to increase the time step Δt .

In this project we derive a number of implicit schemes and numerically verify whether these hypotheses are correct or not. Whatever scheme we use to resolve (1.3), we also have to keep in mind a possibility for parallel computing:

”A good algorithm is not simply the one that converges faster but also the one that is parallelizable”

G.E.Karniadakis, R.M.Kirby (2003)

1.3 Literature overview

Some relevant works have appeared recently. In [5-8] the implicit approach was introduced for steady form, i.e. time independent flows. In this project we are more interested on time dependent case. In [9-12] some new implicit schemes for time dependent case were proposed. Let us give a brief overview of these works.

By evaluating the integral on the right-hand side of (1.7) implicitly by using the trapezoidal method and discretizing it in momentum space and configuration space, Sankaranarayanan, Shan, Kevrekidis and Sundaresan (see [9]) obtained:

$$\begin{aligned}
 & f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) \\
 = & -\frac{1}{\tau} \left(\frac{f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t)}{2} + \frac{f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i^{(eq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t)}{2} \right)
 \end{aligned}
 \tag{1.12}$$

(1.12) coupled with (1.10)-(1.11) is the collision implicit scheme which has second-order accuracy both in time and space. Implementation of this scheme requires iterative calculations. However, these iterations are local in nature. It becomes obvious if we rewrite (1.12) as

$$f_i(\mathbf{x}, t + \Delta t) - f_i(\mathbf{x} - \mathbf{e}_i \Delta t, t)$$

$$= -\frac{1}{\tau} \left(\frac{f_i(\mathbf{x} - \mathbf{e}_i \Delta t, t) - f_i^{(eq)}(\mathbf{x} - \mathbf{e}_i \Delta t, t)}{2} + \frac{f_i(\mathbf{x}, t + \Delta t) - f_i^{(eq)}(\mathbf{x}, t + \Delta t)}{2} \right)$$

So, iterations at each node can be done independent of other nodes. It was shown that (1.12) reduces to NS equations in the low Mach number limit, with the kinematic viscosity $\nu = \tau c_s^2 \Delta t$. A two-component system was considered and the collision implicit scheme (1.12) was applied for each component. A number of computational experiments comparing LBM and collision implicit scheme were performed. The authors found that the collision implicit scheme (1.12) could be solved by simple successive substitution (fixed point iteration) in no more than three substitution steps. This added some computational burden to each time step over that for the LBM. The premium is the stability of the collision implicit scheme for small kinematic viscosity ν at high Reynolds number. However, the possibility to increase the simulation time step was not discussed by the authors.

The continuous Boltzmann BGK equation (1.3) discretized in momentum space has the following form:

$$\frac{\partial f_i}{\partial t} + \mathbf{e}_i \cdot \nabla f_i = -\frac{1}{\lambda} [f_i - f_i^{(eq)}] \quad (1.13)$$

Equation (1.13) is called the Discrete Boltzmann equation (DBE).

By using a first order upwind space discretization for the advection term

$$\mathbf{e}_i \cdot \nabla f_i(\mathbf{x}, t) = \frac{f_i(\mathbf{x}, t) - f_i(\mathbf{x} - \mathbf{e}_i \Delta \mathbf{x}, t)}{\Delta \mathbf{x}},$$

by replacing the time derivative $\frac{\partial f_i(\mathbf{x}, t)}{\partial t}$ with a first order time difference, and by evaluating the collision term implicitly at $t + \Delta t$, Cao, Chen, Jin and Martinez (see [10]) obtained the following collision semi-implicit scheme

$$f_i(\mathbf{x}, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{\Delta t}{\Delta \mathbf{x}} [f_i(\mathbf{x}, t) - f_i(\mathbf{x} - \mathbf{e}_i \Delta \mathbf{x}, t)] - \frac{\Delta t}{\lambda} \left(f_i(\mathbf{x}, t + \Delta t) - f_i^{(eq)}(\mathbf{x}, t + \Delta t) \right) \quad (1.14)$$

Scheme (1.14) was only proposed by the authors but not implemented. Moreover, the accuracy in time and space of this scheme was not discussed. Obviously, it also requires solving a system of nonlinear equations on each node.

By taking the advection term implicitly and keeping the collision term explicit, Seta and Takahashi (see [11]) derived the following advection semi-implicit finite difference scheme

$$\begin{aligned} & \frac{f_i(\mathbf{x}, t + \Delta t) - f_i(\mathbf{x}, t)}{\Delta t} + (\mathbf{e}_i)_\alpha \frac{f_i(\mathbf{x}_\alpha, t + \Delta t) - f_i(\mathbf{x}_\alpha - \Delta \mathbf{x}_\alpha, t + \Delta t)}{\Delta \mathbf{x}_\alpha} \\ &= -\frac{1}{\lambda} \left(f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t) \right), \quad (\mathbf{e}_i)_\alpha \geq 0 \end{aligned} \quad (1.15)$$

$$\begin{aligned} & \frac{f_i(\mathbf{x}, t + \Delta t) - f_i(\mathbf{x}, t)}{\Delta t} + (\mathbf{e}_i)_\alpha \frac{f_i(\mathbf{x}_\alpha + \Delta \mathbf{x}_\alpha, t + \Delta t) - f_i(\mathbf{x}_\alpha, t + \Delta t)}{\Delta \mathbf{x}_\alpha} \\ &= -\frac{1}{\lambda} \left(f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t) \right), \quad (\mathbf{e}_i)_\alpha < 0 \end{aligned} \quad (1.16)$$

where α denotes the Cartesian component of a vector. The authors only analyzed the numerical stability of scheme (1.15) - (1.16) by means of a von Neumann stability analysis.

At this point, we should note that a closer look at schemes (1.14) and (1.15)-(1.16) by applying the Chapman-Enskog analysis reveals that they recover accurate physics only at steady state (see [12]). Obviously, this is not what we want. We would like to derive implicit schemes which are able to simulate both time-dependent and time-independent flows.

Different numerical techniques to derive implicit schemes for DBE were introduced by Lee and Lin (see [12]). DBE (1.13) was discretized along characteristics by using the θ -method.

$$\begin{aligned} f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) &= -\frac{1 - \theta_c}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t) \right) \\ &\quad - \frac{\theta_c}{\tau} \left(f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i^{(eq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) \right) \end{aligned} \quad (1.17)$$

where $\theta_c \in [0, 1]$. To solve (1.17) the solution procedure was split into the following steps:

- Pre-streaming collision step

$$\tilde{f}_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) - \frac{1 - \theta_c}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t) \right) \quad (1.18)$$

- Streaming step

$$\tilde{f}_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = \tilde{f}_i(\mathbf{x}, t) \quad (1.19)$$

- Post-streaming collision step

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = \tilde{f}_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - \frac{\theta_c}{\tau} \left(f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i^{(eq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) \right) \quad (1.20)$$

Equation (1.20) can be rearranged as

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = \frac{\tilde{f}_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) + \frac{\theta_c}{\tau} f_i^{(eq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t)}{1 + \frac{\theta_c}{\tau}} \quad (1.21)$$

(1.21) requires solving a system of nonlinear equations on each node. To avoid it, the authors used the following approximation: the macroscopic density and velocity used in calculation of $f_i^{(eq)}$ at time $t + \Delta t$ in (1.21) were obtained by taking moments of \tilde{f}_i at time $t + \Delta t$

$$\rho = \sum_i \tilde{f}_i, \quad \rho \mathbf{u} = \sum_i \mathbf{e}_i \tilde{f}_i$$

In the conventional LBM the grid points coincide with lattice points and the streaming step expressed by equation (1.19) becomes a perfect shift. The Courant-Friedrichs-Lewy (CFL) number for perfect shift equals unity, which couples grid distance and time step. As a result, large time step is achieved at the expense of spatial resolution. To get rid off this fact, the authors expressed equations (1.18)-(1.20) in an Eulerian framework as is shown below:

- Pre-streaming collision step

$$\tilde{f}_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) - \frac{1 - \theta_c}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t) \right) \quad (1.22)$$

- Streaming step

$$\frac{\partial \tilde{f}_i}{\partial t} + \mathbf{e}_i \cdot \nabla \tilde{f}_i = 0 \quad (1.23)$$

- Post-streaming collision step

$$f_i(\mathbf{x}, t + \Delta t) = \tilde{f}_i(\mathbf{x}, t + \Delta t) - \frac{\theta_c}{\tau} \left(f_i(\mathbf{x}, t + \Delta t) - f_i^{(eq)}(\mathbf{x}, t + \Delta t) \right) \quad (1.24)$$

Here \mathbf{x} is defined at grid points not necessarily lining up along characteristics. Streaming step equation (1.23) can be resolved by any second-order accurate scheme suited for the pure advection equation to replace the perfect shift on uniform lattice.

Implicit Taylor-Galerkin finite element method was used to solve (1.23)

$$\frac{\tilde{f}_i^{n+1} - \tilde{f}_i^n}{\Delta t} = -(\mathbf{e}_i)_s \frac{\partial \tilde{f}_i^n}{\partial x_s} + \frac{\Delta t}{2} (\mathbf{e}_i)_r \frac{\partial}{\partial x_r} \left[(\mathbf{e}_i)_s \frac{\partial}{\partial x_s} (\theta_a \tilde{f}_i^{n+1} + (1 - \theta_a) \tilde{f}_i^n) \right] \quad (1.25)$$

where $\theta_a \in [0, 1]$. Scheme (1.25) is explicit for $\theta_a = 0$ and implicit for $0 < \theta_a \leq 1$.

Numerical experiments were performed to compare explicit and implicit schemes in terms of stability and time step. The main conclusion of the authors was that while collision implicit scheme ($0 < \theta_c \leq 1$) has no advantage over collision explicit scheme ($\theta_c = 0$) in terms of time step, the advection implicit scheme ($0 < \theta_a \leq 1$) can indeed allow a larger time step than the advection explicit scheme ($\theta_a = 0$).

1.4 Outline of contents

Let us give a brief outline of contents of this thesis.

In Chapter 2 we further investigate the collision implicit scheme. The scheme will be directly derived from the continuous Boltzmann BGK equation (1.3). In Appendix A we will provide a Chapman-Enskog analysis of the scheme to show that it recovers NS equations (1.1)-(1.2) when the proper relation between relaxation time and kinematic viscosity is chosen. We will give a detailed description of the implementation aspects of the collision implicit scheme, such as algorithm, boundary conditions and iterative method. We will use the Newton-Raphson iterative method to resolve the collision implicit scheme. The reader can find a bit more information about it in Appendix B. Numerical experiments for steady 2D Poiseuille Channel flow, for flow over Backward-Facing step and for unsteady 2D Pulsatile Channel flow will be conducted to compare the collision implicit scheme with common LBM.

In Chapter 3 we will study the advection implicit scheme. The scheme will be directly derived from DBE (1.13). Numerical experiments for 2D Poiseuille Channel flow will be conducted for different CFL numbers.

In Chapter 4 we will give a brief summary of the thesis and outline future plans.

Chapter 2

Collision implicit scheme

2.1 Theory

We derive the collision implicit scheme directly from the continuous Boltzmann BGK equation. First, we rewrite (1.7) as

$$\begin{aligned} & f(\mathbf{x} + \xi \Delta t, \xi, t + \Delta t) - f(\mathbf{x}, \xi, t) \\ &= -\frac{1 - \theta_c}{\lambda} \int_0^{\Delta t} \left(f(\mathbf{x} + \xi t', \xi, t + t') - f^{(eq)}(\mathbf{x} + \xi t', \xi, t + t') \right) dt' \\ & \quad - \frac{\theta_c}{\lambda} \int_0^{\Delta t} \left(f(\mathbf{x} + \xi t', \xi, t + t') - f^{(eq)}(\mathbf{x} + \xi t', \xi, t + t') \right) dt' \end{aligned} \quad (2.1)$$

where $\theta_c \in [0, 1]$.

By evaluating the first integral on the right-hand side of (2.1) explicitly, namely at $t' = 0$ and evaluating the second integral on the right-hand side of (2.1) implicitly, namely at $t' = \Delta t$, we get

$$\begin{aligned} f(\mathbf{x} + \xi \Delta t, \xi, t + \Delta t) - f(\mathbf{x}, \xi, t) &= -\frac{1 - \theta_c}{\tau} \left(f(\mathbf{x}, \xi, t) - f^{(eq)}(\mathbf{x}, \xi, t) \right) \\ & \quad - \frac{\theta_c}{\tau} \left(f(\mathbf{x} + \xi \Delta t, \xi, t + \Delta t) - f^{(eq)}(\mathbf{x} + \xi \Delta t, \xi, t + \Delta t) \right) \end{aligned} \quad (2.2)$$

By discretizing equation (2.2) in the phase space in the same way as it was done by He and Luo (see [2]), we derive the following implicit scheme

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

$$-\frac{\theta_c}{\tau} \left(f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i^{(eq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) \right) \quad (2.3)$$

We call (2.3) coupled with (1.10)-(1.11) collision implicit scheme for D2Q9 model. Note that by putting $\theta_c = 0$ in (2.3), we get LBM (1.9), and that by putting $\theta_c = \frac{1}{2}$, we get the collision implicit scheme (1.12).

Scheme (2.3) has second order accuracy both in space and time and recovers NS equations (1.1)-(1.2) in the limit of low Mach number when the relaxation parameter satisfies

$$\nu = (\tau - 0.5 + \theta_c) c_s^2 \Delta t \quad (2.4)$$

Derivation of (2.4) is given in Appendix A. We summarize all in Figure 2.1.

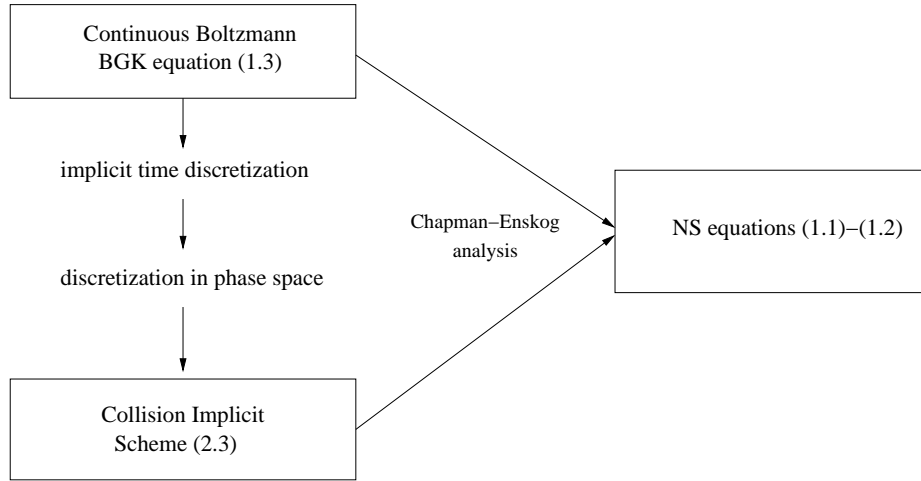


Figure 2.1: NS equations, Boltzmann BGK equation and Collision Implicit Scheme.

Let us try to analyse whether the implicit scheme may have advantages over LBM or not. Since for LBM we have $\nu = (\tau - 0.5) c_s^2 \Delta t$, we rewrite (1.9) as

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{1}{\frac{\nu}{c_s^2 \Delta t} + 0.5} \left(f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t) \right) \quad (2.5)$$

Now we look at the collision implicit scheme. By rearranging the terms in (2.3), we obtain

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

$$+ \frac{\theta_c}{\tau + \theta_c} \left(f_i^{(eq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i^{(eq)}(\mathbf{x}, t) \right) \quad (2.6)$$

By using (2.4), (2.6) gets the following form

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{1}{\frac{\nu}{c_s^2 \Delta t} + 0.5} \left(f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t) \right) + \frac{\theta_c}{\tau + \theta_c} \left(f_i^{(eq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i^{(eq)}(\mathbf{x}, t) \right) \quad (2.7)$$

The difference between (2.5) and (2.7) is only in last term of (2.7). By using Taylor expansion, we have

$$f_i^{(eq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i^{(eq)}(\mathbf{x}, t) = \Delta t \left(\frac{\partial f_i^{(eq)}}{\partial t} + \mathbf{e}_i \cdot \nabla f_i^{(eq)} \right) + O(\Delta t^2) \quad (2.8)$$

By neglecting the term $O(\Delta t^2)$ in (2.8) and using Lagrangian derivative, we obtain

$$f_i^{(eq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i^{(eq)}(\mathbf{x}, t) = \Delta t D_t f_i^{(eq)}$$

For time-independent flows derivative in time is equal to zero. So, the last term in (2.7) does not affect stability at all and the collision implicit scheme has the same stability range as LBM.

For time-dependent flows the last term in (2.7) has order $O(\Delta t)$. It might not be essential for stability range of the scheme. The answer will follow from our numerical experiments.

We conduct a number of simulations by using both LBM (1.9) and collision implicit scheme (2.3) for steady 2D Poiseuille channel flow, flow over a Backward-Facing Step (BFS) and unsteady 2D Pulsatile channel flow, to verify whether collision implicit scheme (2.3) allows us to expand the parameter space of LBM or not.

2.2 Implementation aspects

2.2.1 Algorithms

Before we start our simulations let us discuss implementation aspects. First of all, the following pseudo code is used for the explicit LBM:

```

Start Program
Initialize lattice
Set initial density and velocity
Set initial ditribution
Do {
Move fluid particles
Apply boundary conditions
Calculate density and velocity by using (1.10)
Calculate equilibrium distributions by using (1.11)
Collide fluid particles
} While(condition for convergence is not satisfied)
Write data to file
End Program

```

For the collision implicit scheme we use the following pseudo code:

```

Start Program
Initialize lattice
Set initial density and velocity
Set initial ditribution
Do {
Resolve the system of nonlinear equations on each node
Apply boundary conditions
Calculate density and velocity by using (1.10)
Calculate equilibrium distributions by using (1.11)
} While(condition for convergence is not satisfied)
Write data to file
End Program

```

A few words about these pseudo codes. Initially, for both LBM and collision implicit scheme, on each node we set the density ρ and velocity components u and v to some initial values. Using these values, we set the initial distributions f_i to the equilibrium distributions $f_i^{(eq)}$ evaluated by (1.11).

The simulation loop continues until the condition for convergence is no longer satisfied. In all simulations for convergence the following condition is used:

$$\frac{\sum_{(i,j)} \sqrt{(u(i,j) - \bar{u}(i,j))^2 + (v(i,j) - \bar{v}(i,j))^2}}{\sum_{(i,j)} \sqrt{\bar{u}(i,j)^2 + \bar{v}(i,j)^2}} < \varepsilon, \quad (2.9)$$

where above sums are over all fluid nodes (i, j) ; $\bar{u}(i, j)$ and $\bar{v}(i, j)$ are values of velocity components on the previous time step. Basically, (2.9) means convergence in l_2 norm.

2.2.2 Boundary conditions

One of the most crucial questions in implementation is the way of applying boundary conditions. Basically, there are two different kind of boundaries: walls and inlet-outlet.

No-slip boundary condition

On the walls we consider no-slip boundary condition, which means zero fluid velocity there. In our simulations we use two different implementations: on-grid and mid-grid.

Mid-grid bounce back boundary condition for the node on the North wall is shown in Figure 2.2. Filled nodes are fluid nodes, empty nodes are solid nodes and dashed line indicates the wall. After the streaming operation some populations are out of our fluid area. We should revert them back into the fluid. So, populations 7, 4, 8 on the adjacent to the wall fluid nodes receive populations 5, 2, 6, respectively, from the solid node on the wall. The similar procedure is applied for the nodes on the South wall.

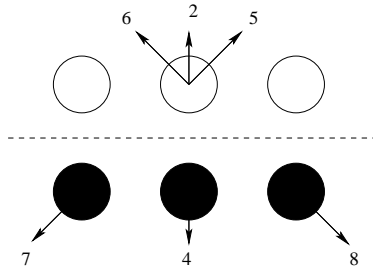


Figure 2.2: Mid-grid bounce back, no-slip boundary condition.

On-grid bounce back boundary condition is shown in Figure 2.3. It is well-known that directly reverting the populations into the fluid gives only first order accuracy. We use a slightly different way, proposed by Zou and He (see [13]). Unknown populations for the node on the North wall are defined in the following way:

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

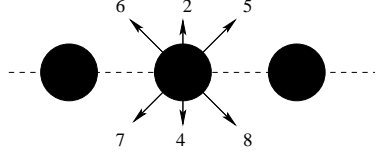


Figure 2.3: On-grid bounce back, no-slip boundary condition.

These assure the zero fluid velocity on the node:

$$\rho u = f_1 + f_5 + f_8 - f_3 - f_6 - f_7 = 0, \quad \rho v = f_2 + f_5 + f_6 - f_4 - f_7 - f_8 = 0$$

Unknown populations for the node on the South wall are defined in a similar way:

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

We will call this way ”modified on-grid bounce back no-slip boundary condition”. This way adds some extra computation but gives the second order accuracy.

Inlet-outlet boundary conditions

For inlet-outlet boundary conditions there are two common ways: periodic boundary condition and pressure difference boundary condition.

Periodic boundary condition is easy to implement by assigning the incoming populations on the inlet to the corresponding outgoing populations on the outlet and assigning the incoming populations on the outlet to the corresponding outgoing populations on the inlet. Despite the simplicity, it perfectly fits the simulations of fluid flow on symmetric geometries, such as a simple 2D channel. However, for complex asymmetric geometries, such as a Backward-facing step, we cannot use the periodic boundary condition.

In all numerical simulations we apply pressure difference boundary condition. To implement it we use the way proposed by Zou and He (see [13]). Let us briefly describe it here.

On the inlet we define the parabolic velocity profile for the velocity component u in x -direction, while the velocity component v in y -direction is assigned to zero. The density and unknown incoming populations are calculated in the following

way:

$$\begin{aligned}\rho &= \frac{f_0 + f_2 + f_4 + 2(f_3 + f_6 + f_7)}{1 - u} \\ f_1 &= f_3 + \frac{2}{3}\rho u \\ f_5 &= f_7 + \frac{1}{6}\rho u - \frac{1}{2}(f_2 - f_4) \\ f_8 &= f_6 + \frac{1}{6}\rho u + \frac{1}{2}(f_2 - f_4)\end{aligned}$$

On the outlet we define the density ρ , i.e. the pressure is fixed. Moreover, we assume that $v = 0$. Then unknowns are calculated as:

$$\begin{aligned}u &= \frac{f_0 + f_2 + f_4 + 2(f_1 + f_5 + f_8)}{\rho} - 1 \\ f_3 &= f_1 - \frac{2}{3}\rho u \\ f_6 &= f_8 - \frac{1}{6}\rho u - \frac{1}{2}(f_2 - f_4) \\ f_7 &= f_5 - \frac{1}{6}\rho u + \frac{1}{2}(f_2 - f_4)\end{aligned}$$

This approach we call velocity - pressure boundary condition (VPB).

If the density is given both on the inlet and on the outlet, we have pressure - pressure boundary condition (PPB). Velocity for (PPB) on the inlet is calculated by:

$$u = 1 - \frac{f_0 + f_2 + f_4 + 2(f_3 + f_6 + f_7)}{\rho}$$

All other unknowns are calculated exactly in the same way as for VPB.

In numerical simulations we also use another way, which we call velocity - free flux boundary condition. On the inlet we define the parabolic velocity profile and calculate the unknowns in the same way as for VPB. On the outlet we assume that $\frac{\partial u}{\partial x} = 0$, which basically means free flux. This can be implemented just by copying the values from the second last nodes to the corresponding outlet nodes. The length of the channel should be long enough for this assumption. Then unknowns are calculated again in the same way as for VPB.

2.2.3 Iterative method

To resolve (2.3) we use an iterative method. First, we rewrite it in the following form:

$$\begin{aligned} & f_i(\mathbf{x}, t + \Delta t) - \frac{\theta_c}{\tau + \theta_c} f_i^{(eq)}(\mathbf{x}, t + \Delta t) \\ &= \left(1 - \frac{1}{\tau + \theta_c}\right) f_i(\mathbf{x} - \mathbf{e}_i \Delta t, t) + \frac{1 - \theta_c}{\tau + \theta_c} f_i^{(eq)}(\mathbf{x} - \mathbf{e}_i \Delta t, t) \end{aligned} \quad (2.10)$$

Note that all terms on right-hand side of (2.10) are known. By applying (2.10) for all i on lattice node, we get the system of nine nonlinear equations for our collision implicit scheme. However, it is clear that the system of nonlinear equations on each node can be resolved independently. This retains the possibility for parallel computing. To solve the system of nonlinear equations we use the Newton-Raphson (NR) iterative method. We describe the implementation of NR iterative solver in Appendix B.

2.3 Numerical Experiments

2.3.1 2D Poiseuille Channel Flow

Theory

We consider Poiseuille flow in a 2D channel. The geometry is shown in Figure 2.4. Velocity of fluid is $\mathbf{V} = (u, v)$; u and v are the velocity components in the x -direction and y -direction, respectively.

For the steady flow in a tube we have $u = u(y)$, ($0 \leq y \leq H$) and $v = 0$, where H is the height of the channel. Equation of continuity (1.1) is satisfied identically. From equation (1.2) we obtain

$$\frac{\partial P}{\partial y} = 0, \quad (2.11)$$

$$-\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \frac{d^2 u}{dy^2} = 0 \quad (2.12)$$

Equation (2.11) shows that the pressure is independent on y , i.e. it is constant along the y -axis. By differentiating (2.12) with respect to x , we get $\frac{\partial^2 P}{\partial x^2} = 0$, which means that $\frac{\partial P}{\partial x} = \text{const}$. Then from (2.12) we obtain

$$\frac{d^2 u}{dy^2} = \frac{1}{\rho \nu} \frac{\partial P}{\partial x} \equiv \text{const}$$

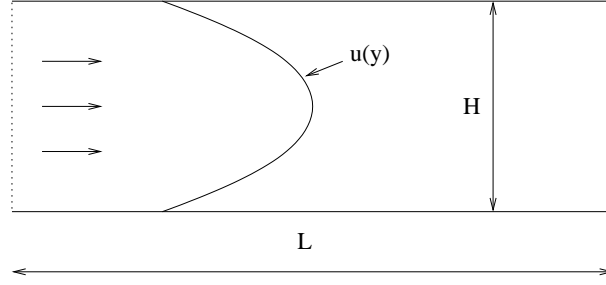


Figure 2.4: 2D Channel Flow.

By using boundary conditions $u(0) = u(H) = 0$, we can trivially find a solution for last equation:

$$u = -\frac{1}{2\rho\nu} \frac{\partial P}{\partial x} y(H-y), \quad 0 \leq y \leq H \quad (2.13)$$

So, the velocity component u has a parabolic profile with maximum value on the center of tube

$$u_{max} = u(H/2) = -\frac{H^2}{8\rho\nu} \frac{\partial P}{\partial x} \quad (2.14)$$

By combining (2.14) and (2.13), we obtain the analytical solution of 2D Poiseuille channel flow

$$u = \frac{4u_{max}}{H^2} y(H-y), \quad v = 0, \quad 0 \leq y \leq H \quad (2.15)$$

Since $\frac{\partial P}{\partial x} \equiv const$, it can be replaced by $\frac{P_{outlet} - P_{inlet}}{L}$, where L is the length of the channel. Then from (2.14) we get

$$u_{max} = \frac{H^2(P_{inlet} - P_{outlet})}{8L\rho\nu}. \quad (2.16)$$

Simulations

We conduct a number of experiments by using both LBM (1.9) and the collision implicit scheme (2.3) for the 2D Poiseuille flow. In all simulations for tube flow we use the following conditions:

- The length of the tube is twice larger than the height, so $L = 2H$;
- Initial density is $\rho_0 = 1.0$;
- Initial velocity components are $u_0 = 0.01$ and $v_0 = 0.0$;

- VPB is used for inlet-outlet and modified on-grid bounce back no-slip boundary condition is used for walls;
- Density at the outlet is $\rho_{out} = 1.0$;
- In convergence condition (2.9) we use $\epsilon = 10^{-12}$;
- $\theta = 0.5$ is chosen in (2.3);
- Relative error E is measured as

$$E = \frac{\sum_{(i,j)} (|u(i,j) - \bar{u}(i,j)| + |v(i,j) - \bar{v}(i,j)|)}{\sum_{(i,j)} (|\bar{u}(i,j)| + |\bar{v}(i,j)|)}, \quad (2.17)$$

where above sums are over all fluid nodes (i, j) ; $\bar{u}(i, j)$ and $\bar{v}(i, j)$ are analytical solutions defined by (2.15).

Experiment 1

Relaxation time is $\tau = 1.0$ for LBM and $\tau = 0.5$ for collision implicit scheme, so viscosity $\nu = 1/6$ in both cases. We conduct simulations with different lattice sizes $N_x \times N_y = 20 \times 10, 40 \times 20, 80 \times 40, 160 \times 80$. Centreline velocity in the tube is chosen $u_{max} = 0.1, 0.05, 0.025, 0.0125$, accordingly. So, Reynolds number $Re = \frac{u_{max}H}{\nu} = 6$ is fixed in all cases.

Experiment 2

Relaxation time is $\tau = 0.6$ for LBM and $\tau = 0.1$ for collision implicit scheme, so viscosity $\nu = 1/30$ in both cases. We conduct simulations with different lattice sizes $N_x \times N_y = 20 \times 10, 40 \times 20, 80 \times 40, 160 \times 80$. Centreline velocity in the tube is chosen $u_{max} = 0.1, 0.05, 0.025, 0.0125$, accordingly. So, Reynolds number $Re = \frac{u_{max}H}{\nu} = 30$ is fixed in all cases.

Results

We performed our first experiments to study the convergence of both the LBM and the collision implicit scheme. Relative errors as the function of lattice size using the LBM and the collision implicit scheme are shown in Figure 2.5. We see the same error behaviour, which is in agreement with analysis we made in

theoretical part of this chapter. Since the lines have slope -2, both schemes have second order convergence. We also observed that the number of simulation time

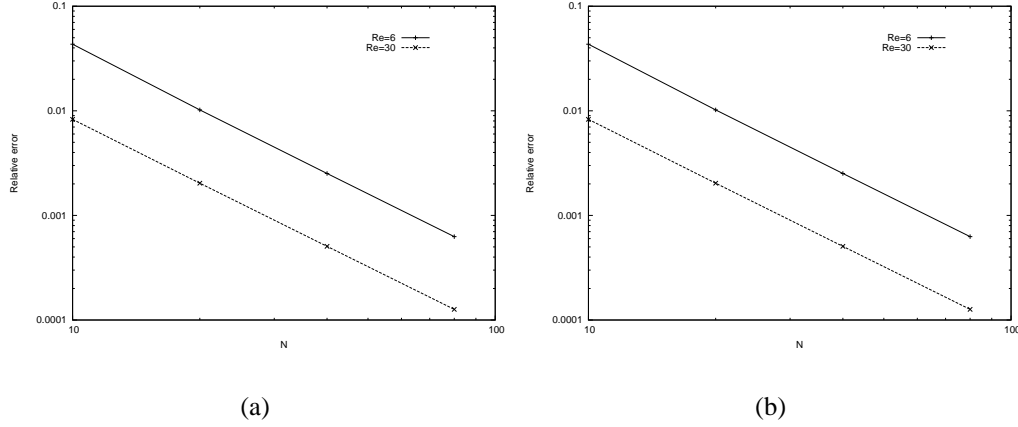


Figure 2.5: Relative errors for $Re = 6$ and $Re = 30$ using the LBM (a) and using the collision implicit scheme (b).

steps, it takes for convergence, is the same in both cases. This is not surprising for us because in derivation of collision implicit scheme we assumed that Δt is the same as it was for LBM. We measured the total simulation time for both schemes. It turned out that the collision implicit scheme is approximately 12 times more expensive than LBM. In all cases we obtain a fully developed parabolic velocity profile at the outlet of the tube. Velocities normalised by u_{max} for $Re = 30$ and lattice size $N_x \times N_y = 80 \times 40$ using both LBM and the collision implicit scheme are shown in Figure 2.6. From these plots we see that simulation results perfectly fit the analytical solution (solid line).

Experiment 3

Relaxation time is $\tau = 0.56$ for LBM and $\tau = 0.06$ for collision implicit scheme, so viscosity $\nu = 0.02$ in both cases. We conduct simulations with lattice size $N_x \times N_y = 80 \times 40$. Centreline velocity in the tube is chosen $u_{max} = 0.1$ in both cases. So, Reynolds number $Re = \frac{u_{max}H}{\nu} = 200$ and the Mach number $M_a = \frac{u_{max}}{c_s} = 0.173205 \ll 1$.

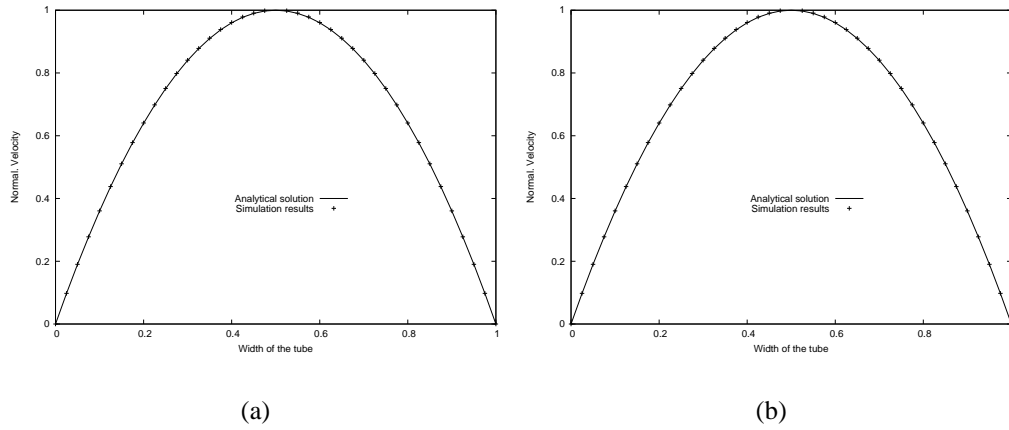


Figure 2.6: Normalised velocity profiles at the outlet of the channel for $Re = 30$ and lattice size $N_x \times N_y = 80 \times 40$ using the LBM (a) and using the collision implicit scheme (b).

Results

Both schemes are unstable. Note that higher Reynolds numbers can be achieved by increasing the lattice size.

Conclusion

From results of numerical experiments for the 2D Poiseuille Channel Flow we confirm that the collision implicit scheme does not allow us to expand the parameter space of the LBM for steady flows.

2.3.2 Backward-Facing Step

Model

Now we conduct simulation of flow over a Backward-Facing Step (BFS) by using LBM (1.9) and collision implicit scheme (2.3). The BFS model, which is the channel with sudden enlargement, is shown in Figure 2.7. It is well-known that in the laminar flow over BFS there is a recirculation zone $0 < x < r$ with the negative velocity u . At $x = r$ velocity component u changes sign and becomes positive. We define the reattachment length as $x_r = \frac{r}{h}$. The values of reattachment length were

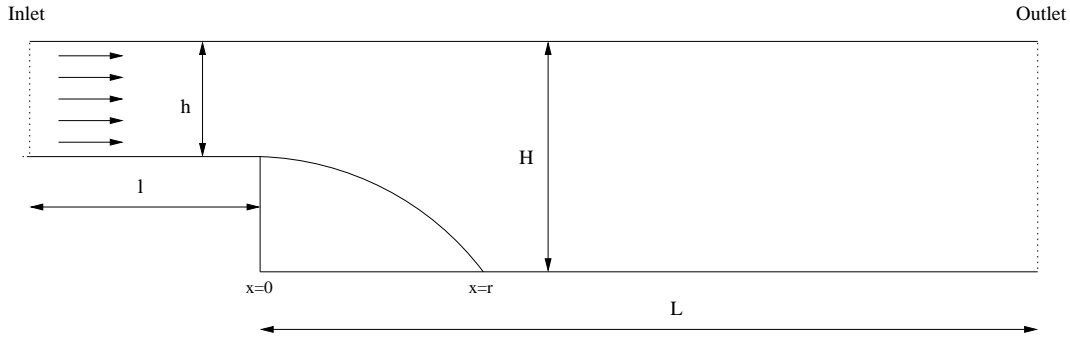


Figure 2.7: Backward-Facing Step.

measured experimentally by Armaly, Durst, Pereira, and Schonung (see [14]).

Simulations

In all simulations of flow over a Backward-Facing Step we use the following conditions:

- The inlet channel length is $l = 5h$ and the channel length is $L = 21h$; the ratio between the step height h and the channel height H is $23/47$; the number of lattice nodes on the inlet is 23;
- Initial mass density $\rho_0 = 1.0$;
- Initial velocity $u_0 = 0.01$; $v_0 = 0.0$ are defined on all lattice nodes;
- For inlet - outlet we use velocity - free flux boundary condition and for walls we use mid-grid bounce back no-slip boundary condition;
- Parabolic velocity profile defined on the inlet has centreline velocity $u_{max} = 0.1$, so the Mach number $Ma = \frac{u_{max}}{c_s} = 0.173205 \ll 1$;
- $\varepsilon = 10^{-7}$ is used in convergence condition (2.9);
- $\theta_c = 0.5$ is used in (2.3).

Experiment 1

Relaxation time is $\tau = 0.692$ for LBM and $\tau = 0.192$ for collision implicit scheme. So, viscosity $\nu = 0.064$ and the Reynolds number $Re = \frac{4u_{max}(H-h)}{3\nu} = 50$ are the same in both cases.

Results

Convergence condition is satisfied after 30000 simulation time steps in both simulations. Reattachment length obtained by both schemes is the same $x_r = 1.7391$.

Experiment 2

Relaxation time is $\tau = 0.596$ for LBM and $\tau = 0.096$ for the collision implicit scheme. So, viscosity $\nu = 0.032$ and the Reynolds number $Re = 100$ are the same in both cases.

Results

Convergence condition is satisfied after 41800 simulation time steps in both simulations. Reattachment length obtained by both schemes is the same $x_r = 2.9565$. For both cases we measure the velocity at four different places in the channel $x = \frac{L}{4}$, $x = \frac{L}{3}$, $x = \frac{L}{2}$, $x = L$ and normalise it by u_{max} . In Figure 2.8 there are normalised velocity profiles by using LBM and collision implicit scheme.

From the plots we observe that both schemes produce the same velocity profiles. At the outlet of the channel for both cases we have a fully developed parabolic velocity profile.

Experiment 3

Relaxation time is $\tau = 0.564$ for LBM and $\tau = 0.064$ for collision implicit scheme. So, viscosity $\nu = 0.021333$ and the Reynolds number $Re = 150$ are the same in both cases.

Results

Convergence condition is satisfied after 51900 simulation time steps in both simulations. Reattachment length obtained by both schemes is the same $x_r = 4.0435$.

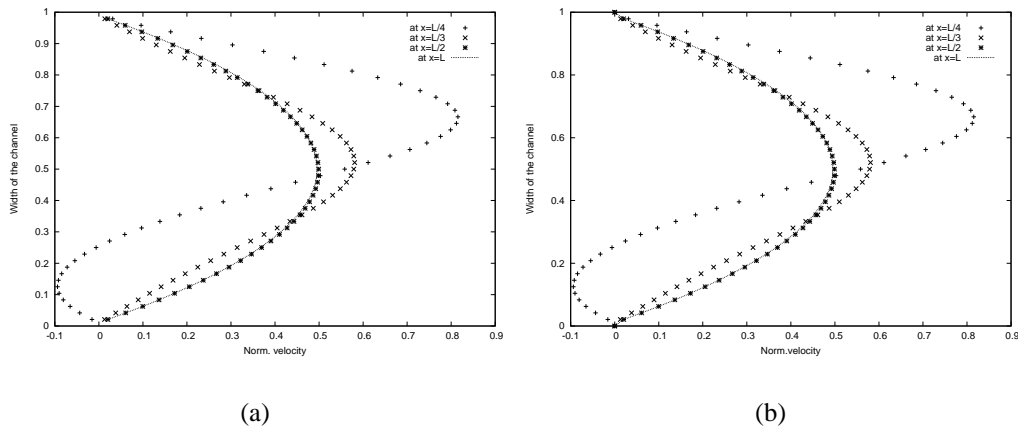


Figure 2.8: Normalised velocity profiles at four different places of the channel for $Re = 100$ by using the LBM (a) and using the collision implicit scheme (b).

It is interesting to make the comparison between obtained values of reattachment length with the experimental data obtained in [14]. This is provided in Figure 2.9. We see the small difference between numerical and experimental results. Probably, it is due to the different ratio between the step height h and the channel height H , $23/47$ in our case against $49/51$ used in [14].

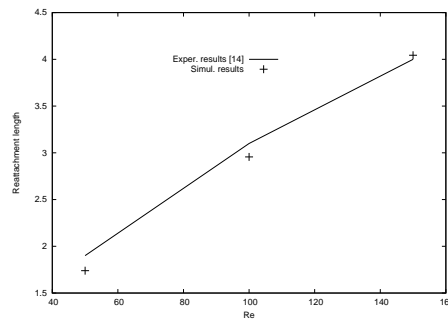


Figure 2.9: Comparison of simulation results and the experimental data of reattachment length for $Re=50, 100, 150$

Experiment 4

Relaxation time is $\tau = 0.548$ for LBM and $\tau = 0.048$ for collision implicit scheme. So, viscosity $\nu = 0.016$ and the Reynolds number $Re = 200$ are the same in both cases.

Results

Both schemes are unstable.

Conclusion

From the results of the simulations of the flow over a BFS we observe again the fact that the collision implicit scheme does not allow us to improve the stability range of LBM.

2.3.3 2D Pulsatile Channel Flow

It is well known that NS equations can be resolved analytically for time-dependent 2D Pulsatile channel flow (see [16]).

Artoli, Hoekstra and Sliot used LBM to obtain accurate numerical solutions of 2D Pulsatile channel flow (see [17]).

We conduct a number of experiments here by using both LBM and collision implicit scheme for this test problem. In all simulations we use the following conditions:

- The length of the tube is ten times larger than the height, with lattice size $N_x \times N_y = 200 \times 20$;
- Initial density is $\rho_0 = 1.0$;
- Initial velocity components are $u_0 = 0.0$ and $v_0 = 0.0$;
- PPB is used for inlet-outlet and modified on-grid bounce back no-slip boundary condition is used for walls;
- Density at the outlet is $\rho_{out} = 1.0$;
- The period of the driving pressure is $T = 1800$;
- Density at the inlet varies with time $\rho_{in} = 1.0 + A \cos(\frac{2\pi t}{T})$, where $A = 0.1$;

- $\theta_c = 0.5$ is chosen in (2.3);
- Convergence criterion is: difference between results from two successive periods in l_2 norm less than 10^{-7} .

Experiment 1

Relaxation time is $\tau = 0.6$ for LBM and $\tau = 0.1$ for collision implicit scheme, so viscosity $\nu = \frac{1}{30}$ in both cases. So, Womersley number $\alpha = \frac{H}{2} \sqrt{\frac{2\pi}{T\nu}} \approx 3.236$

Results

Velocity profiles at the middle of the tube at $t = 0, \frac{T}{8}, \frac{T}{4}, \frac{3T}{8}, \frac{T}{2}$ produced by both schemes are shown in Figure 2.10. Although the shapes are quite similar, there is a small time shift between them.

Time shift between analytical solutions and simulation results, obtained by LBM, was observed in [17]. Unfortunately, the reason of this time delay is not clear. At this point we do not know whether this shift is removed by collision implicit scheme or not.

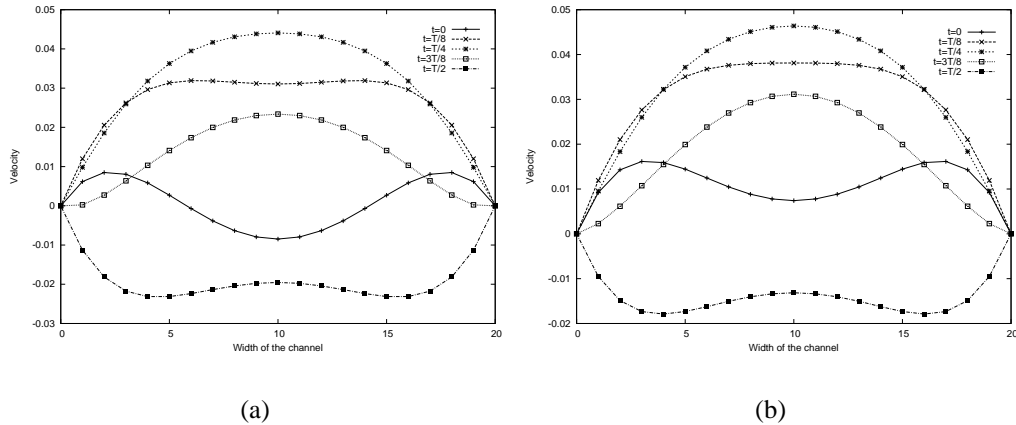


Figure 2.10: Velocity profiles at the middle of the channel at different times within the same period by using the LBM (a) and using the collision implicit scheme (b).

Experiment 2

Relaxation time is $\tau = 0.56$ for LBM and $\tau = 0.06$ for collision implicit scheme, so viscosity $\nu = 0.02$ in both cases. So, Womersley number $\alpha = \frac{H}{2} \sqrt{\frac{2\pi}{T\nu}} \approx 4.1777$

Results

Both schemes are unstable.

Conclusion

Even with unsteady flows we don't gain any improvements in terms of stability range when we use collision implicit scheme to compare with LBM.

2.4 Conclusions and discussion

Numerical experiments for steady 2D Poiseuille Channel flow, flow over BFS and for unsteady 2D Pulsatile Channel flow were conducted by using LBM and collision implicit scheme. Both schemes have second order convergence. Numerically it was verified that the collision implicit scheme does not enlarge the parameter space of LBM for both time-independent and time-dependent flows.

However, as we mentioned before, the collision implicit scheme (2.3) with $\theta_c = 0.5$ was successfully applied for the two-component system (see [9]). Surprisingly, it allowed to simulate flows with lattice viscosity as small as $\nu = 0.01$.

Chapter 3

Advection implicit scheme

3.1 Theory

The discretization in phase space we used in derivation of collision implicit LBE and common LBM was based on results from [2]. He and Luo accomplished it by discretizing momentum space in such a way that a lattice structure in configuration space was obtained directly. So, the discretization of configuration space is determined by the discretization of momentum space. However, these can be done independently (see [10]). Let us assume that continuous BGK equation (1.3) is discretized in momentum space. So, some set of discrete velocities is defined. Then we arrive to DBE (1.13). For the last one we can use any implicit schemes for the time and space discretization to obtain advection implicit LBE. We summarize this in Figure 3.1.

For simplicity we assume that we have a uniform Cartesian grid. So, $\Delta x = \Delta y$. We define a Courant-Friedrichs-Lewy (*CFL*) number as $CFL = \frac{\Delta t \max |\mathbf{e}_i|}{\Delta x}$.

By using the Crank-Nikolson scheme for time discretization in DBE, we obtain

$$\begin{aligned} & \frac{f_i(\mathbf{x}, t + \Delta t) - f_i(\mathbf{x}, t)}{\Delta t} + \frac{1}{2} (\mathbf{e}_i \cdot \nabla f_i(\mathbf{x}, t + \Delta t) + \mathbf{e}_i \cdot \nabla f_i(\mathbf{x}, t)) \\ &= -\frac{1}{2\lambda} \left[f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t) \right] - \frac{1}{2\lambda} \left[f_i(\mathbf{x}, t + \Delta t) - f_i^{(eq)}(\mathbf{x}, t + \Delta t) \right] \quad (3.1) \end{aligned}$$

For spatial derivatives we use the second order upwind scheme. Then we get

$$f_i(\mathbf{x}, t + \Delta t) + \frac{CFL}{4} [3f_i(\mathbf{x}, t + \Delta t) - 4f_i(\mathbf{x} - \mathbf{e}_i \Delta x, t + \Delta t) + f_i(\mathbf{x} - 2\mathbf{e}_i \Delta x, t + \Delta t)]$$

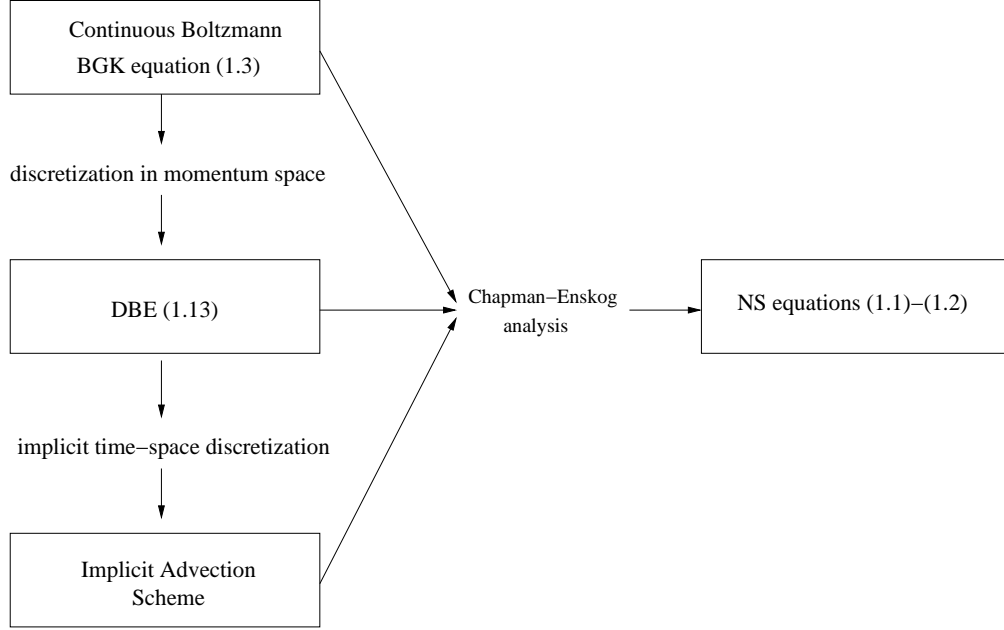


Figure 3.1: NS equations, Boltzmann BGK equation, DBE and Advection Implicit Scheme.

$$\begin{aligned}
 &= f_i(\mathbf{x}, t) - \frac{CFL}{4} [3f_i(\mathbf{x}, t) - 4f_i(\mathbf{x} - \mathbf{e}_i \Delta x, t) + f_i(\mathbf{x} - 2\mathbf{e}_i \Delta x, t)] \\
 &\quad - \frac{1}{2\tau} [f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t)] - \frac{1}{2\tau} [f_i(\mathbf{x}, t + \Delta t) - f_i^{(eq)}(\mathbf{x}, t + \Delta t)] \quad (3.2)
 \end{aligned}$$

We split the solution procedure into the following steps, as it was done in [12]:

- Pre-streaming collision step

$$\tilde{f}_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) - \frac{1}{2\tau} (f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t)) \quad (3.3)$$

- Streaming step

$$\begin{aligned}
 &\tilde{f}_i(\mathbf{x}, t + \Delta t) \\
 &+ \frac{CFL}{4} [3\tilde{f}_i(\mathbf{x}, t + \Delta t) - 4\tilde{f}_i(\mathbf{x} - \mathbf{e}_i \Delta x, t + \Delta t) + \tilde{f}_i(\mathbf{x} - 2\mathbf{e}_i \Delta x, t + \Delta t)] \\
 &= \tilde{f}_i(\mathbf{x}, t) - \frac{CFL}{4} [3\tilde{f}_i(\mathbf{x}, t) - 4\tilde{f}_i(\mathbf{x} - \mathbf{e}_i \Delta x, t) + \tilde{f}_i(\mathbf{x} - 2\mathbf{e}_i \Delta x, t)] \quad (3.4)
 \end{aligned}$$

- Post-streaming collision step

$$f_i(\mathbf{x}, t + \Delta t) = \tilde{f}_i(\mathbf{x}, t + \Delta t) - \frac{1}{2\tau} \left(f_i(\mathbf{x}, t + \Delta t) - f_i^{(eq)}(\mathbf{x}, t + \Delta t) \right) \quad (3.5)$$

The macroscopic density and velocity used in calculation of $f_i^{(eq)}$ at time $t + \Delta t$ in (3.4) are obtained by taking moments of \tilde{f}_i at time $t + \Delta t$

$$\rho = \sum_i \tilde{f}_i, \quad \rho \mathbf{u} = \sum_i \mathbf{e}_i \tilde{f}_i$$

Another possible way could be the following approximation

$$f_i^{(eq)}(\mathbf{x}, t + \Delta t) = 2f_i^{(eq)}(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t - \Delta t), \quad (3.6)$$

proposed by Mei and Shyy (see [18]). However, as it was pointed out by the authors, the extrapolation (3.5) may contribute to instability of the scheme.

Note that Eqs.(3.2)-(3.4) correspond to Eqs.(1.22)-(1.24) when $\theta_c = \frac{1}{2}$. Then by using (2.4), we find the kinematic viscosity for advection implicit scheme $\nu = \tau c_s^2 \Delta t$. Moreover, we conclude that scheme (3.1) has second order accuracy both in time and space.

In advection implicit schemes (3.1) we have the *CFL* number, which was not the case for collision implicit scheme. The *CFL* number allows us to control simulation time step Δt . It is clear that we can increase the simulation time step Δt by increasing the *CFL* number. So, the question is: can we take the *CFL* number in advection implicit scheme larger than the unity or not. This will be checked in our numerical experiments.

3.2 Implementation aspects

Pre-streaming collision step and post-streaming collision step, defined by (3.2) and (3.4), are local and can be implemented trivially. For distributions corresponding to the lattice velocity \mathbf{e}_0 (3.3) gives the trivial solution $\tilde{f}_0(\mathbf{x}, t + \Delta t) = \tilde{f}_0(\mathbf{x}, t)$. To find all other distributions in (3.3), we have to find the solution of linear sparse system $Ax = b$ of size $8N$, where N is the number of lattice nodes.

Due to the boundary conditions, A is not a symmetric matrix. This increases the computational cost significantly. Moreover, it is much more difficult to parallelize the work now to compare with common LBM.

For the matrix A we use LU decomposition. Then the system can be resolved by forward substitution and backward substitution (see [19]).

Advanced optimisation methods, such as bi-conjugate gradient stabilized solver (see [15]), could be used to improve the efficiency.

3.3 Numerical Experiments

We conduct the number of experiments by using the advection implicit scheme (3.1) for the 2D Poiseuille flow. In all simulations for tube flow we use the following conditions:

- The lattice size is $N_x \times N_y = 20 \times 10$;
- CFL number varies from 1 to 10;
- Relaxation time corresponding to the case $CFL = c$ is $\tau = \frac{1}{2c}$;
- Initial density is $\rho_0 = 1.0$;
- Initial velocity components are $u_0 = 0.01$ and $v_0 = 0.0$;
- VPB is used for inlet-outlet and modified on-grid bounce back no-slip boundary condition is used for walls;
- Parabolic velocity profile defined on the inlet has the centreline velocity $u_{max} = 0.1$, so the Mach number $M_a = \frac{u_{max}}{c_s} = 0.173205 \ll 1$;
- Density at the outlet is $\rho_{out} = 1.0$;
- In convergence condition (2.9) we use $\epsilon = 10^{-12}$.

Results

The advection implicit scheme (3.1) is stable only for $CFL \leq 5$. The relative error for $1 \leq CFL \leq 5$, measured by (2.17), is 0.0437 in all cases. Density difference $\Delta\rho$ between inlet and outlet of the tube for $1 \leq CFL \leq 5$ is measured. In all cases $\Delta\rho \approx 0.089$.

By using (2.16), we find the viscosity

$$\nu = \frac{H^2 \Delta\rho}{24L(\rho_{out} + \frac{\Delta\rho}{2})u_{max}} \approx 0.1775$$

It is the same for all $1 \leq CFL \leq 5$. This assures that the Reynolds number $Re \approx 5.6338$ is fixed in all simulations.

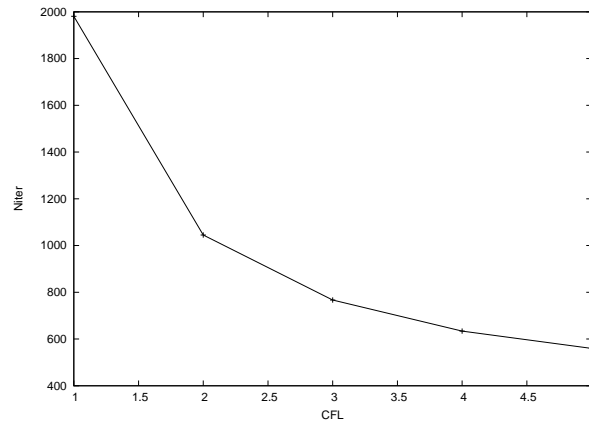


Figure 3.2: Number of iterations required for convergence for different CFL numbers.

The number of iterations required for convergence as a function of CFL number is shown in Figure 3.2. The number of iterations decreases almost by the factor of 2. However, the total simulation CPU time even for $CFL = 5$ is more than 20 times larger than the time for LBM. It is not surprising for us because LU decomposition is computationally expensive.

3.4 Conclusions and discussion

The advection implicit scheme (3.1) allowed us to increase the simulation time step Δt only 5 times. So, Crank-Nikolson scheme for time discretization is conditionally stable for DBE. Implicit second order Taylor-Galerkin scheme is more stable in this case (see [12]). We could also go to the third order unconditionally stable implicit schemes.

Solving the sparse linear system by using LU decomposition is computationally expensive. More advanced techniques should be used to decrease the total simulation time.

Chapter 4

Summary and Future Plans

4.1 Summary

The collision implicit schemes is derived directly from continuous Boltzmann BGK equation and the advection implicit scheme is derived directly from Discrete Boltzmann equation. Constructed implicit schemes recover Navier-Stokes equations in the limit of low Mach number and have second order accuracy both in time and space. It is numerically verified that while the collision implicit scheme has no advantage in terms of stability range over conventional LBM for both steady and unsteady flows, the advection implicit scheme allows us to increase the simulation time step used in LBM.

4.2 Future Plans

- Apply unconditionally stable second or higher order implicit finite difference schemes to derive the advection implicit schemes from DBE;
- Apply iterative solvers, such as bi-conjugate gradient stabilized solver, to resolve the sparse linear system;
- Parallelize the implementation of advection implicit schemes;
- Use nonuniform grids;
- Simulate 2D and 3D time dependent flows by using advection implicit schemes.

Chapter 5

Appendixes

5.1 Appendix A

In this Appendix we study Chapman-Enskog expansion for the collision implicit LBGK scheme:

$$\begin{aligned} f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) &= -\frac{1 - \theta_c}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t) \right) \\ &\quad - \frac{\theta_c}{\tau} \left(f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i^{(eq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) \right) \end{aligned} \quad (\text{A-1})$$

At first we rewrite it in the following way:

$$\begin{aligned} f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) &= -\frac{1}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t) \right) \\ &\quad + \frac{\theta_c}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t) - f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) + f_i^{(eq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) \right), \end{aligned} \quad (\text{A-2})$$

where time increment Δt can be regarded as small parameter of order ε .

To derive NS equations (1.1)-(1.2) we apply Chapman-Enskog multi-scale expansion:

$$\frac{\partial}{\partial t} = \varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2}, \quad \frac{\partial}{\partial x} = \varepsilon \frac{\partial}{\partial x_1} \quad (\text{A-3})$$

Single-particle distribution f_i can be expanded formally about the local equilibrium distribution function $f_i^{(eq)}$ as:

$$f_i = f_i^{(eq)} + \varepsilon f_i^{(neq)}, \quad (\text{A-4})$$

where single-particle distribution f_i and local equilibrium distribution $f_i^{(eq)}$ satisfy the following equalities:

$$\rho = \sum_i f_i = \sum_i f_i^{(eq)}, \quad \rho \mathbf{u} = \sum_i \mathbf{e}_i f_i = \sum_i \mathbf{e}_i f_i^{(eq)} \quad (\text{A-5})$$

and nonequilibrium distribution can be written as:

$$f_i^{(neq)} = f_i^{(1)} + \varepsilon f_i^{(2)} + O(\varepsilon^2) \quad (\text{A-6})$$

By combining (A-4), (A-5) and (A-6), we obtain the following constraints:

$$\sum_i f_i^{(1)} = 0, \quad \sum_i \mathbf{e}_i f_i^{(1)} = 0, \quad \sum_i f_i^{(2)} = 0, \quad \sum_i \mathbf{e}_i f_i^{(2)} = 0 \quad (\text{A-7})$$

By combining (A-4) and (A-6) and neglecting the term $O(\varepsilon^3)$, we get

$$f_i = f_i^{(eq)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} \quad (\text{A-8})$$

By using (A-4), we rewrite (A-2) as:

$$\begin{aligned} f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) &= -\frac{\varepsilon}{\tau} f_i^{(neq)}(\mathbf{x}, t) \\ &+ \frac{\varepsilon \theta_c}{\tau} \left[f_i^{(neq)}(\mathbf{x}, t) - f_i^{(neq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) \right] \end{aligned} \quad (\text{A-9})$$

Now we use Taylor expansion

$$\begin{aligned} f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) &= f_i(\mathbf{x}, t) + \Delta t \frac{\partial f_i(\mathbf{x}, t)}{\partial t} + \Delta t \mathbf{e}_i \cdot \nabla f_i(\mathbf{x}, t) + \frac{\Delta t^2}{2} \frac{\partial^2 f_i(\mathbf{x}, t)}{\partial t^2} \\ &+ \frac{\Delta t^2}{2} \mathbf{e}_i \mathbf{e}_i \nabla \nabla f_i(\mathbf{x}, t) + \Delta t^2 \mathbf{e}_i \cdot \nabla \frac{\partial f_i(\mathbf{x}, t)}{\partial t} + O(\Delta t^3) \end{aligned} \quad (\text{A-10})$$

$$\begin{aligned} f_i^{(neq)}(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) &= f_i^{(neq)}(\mathbf{x}, t) + \Delta t \frac{\partial f_i^{(neq)}(\mathbf{x}, t)}{\partial t} + \Delta t \mathbf{e}_i \cdot \nabla f_i^{(neq)}(\mathbf{x}, t) \\ &+ \frac{\Delta t^2}{2} \frac{\partial^2 f_i^{(neq)}(\mathbf{x}, t)}{\partial t^2} + \frac{\Delta t^2}{2} \mathbf{e}_i \mathbf{e}_i \nabla \nabla f_i^{(neq)}(\mathbf{x}, t) + \Delta t^2 \mathbf{e}_i \cdot \nabla \frac{\partial f_i^{(neq)}(\mathbf{x}, t)}{\partial t} + O(\Delta t^3) \end{aligned} \quad (\text{A-11})$$

By putting (A-10) and (A-11) into (A-9), neglecting the $O(\Delta t^3)$ term and dividing both sides by Δt , we get

$$\begin{aligned} & \frac{\partial f_i(\mathbf{x}, t)}{\partial t} + \mathbf{e}_i \cdot \nabla f_i(\mathbf{x}, t) + \frac{\varepsilon}{2} \frac{\partial^2 f_i(\mathbf{x}, t)}{\partial t^2} + \frac{\varepsilon}{2} \mathbf{e}_i \mathbf{e}_i \nabla \nabla f_i(\mathbf{x}, t) + \varepsilon \mathbf{e}_i \cdot \nabla \frac{\partial f_i(\mathbf{x}, t)}{\partial t} \\ &= -\frac{1}{\tau} f_i^{(neq)}(\mathbf{x}, t) - \frac{\theta_c}{\tau} \left[\varepsilon \frac{\partial f_i^{(neq)}(\mathbf{x}, t)}{\partial t} + \varepsilon \mathbf{e}_i \cdot \nabla f_i^{(neq)}(\mathbf{x}, t) + \frac{\varepsilon^2}{2} \frac{\partial^2 f_i^{(neq)}(\mathbf{x}, t)}{\partial t^2} \right. \\ & \quad \left. + \frac{\varepsilon^2}{2} \mathbf{e}_i \mathbf{e}_i \nabla \nabla f_i^{(neq)}(\mathbf{x}, t) + \varepsilon^2 \mathbf{e}_i \cdot \nabla \frac{\partial f_i^{(neq)}(\mathbf{x}, t)}{\partial t} \right] \end{aligned} \quad (\text{A-12})$$

Now we apply multi-scale expansions (A-3) to terms in (A-12).

$$\begin{aligned} \frac{\partial f_i}{\partial t} &= \left(\varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} \right) \left(f_i^{(eq)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} \right) \\ &= \varepsilon \frac{\partial f_i^{(eq)}}{\partial t_1} + \varepsilon^2 \left(\frac{\partial f_i^{(1)}}{\partial t_1} + \frac{\partial f_i^{(eq)}}{\partial t_2} \right) + O(\varepsilon^3); \end{aligned} \quad (\text{A-13})$$

$$\begin{aligned} \mathbf{e}_i \cdot \nabla f_i &= \mathbf{e}_{i\alpha} \frac{\partial f_i}{\partial x_\alpha} = \varepsilon \mathbf{e}_{i\alpha} \frac{\partial}{\partial x_{1\alpha}} \left(f_i^{(eq)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} \right) \\ &= \varepsilon \mathbf{e}_{i\alpha} \frac{\partial f_i^{(eq)}}{\partial x_{1\alpha}} + \varepsilon^2 \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} + O(\varepsilon^3); \end{aligned} \quad (\text{A-14})$$

$$\frac{\partial^2 f_i}{\partial t^2} = \left(\varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} \right)^2 \left(f_i^{(eq)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} \right) = \varepsilon^2 \frac{\partial^2 f_i^{(eq)}}{\partial t_1^2} + O(\varepsilon^3); \quad (\text{A-15})$$

$$\begin{aligned} \mathbf{e}_i \mathbf{e}_i \nabla \nabla f_i &= \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial^2 f_i}{\partial x_\alpha \partial x_\beta} = \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial}{\partial x_\alpha} \frac{\partial}{\partial x_\beta} \left(f_i^{(eq)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} \right) \\ &= \varepsilon^2 \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial}{\partial x_{1\alpha}} \frac{\partial}{\partial x_{1\beta}} \left(f_i^{(eq)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} \right) = \varepsilon^2 \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial^2 f_i^{(eq)}}{\partial x_{1\alpha} \partial x_{1\beta}} + O(\varepsilon^3); \end{aligned} \quad (\text{A-16})$$

$$\begin{aligned} \mathbf{e}_i \cdot \nabla \frac{\partial f_i}{\partial t} &= \mathbf{e}_{i\alpha} \frac{\partial^2 f_i}{\partial t \partial x_\alpha} = \mathbf{e}_{i\alpha} \left(\varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} \right) \varepsilon \frac{\partial}{\partial x_{1\alpha}} \left(f_i^{(eq)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} \right) \\ &= \varepsilon^2 \mathbf{e}_{i\alpha} \frac{\partial^2 f_i^{(eq)}}{\partial t_1 \partial x_{1\alpha}} + O(\varepsilon^3); \end{aligned} \quad (\text{A-17})$$

$$\begin{aligned} \frac{\partial f_i^{(neq)}}{\partial t} &= \left(\varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} \right) (f_i^{(1)} + \varepsilon f_i^{(2)}) \\ &= \varepsilon \frac{\partial f_i^{(1)}}{\partial t_1} + \varepsilon^2 \left(\frac{\partial f_i^{(2)}}{\partial t_1} + \frac{\partial f_i^{(1)}}{\partial t_2} \right) + O(\varepsilon^3); \quad (\text{A-18}) \end{aligned}$$

$$\begin{aligned} \mathbf{e}_i \cdot \nabla f_i^{(neq)} &= \mathbf{e}_{i\alpha} \frac{\partial f_i^{(neq)}}{\partial x_\alpha} = \varepsilon \mathbf{e}_{i\alpha} \frac{\partial}{\partial x_{1\alpha}} (f_i^{(1)} + \varepsilon f_i^{(2)}) \\ &= \varepsilon \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} + \varepsilon^2 \mathbf{e}_{i\alpha} \frac{\partial f_i^{(2)}}{\partial x_{1\alpha}} + O(\varepsilon^3); \quad (\text{A-19}) \end{aligned}$$

$$\frac{\partial^2 f_i^{(neq)}}{\partial t^2} = \left(\varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} \right)^2 (f_i^{(1)} + \varepsilon f_i^{(2)}) = \varepsilon^2 \frac{\partial^2 f_i^{(1)}}{\partial t_1^2} + O(\varepsilon^3); \quad (\text{A-20})$$

$$\begin{aligned} \mathbf{e}_i \mathbf{e}_i \nabla \nabla f_i^{(neq)} &= \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial^2 f_i^{(neq)}}{\partial x_\alpha \partial x_\beta} = \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial}{\partial x_\alpha} \frac{\partial}{\partial x_\beta} (f_i^{(1)} + \varepsilon f_i^{(2)}) \\ &= \varepsilon^2 \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial}{\partial x_{1\alpha}} \frac{\partial}{\partial x_{1\beta}} (f_i^{(1)} + \varepsilon f_i^{(2)}) = \varepsilon^2 \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial^2 f_i^{(1)}}{\partial x_{1\alpha} \partial x_{1\beta}} + O(\varepsilon^3); \quad (\text{A-21}) \end{aligned}$$

$$\begin{aligned} \mathbf{e}_i \cdot \nabla \frac{\partial f_i^{(neq)}}{\partial t} &= \mathbf{e}_{i\alpha} \frac{\partial^2 f_i^{(neq)}}{\partial t \partial x_\alpha} = \mathbf{e}_{i\alpha} \left(\varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} \right) \varepsilon \frac{\partial}{\partial x_{1\alpha}} (f_i^{(1)} + \varepsilon f_i^{(2)}) \\ &= \varepsilon^2 \mathbf{e}_{i\alpha} \frac{\partial^2 f_i^{(1)}}{\partial t_1 \partial x_{1\alpha}} + O(\varepsilon^3) \quad (\text{A-22}) \end{aligned}$$

By neglecting the term $O(\varepsilon^3)$ in all (A-13)-(A-22), putting them in (A-12) and dividing both sides by ε , we obtain

$$\begin{aligned} &\frac{\partial f_i^{(eq)}}{\partial t_1} + \varepsilon \left(\frac{\partial f_i^{(1)}}{\partial t_1} + \frac{\partial f_i^{(eq)}}{\partial t_2} \right) + \mathbf{e}_{i\alpha} \frac{\partial f_i^{(eq)}}{\partial x_{1\alpha}} + \varepsilon \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} + \frac{\varepsilon^2}{2} \frac{\partial^2 f_i^{(eq)}}{\partial t_1^2} + \varepsilon^2 \mathbf{e}_{i\alpha} \frac{\partial^2 f_i^{(eq)}}{\partial t_1 \partial x_{1\alpha}} \\ &+ \frac{\varepsilon^2}{2} \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial^2 f_i^{(eq)}}{\partial x_{1\alpha} \partial x_{1\beta}} = -\frac{1}{\tau \varepsilon} [f_i^{(1)} + \varepsilon f_i^{(2)}] - \frac{\theta_c}{\tau} \left[\varepsilon \frac{\partial f_i^{(1)}}{\partial t_1} + \varepsilon^2 \frac{\partial f_i^{(2)}}{\partial t_1} + \varepsilon^2 \frac{\partial f_i^{(1)}}{\partial t_2} \right. \\ &\left. + \varepsilon \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} + \varepsilon^2 \mathbf{e}_{i\alpha} \frac{\partial f_i^{(2)}}{\partial x_{1\alpha}} + \frac{\varepsilon^3}{2} \frac{\partial^2 f_i^{(1)}}{\partial t_1^2} + \frac{\varepsilon^3}{2} \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial^2 f_i^{(1)}}{\partial x_{1\alpha} \partial x_{1\beta}} + \varepsilon^3 \mathbf{e}_{i\alpha} \frac{\partial^2 f_i^{(1)}}{\partial t_1 \partial x_{1\alpha}} \right] \end{aligned} \quad (\text{A-23})$$

From (A-23) we obtain the equation to the order ε^0

$$\frac{\partial f_i^{(eq)}}{\partial t_1} + \mathbf{e}_{i\alpha} \frac{\partial f_i^{(eq)}}{\partial x_{1\alpha}} = -\frac{1}{\tau\varepsilon} f_i^{(1)} \quad (\text{A-24})$$

and equation to the order ε^1

$$\begin{aligned} \frac{\partial f_i^{(1)}}{\partial t_1} + \frac{\partial f_i^{(eq)}}{\partial t_2} + \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} + \frac{\varepsilon}{2} \left(\frac{\partial^2 f_i^{(eq)}}{\partial t_1^2} + \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial^2 f_i^{(eq)}}{\partial x_{1\alpha} \partial x_{1\beta}} + 2\mathbf{e}_{i\alpha} \frac{\partial^2 f_i^{(eq)}}{\partial t_1 \partial x_{1\alpha}} \right) \\ = -\frac{1}{\tau\varepsilon} f_i^{(2)} - \frac{\theta_c}{\tau} \left(\frac{\partial f_i^{(1)}}{\partial t_1} + \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} \right) \end{aligned} \quad (\text{A-25})$$

We can rewrite (A-25) as:

$$\begin{aligned} \frac{\partial f_i^{(1)}}{\partial t_1} + \frac{\partial f_i^{(eq)}}{\partial t_2} + \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} + \frac{\varepsilon}{2} \left[\frac{\partial}{\partial t_1} \left(\frac{\partial f_i^{(eq)}}{\partial t_1} + \mathbf{e}_{i\alpha} \frac{\partial f_i^{(eq)}}{\partial x_{1\alpha}} \right) \right. \\ \left. + \mathbf{e}_{i\alpha} \frac{\partial}{\partial x_{1\alpha}} \left(\frac{\partial f_i^{(eq)}}{\partial t_1} + \mathbf{e}_{i\beta} \frac{\partial f_i^{(eq)}}{\partial x_{1\beta}} \right) \right] = -\frac{1}{\tau\varepsilon} f_i^{(2)} - \frac{\theta_c}{\tau} \left(\frac{\partial f_i^{(1)}}{\partial t_1} + \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} \right) \end{aligned} \quad (\text{A-26})$$

Now by using (A-24), from last equation we obtain

$$\begin{aligned} \frac{\partial f_i^{(1)}}{\partial t_1} + \frac{\partial f_i^{(eq)}}{\partial t_2} + \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} - \frac{1}{2\tau} \left(\frac{\partial f_i^{(1)}}{\partial t_1} + \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} \right) \\ = -\frac{1}{\tau\varepsilon} f_i^{(2)} - \frac{\theta_c}{\tau} \left(\frac{\partial f_i^{(1)}}{\partial t_1} + \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} \right) \end{aligned} \quad (\text{A-27})$$

or

$$\frac{\partial f_i^{(1)}}{\partial t_1} + \frac{\partial f_i^{(eq)}}{\partial t_2} + \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} + \frac{2\theta_c - 1}{2\tau} \left(\frac{\partial f_i^{(1)}}{\partial t_1} + \mathbf{e}_{i\alpha} \frac{\partial f_i^{(1)}}{\partial x_{1\alpha}} \right) = -\frac{1}{\tau\varepsilon} f_i^{(2)} \quad (\text{A-28})$$

By taking zeroth order velocity moment of (A-24) and using (A-5) and (A-7), we obtain

$$\frac{\partial \rho}{\partial t_1} + \frac{\partial \rho \mathbf{u}_\alpha}{\partial x_{1\alpha}} = 0 \quad (\text{A-29})$$

By taking zeroth order velocity moment of (A-28) and using (A-5) and (A-7), we obtain

$$\frac{\partial \rho}{\partial t_2} = 0 \quad (\text{A-30})$$

We multiply (A-29) by ε , (A-30) by ε^2 and add them

$$\varepsilon \frac{\partial \rho}{\partial t_1} + \varepsilon \frac{\partial \rho \mathbf{u}_\alpha}{\partial x_{1\alpha}} + \varepsilon^2 \frac{\partial \rho}{\partial t_2} = 0 \quad (\text{A-31})$$

By using (A-3) in (A-31), we obtain the mass conservation equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (\text{A-32})$$

For incompressible fluid flow density ρ is constant. Then (A-32) reduces to (1.1). By taking first order velocity moment of (A-24) and applying (A-5) and (A-7), we obtain

$$\frac{\partial \rho \mathbf{u}_\alpha}{\partial t_1} + \sum_i \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial f_i^{(eq)}}{\partial x_{1\beta}} = 0 \quad (\text{A-33})$$

By taking first order velocity moment of (A-28) and applying (A-5) and (A-7), we obtain

$$\frac{\partial \rho \mathbf{u}_\alpha}{\partial t_2} + \frac{2\theta_c + 2\tau - 1}{2\tau} \sum_i \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial f_i^{(1)}}{\partial x_{1\beta}} = 0 \quad (\text{A-34})$$

We multiply (A-33) by ε , (A-34) by ε^2 and add them

$$\varepsilon \frac{\partial \rho \mathbf{u}_\alpha}{\partial t_1} + \varepsilon \sum_i \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial f_i^{(eq)}}{\partial x_{1\beta}} + \varepsilon^2 \frac{\partial \rho \mathbf{u}_\alpha}{\partial t_2} + \varepsilon^2 \frac{2\theta_c + 2\tau - 1}{2\tau} \sum_i \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \frac{\partial f_i^{(1)}}{\partial x_{1\beta}} = 0 \quad (\text{A-35})$$

By using (A-3) in (A-35), we obtain

$$\frac{\partial \rho \mathbf{u}_\alpha}{\partial t} + \frac{\partial}{\partial x_\beta} \sum_i \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} \left(f_i^{(eq)} + \varepsilon \frac{2\theta_c + 2\tau - 1}{2\tau} f_i^{(1)} \right) = 0 \quad (\text{A-36})$$

or in the vector notation

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\Pi^{(0)} + \Pi^{(1)}) = 0, \quad (\text{A-37})$$

where the momentum flux tensors are defined by

$$\Pi^{(0)} = \sum_i \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} f_i^{(eq)} = P \delta_{\alpha\beta} + \rho \mathbf{u}_\alpha \mathbf{u}_\beta \quad (\text{A-38})$$

$$\Pi^{(1)} = \varepsilon \frac{2\theta_c + 2\tau - 1}{2\tau} \sum_i \mathbf{e}_{i\alpha} \mathbf{e}_{i\beta} f_i^{(1)} = v \left(\frac{\partial \rho \mathbf{u}_\beta}{\partial x_\alpha} + \frac{\partial \rho \mathbf{u}_\alpha}{\partial x_\beta} \right) \quad (\text{A-39})$$

with kinematic viscosity defined as:

$$v = (\tau + \theta_c - 0.5) c_s^2 \Delta t \quad (\text{A-40})$$

5.2 Appendix B

In this Appendix we describe the Newton-Raphson iterative procedure to solve the system of nonlinear equations (2.10) as it is given in [19]. First, we rewrite (2.10) in the following form:

$$F_i(y_0, y_1, \dots, y_8) \equiv y_i - a * g_i(y_0, y_1, \dots, y_8) - c_i = 0, \quad i = 0, 1, \dots, 8 \quad (\text{B-1})$$

where

$$y_i = f_i(\mathbf{x}, t + \Delta t), \quad a = \frac{\theta_c}{\tau + \theta_c}, \quad g_i(y_0, y_1, \dots, y_8) = f_i^{(eq)}(\mathbf{x}, t + \Delta t),$$

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

Initially we put $y_i^0 = f_i(\mathbf{x}, t)$. Newton-Raphson iterative method implies finding the corrections Δy_i by solving the system of linear equations

$$\sum_{j=0}^8 \alpha_{ij} \Delta y_j = \beta_i, \quad i = 0, 1, \dots, 8 \quad (\text{B-2})$$

where coefficients

$$\alpha_{ij} = \frac{\partial F_i}{\partial y_j}, \quad \beta_i = -F_i$$

are evaluated for the old values y_i^n .

System (B-2) can be resolved by LU decomposition. The corrections are then added to the solution vector

$$y_i^{n+1} = y_i^n + \Delta y_i, \quad i = 0, 1, \dots, 8$$

Iterations continue until all Δy_i have values smaller than some small number. In our simulations this number is equal to 10^{-6} .

Finally, we describe the way to find the partial derivatives $\frac{\partial F_i}{\partial y_j}$. By using (1.12)-(1.13), we write the equations of system (B-1) explicitly

$$\begin{aligned}
F_0 &\equiv y_0 - a \frac{4\rho}{9} \left[1 - \frac{3}{2}(u^2 + v^2) \right] - c_0 = 0, \\
F_1 &\equiv y_1 - a \frac{\rho}{9} \left[1 + 3u + 3u^2 - \frac{3}{2}v^2 \right] - c_1 = 0, \\
F_2 &\equiv y_2 - a \frac{\rho}{9} \left[1 + 3v + 3v^2 - \frac{3}{2}u^2 \right] - c_2 = 0, \\
F_3 &\equiv y_3 - a \frac{\rho}{9} \left[1 - 3u + 3u^2 - \frac{3}{2}v^2 \right] - c_3 = 0, \\
F_4 &\equiv y_4 - a \frac{\rho}{9} \left[1 - 3v + 3v^2 - \frac{3}{2}u^2 \right] - c_4 = 0, \\
F_5 &\equiv y_5 - a \frac{\rho}{36} [1 + 3u + 3v + 3u^2 + 3v^2 + 9uv] - c_5 = 0, \\
F_6 &\equiv y_6 - a \frac{\rho}{36} [1 - 3u + 3v + 3u^2 + 3v^2 - 9uv] - c_6 = 0, \\
F_7 &\equiv y_7 - a \frac{\rho}{36} [1 - 3u - 3v + 3u^2 + 3v^2 + 9uv] - c_7 = 0, \\
F_8 &\equiv y_8 - a \frac{\rho}{36} [1 + 3u - 3v + 3u^2 + 3v^2 - 9uv] - c_8 = 0.
\end{aligned}$$

where

$$\rho = \sum_{i=0}^8 y_i, \quad u = \frac{y_1 + y_5 + y_8 - y_3 - y_6 - y_7}{\rho}, \quad v = \frac{y_2 + y_5 + y_6 - y_4 - y_7 - y_8}{\rho}$$

Now, by doing some trivial calculations, we find all partial derivatives $\frac{\partial F_i}{\partial y_j}$. We give here, as example, only the case for $i = 0$.

$$\begin{aligned}
\frac{\partial F_0}{\partial y_0} &= 1 - \frac{4a}{9} \left[1 + \frac{3}{2}(u^2 + v^2) \right], \\
\frac{\partial F_0}{\partial y_1} &= -\frac{4a}{9} \left[1 + \frac{3}{2}(u^2 + v^2) - 3u \right], \\
\frac{\partial F_0}{\partial y_2} &= -\frac{4a}{9} \left[1 + \frac{3}{2}(u^2 + v^2) - 3v \right],
\end{aligned}$$

$$\frac{\partial F_0}{\partial y_3} = -\frac{4a}{9} \left[1 + \frac{3}{2}(u^2 + v^2) + 3u \right],$$

$$\frac{\partial F_0}{\partial y_4} = -\frac{4a}{9} \left[1 + \frac{3}{2}(u^2 + v^2) + 3v \right],$$

$$\frac{\partial F_0}{\partial y_5} = -\frac{4a}{9} \left[1 + \frac{3}{2}(u^2 + v^2) - 3(u + v) \right],$$

$$\frac{\partial F_0}{\partial y_6} = -\frac{4a}{9} \left[1 + \frac{3}{2}(u^2 + v^2) + 3(u - v) \right],$$

$$\frac{\partial F_0}{\partial y_7} = -\frac{4a}{9} \left[1 + \frac{3}{2}(u^2 + v^2) + 3(u + v) \right],$$

$$\frac{\partial F_0}{\partial y_8} = -\frac{4a}{9} \left[1 + \frac{3}{2}(u^2 + v^2) - 3(u - v) \right].$$

List of Figures

1.1	Nine velocities for the D2Q9 model.	3
1.2	NS equations, LBM and Boltzmann BGK equation.	4
2.1	NS equations, Boltzmann BGK equation and Collision Implicit Scheme.	12
2.2	Mid-grid bounce back, no-slip boundary condition.	15
2.3	On-grid bounce back, no-slip boundary condition.	16
2.4	2D Channel Flow.	19
2.5	Relative errors for $Re = 6$ and $Re = 30$ using the LBM (a) and using the collision implicit scheme (b).	21
2.6	Normalised velocity profiles at the outlet of the channel for $Re = 30$ and lattice size $N_x \times N_y = 80 \times 40$ using the LBM (a) and using the collision implicit scheme (b).	22
2.7	Backward-Facing Step.	23
2.8	Normalised velocity profiles at four different places of the channel for $Re = 100$ by using the LBM (a) and using the collision implicit scheme (b).	25
2.9	Comparison of simulation results and the experimental data of reattachment length for $Re=50,100,150$	25
2.10	Velocity profiles at the middle of the channel at different times within the same period by using the LBM (a) and using the collision implicit scheme (b).	27
3.1	NS equations, Boltzmann BGK equation, DBE and Advection Implicit Scheme.	30
3.2	Number of iterations required for convergence for different CFL numbers.	33

Bibliography

- [1] P.L.BHATNAGAR, E.P.GROSS AND M.KROOK, *A model for collisional processes in gases I: small amplitude processes in charged and neutral one-component system*, Phys. Review, vol.94, 511 (1954).
- [2] X.HE AND L.S.LUO, *A priori derivation of the lattice Boltzmann equation*, Phys. Review E, vol.55, No.6, pp.6333-6336 (1997).
- [3] S.CHEN AND G.DOOLEN, *Lattice Boltzmann method for fluid flows*, Ann. Rev. Fluid Mech. 30, 329 (1998).
- [4] J.D.STERLING AND S.CHEN, *Stability Analysis of the Lattice Boltzmann Methods*, J. Comp. Phys. 123, 196 (1996).
- [5] J.TOELKE,M.KRAFCZYK,M.SCHULZ, E.RANK AND R.BERRIOS, *Implicit discretization and nonuniform mesh refinement approaches for FD discretizations of LBGK models*, Int. J. Mod. Phys. C, vol.9, pp.1143-1157 (1998).
- [6] NR.VERBERG AND A.LADD, *Simulation of low Reynolds-number flow via a time-independent lattice-Boltzmann method*, Phys. Rev. E, vol.60, pp.3366-3373 (1999).
- [7] L.MIEUSSENS, *Discrete velocity model and implicit scheme for the BGK equation of Rarefied Gas Dynamics*, Math. Models and Meth. Appl. Sc., Rev. E, vol.10, No. 8, pp.1121-1149 (2000).
- [8] M.BERNASCHI,S.SUCCI, AND H.CHEN, *Accelerated Lattice Boltzmann Schemes for Steady-State Flow Simulations*, J. Scientific Comp., vol.16, No.2, pp.135-144 (2001).
- [9] B.SANKARANARAYANAN, X.SHAN, I.G.KEVREKIDIS AND S.SUNDARESAN, *Analysis of drag and virtual mass forces in bubbly suspensions using an implicit formulation of the lattice Boltzmann method*, J. Fluid Mech., vol.452, pp.61-96 (2002).

- [10] N.CAO, S.CHEN, S.JIN AND D.MARTINEZ, *Physical symmetry and lattice symmetry in the lattice Boltzmann method*, Phys. Review E, vol.55, N1, pp.21-24 (1997).
- [11] T.SETA AND R.TAKAHASHI, *Numerical stability Analysis of FDLBM*, J. Stat. Phys., vol.107, N1/2, pp.557-572 (2002).
- [12] T.LEE AND C.L.LIN, *An Eulerian description of the streaming process in the lattice Boltzmann equation*, J. Comp. Phys., vol.185, pp.445-471 (2003).
- [13] Q.ZOU AND X.HE, *On pressure and velocity boundary conditions for the lattice Boltzmann BGK model*, Phys. Fluids, vol.9 (6), pp.1591-1598 (1997).
- [14] B.F.ARMALY, F.DURST, J.C.F.PEREIRA AND B.SCHONUNG, *Experimental and theoretical investigation of backward-facing step flow*, J. Fluid Mech., vol.127, pp.473-496 (1983).
- [15] H.A. VAN DER VORST, *Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., vol.13(2), p.631 (1992).
- [16] L.D.LANDAU, E.M.LIFSHITZ, *Fluid Mechanics*, Pergamon Press Ltd, (1959).
- [17] A.M.M.ARTOLI, A.G.HOEKSTRA AND P.M.A.SLOOT, *Accuracy of 2D pulsatile flow in the lattice Boltzmann BGK method*, Computational Science - ICCS 2002, Proceedings Part I, in series Lecture Notes in Computer Science, vol.2329, pp.361-370 (2002).
- [18] R.MEI AND W.SHYY, *On the Finite Difference-Based Lattice Boltzmann Method in Curvilinear Coordinates*, J. Comp. Phys., vol.143, pp.426-448 (1998).
- [19] W.H.PRESS, B.P.FLANNERY, S.A.TEUKOLSKY, W.T.VETTERLING, *Numerical Recipes in C*, Cambridge, (1988).