FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT • DEPARTMENT INFORMATIK

Lehrstuhl für Informatik 10 (Systemsimulation)

Entropic stabilization of the lattice Boltzmann method

Ghulam Mustafa Majal

Master Thesis

Entropic stabilization of the lattice Boltzmann method

Ghulam Mustafa Majal

Master Thesis

Aufgabensteller:	Prof. Dr. U. Rüde
Betreuer:	Dipl.-Inf.,Dipl-.Math. S. Bogner, Dipl.-Math. R. Ammer
Bearbeitungszeitraum:	02.05.2014 – 02.11.2014

Erklärung:

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Der Universität Erlangen-Nürnberg, vertreten durch den Lehrstuhl für Systemsimulation (Informatik 10), wird für Zwecke der Forschung und Lehre ein einfaches, kostenloses, zeitlich und örtlich unbeschränktes Nutzungsrecht an den Arbeitsergebnissen der Master Thesis einschließlich etwaiger Schutzrechte und Urheberrechte eingeräumt.

Erlangen, den 3. November 2014 ....

Abstract

In this thesis, the entropic lattice Boltzmann method is presented. The standard lattice Boltzmann method using a single time relaxation parameter and the Bhattnager-Gross-Krook (BGK) collision operator becomes unstable for low viscosity flows. On the other hand, the entropic lattice Boltzmann method is said to provide enhanced stability by enforcing a discrete H-Theorem. The only difference in the algorithm of the entropic lattice Boltzmann method from the standard lattice Boltzmann schemes based on the single relaxation time parameter comes in the collision step. Special relaxation parameters are used which ensure that the H-Theorem is never violated. The equilibrium distribution function also takes a different form. In order to test the accuracy and stability of the entropic lattice Boltzmann method, three different simulations were carried out. The first set of experiments were based on the Taylor-Green vortex and the results obtained showed that the accuracy of the entropic lattice Boltzmann method is quite similar to the standard lattice Boltzmann method. The next two experiments based on the 1d shock tube and the lid driven cavity flow were intended to test the stability of the entropic lattice Boltzmann method. For both sets of experiments even though the entropic lattice Boltzmann method did not provide entirely accurate results it was clearly more stable than the standard lattice Boltzmann method.

Contents

Chapter 1

Introduction

In conventional computational fluid dynamics schemes (CFD) the first task at hand is to discretize the Navier-Stokes equations. The Navier-Stokes equations (see Eq.(2.4) and (2.5)) can be discretized using finite element or finite difference or finite volume schemes. Finally, these approximations are solved on a given mesh with specified boundary conditions. A more detailed account on the techniques utilized in CFD can be found in Ref.[1].

On the other hand the lattice Boltzmann method is based on microscopic models and mesoscopic kinetic equations[2]. The crucial concept behind the lattice Boltzmann method is to make use of simplified kinetic models that include the necessary physics of microscopic or mesoscopic processes so that the macroscopic averaged properties obey the desired macroscopic equations[2].

Due to its kinetic nature, the lattice Boltzmann method stands out from conventional computational fluid dynamics methods. Working with kinetic equations is particularly useful because of the ease in implementation of boundary conditions with arbitrary geometry, as well as the opportunity to parallelize the implementation of the method[2]. In the lattice Boltzmann method, the convection operator is linear in phase space, whereas other numerical schemes making use of macroscopic representations have non-linear convection operators[2]. Using the Chapmann-Enskog expansion, the incompressible Navier-Stokes equations can be recovered from lattice Boltzmann method as briefly shown in Ref.[2]. While using the lattice Boltzmann method, we can make use of the equation of state to calculate pressure[2]. On the other hand, if the incompressible Navier-Stokes equations is directly simulated, additional computations are required to calculate pressure numerically by solving the Poisson equation[2].

Despite its many advantages, the standard lattice Boltzmann method with a single relaxation time(SRT) based on the Bhattnager, Gross, Krook (BGK)[3] collision operator fails to remain stable for low viscosity flows[4]. Alternative models have been proposed to improve stability of the standard lattice Boltzmann method. One approach is called the multiple relaxation time (MRT) model[5, 6, 7]. In this thesis however, we will look at another approach which is referred to as the entropic model[8, 9, 10]. When using an SRT based lattice Boltzmann method for simulating low viscosity flows, the particle distribution functions can become negative which leads to instability in the scheme[11]. The entropic model for the lattice Boltzmann method looks to remove these instabilities by enforcing a discrete analogue of Boltzmann's H-Theorem (see Eq.(2.10)) and thus making sure that all particle distribution functions remain positive throughout the simulation[12]. The aim of the thesis is to study the stability and accuracy of this proposed model.

This thesis is organized in a way in which the reader is firstly presented with the sufficient theory and mathematical background behind the algorithms being used for this thesis. Important aspects regarding their implementation are discussed. The aforementioned points can be found in chapter 2. The results obtained during numerical experiments performed using these algorithms are presented and analyzed in chapter 3. Finally, in chapter 4 some concluding remarks are made on the two schemes and the results presented in the previous chapter.

Chapter 2

Theory

In the following sections, the reader is introduced briefly to the basic notions of fluid dynamics and kinetic theory. Furthermore the algorithms for the standard lattice Boltzmann method and the entropic lattice Boltzmann method are presented and few remarks regarding their implementation are discussed.

1 Fluid dynamics

1.1 Preliminaries

A substance is considered a fluid if it is not able to withstand shear stress: component of stress which is parallel to the surface[13]. A fluid has certain physical properties associated with it, the most significant of which include[13]:

- Velocity $u[ms^{-1}]$,
- Density $\rho[kg/m^3]$,
- Temperature $T[K]$,
- Pressure $p[N/m^2]$.

Another important property of fluids that needs to be mentioned is viscosity. We denote the dynamic viscosity by μ and it refers to the tendency of the fluid to resist deformation under shear stress[13]. The kinematic viscosity ν is the ratio between the dynamic viscosity and the density of the fluid, so $\nu = \frac{\mu}{\rho}$[14].

1.2 Dimensionless numbers

The following dimensionless numbers will come up in our forthcoming analysis:

Reynolds number

The Reynolds number gives us the ratio between inertial and viscous forces[14]:

$$Re = \frac{uL}{\nu},$$

where u is the flow velocity, L is the characteristic length and ν is the kinematic viscosity. When the viscous forces are significantly larger than the inertial forces, the fluid is in the laminar regime[13]. If we keep on increasing the Reynolds number, eventually after some critical value, instabilities will occur inside the fluid. At this point the inertial forces are considerably more dominant than the viscous forces and the fluid has entered the transient regime[13].

Knudsen number

The Knudsen number gives the ratio between the mean free path and the characteristic length of the physical domain[15]:

$$Kn = \frac{\lambda}{L},$$

where λ is the mean free path of a molecule: the average distance traveled by a molecule between successive collisions, and L is the characteristic length. Knudsen number is an indicator of when it is suitable to use continuum mechanics and when to use statistical mechanics to describe the motion of the fluid[16]. Continuum mechanics can not provide a sufficient description of a fluid flow regime with a high Knudsen number[16].

Mach number

The Mach number is the ratio between the local flow velocity and the local speed of sound inside that medium[14]:

$$Ma = \frac{u}{u_s},$$

where u is the local flow velocity and u_s is the local speed of sound. The Mach number is an indicator of whether or not a flow regime can be considered incompressible. For incompressible flows the density within the flow can be considered to be constant and vice versa for compressible flows[14]. The lower the Mach number, the lower the density variations inside the flow[14].

1.3 Navier-Stokes equations

The Navier-Stokes equations (named after Claude Navier and Gabriel Stokes) are a set of equations used to provide a theoretical description of fluids. It is important to note that these equations are only applicable if the so called continuum hypothesis holds[14]. This hypothesis assumes that the size of the physical fluid flow domain is larger than the mean free path between molecules[14]. This is where the Knudsen number plays an important role. A terse derivation will be provided for the Navier Stokes equations below.

We begin with the continuity equation, which states that the conservation of mass must hold inside the fluid domain. For a fixed volume in space, mass conservation states that the inflow of mass inside the volume should be balanced by the rate of outflow of mass through the volume[14]. In its final form the continuity equation is written as[14]:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0. \tag{2.1}$$

For incompressible flows ($\rho \equiv const$[14]) this equation reduces to[14]:

$$\frac{\partial u_i}{\partial x_i} = 0. \tag{2.2}$$

The conservation of momentum is obtained by using Newton's second law of motion. If Newton's second law is applied to an infinitesimal fluid element we end up with the so called Cauchy's equation of motion[14]:

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j}\right) = \rho g_i + \frac{\partial \tau_{ij}}{\partial x_j}. \tag{2.3}$$

The left hand side of Eq.(2.3) represents the advective terms and the right hand side consists of body and surface forces. For Newtonian fluids (viscosity does not depend on the shear stress[14]), the stress tensor τ_{ij} can be written as[14]:

$$\tau_{ij} = -\left(P + \frac{2}{3}\mu \frac{\partial u_i}{\partial x_i}\right)\delta_{ij} + 2\mu e_{ij},$$

where P stands for pressure, μ is the dynamic shear viscosity, $e_{ij} \equiv \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$ is the volumetric strain rate tensor and δ_{ij} represents the Kronecker delta where $\delta_{ij} = 1$ if $i = j$ else $\delta_{ij} = 0$[14]. Now that we have a concrete expression for the stress tensor we can plug this back in to Eq.(2.3). The incompressible Navier-Stokes equations consist of four equations with four unknowns[14]:

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{2.4}$$

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j}\right) = -\frac{\partial P}{\partial x_i} + \mu \frac{\partial^2 u_i}{\partial x_j^2} + \rho g_i. \tag{2.5}$$

2 Kinetic theory

2.1 Boltzmann's equation

At the molecular level for a three dimensional space a molecule can be described by six values $(x_1, x_2, x_3, v_x, v_y, v_z)$[15]. Three values for the position vector $(x_1, x_2, x_3) \equiv \vec{x}$ and three for the velocity vector $(v_x, v_y, v_z) \equiv \vec{v}$[15]. This means that the dynamics of a system with N molecules can be described using N points inside a 6 dimensional phase space $\mu = [(\vec{x}, \vec{v}) : \vec{x}, \vec{v} \in \mathbb{R}^3]$[15]. However, the value of N is usually around the order of Avogadro's number (10^{23}), so it is not feasible to run simulations with such precise detail[15].

We now introduce a probability distribution function $f \equiv f(\vec{x}, \vec{v}, t)$ that gives us the probability of finding a particle at position $\vec{x}$ with velocity $\vec{v}$ at time t inside the volume element $d\vec{x}d\vec{v}$ inside the μ space[15]. Mathematically, we can express the particle distribution function around a volume element $d\vec{x}d\vec{v}$ as[17]:

$$f(\vec{x}, \vec{v}, t)d\vec{x}d\vec{v} = dN,$$

where dN is the probable number of molecules at time t which have positions lying inside the position element $d\vec{x}$ and velocity inside the element $d\vec{v}$. If we were to integrate over all such volume elements in the μ space we would get[17]:

$$\int f(\vec{x}, \vec{v}, t)d\vec{x}d\vec{v} = N.$$

Particle distribution functions lie at the heart of kinetic theory. The aim now is to study the evolution of the particle distribution function which is captured by Boltzmann's equation[17]. Before Boltzmann's equation is presented certain assumptions need to be mentioned[15]:

1) Only two particles are involved during collisions.

2) Particles are point like and structureless.

3) Molecules interact via short range potential.

The Boltzmann equation can now be presented, including an external force $\vec{F}$, as[15]:

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} + \vec{F} \cdot \frac{\partial f}{\partial \vec{v}} = C(f). \tag{2.6}$$

The left hand side of the equation expresses the advection of the particles along the trajectories associated with the external force $\vec{F}$[15]. The right hand side represents the collision operator, which describes what happens during collision between two particles[15]. The collision term is an integral expression and a more detailed account of the exact form of the collision operator can be found in Ref.[17]. In order to simplify the procedure to solve Boltzmann's equation, the collision term $C(f)$ is replaced by a much simpler expression which makes Eq.(2.6) mathematically easier to solve while still preserving the basic physics captured by the equation[15]. One such example of the simplified collision operator is the so-called Bhatnager-Gross-Krook(BGK) operator[3]:

$$C = \omega(f^{eq} - f) \tag{2.7}$$

where f^{eq} is the local equilibrium parametrized by the density ρ, velocity u and temperature T. The variable $\omega = 1/\tau$ where τ is defined as the typical time scale for the collision process to relax the particle distribution functions f towards their local equilibrium f^{eq}.

2.2 Maxwell-Boltzmann equilibrium distribution function

The explanation of the equlibrium distribution function provided in this section is based on the description provided in Ref.[15]. We define the equilibrium distribution f^{eq} as the distribution where the collision operator goes to zero, so $C(f^{eq}) = 0$. This definition allows us to show that $\ln f$ is an "additive collision invariant"[15]. Using Lagrange multipliers we can determine what the value of the distribution function should be at thermodynamic equilibrium. The Lagrange multipliers will depend on density ρ, momentum u_a and energy E. The constraints imposed would be the conservation of mass, momentum and energy:

$$m\int f d\vec{v} = \rho,$$

$$m\int f v_a d\vec{v} = \rho u_a,$$

$$m\int f\frac{v^2}{2} d\vec{v} = \rho E,$$

where m is the mass and u_a is the macroscopic velocity of the flow. The Maxwell Boltzmann equilibrium distribution function for three dimensions is given by:

$$f^{eq} = \rho\left(\frac{m^2}{2\pi K_B T}\right)^{3/2} e^{\frac{(-\vec{v}-\vec{u})^2 m^2}{2K_BT}}, \tag{2.8}$$

where T is the fluid temperature and K_B is the Boltzmann constant.

2.3 Boltzmann's H-Theorem and entropy

Define an H-functional[15]:

$$H(t) = \int f \ln f d\vec{v} d\vec{x}, \tag{2.9}$$

where the variable $f \equiv f(\vec{x}, \vec{v}, t)$ is a particle distribution function[15]. Boltzmann showed that this function is monotonically decreasing in time. Mathematically, this can be expressed as[15]:

$$\frac{dH(t)}{dt} \leq 0, \tag{2.10}$$

where equality only holds at equilibrium. Using the H-functional defined in Equation (2.9) we can define entropy as[19]:

$$S = -H(t). \tag{2.11}$$

In the context of the H-theorem this would imply that entropy is a monotonically increasing function:

$$\frac{dS}{dt} \geq 0, \tag{2.12}$$

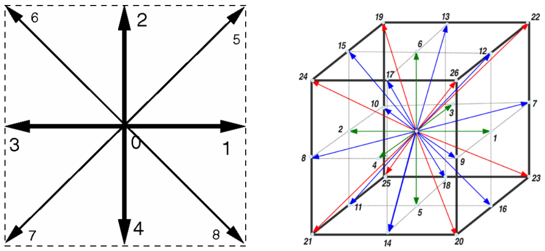
where equality only holds at equilibrium. For a more detailed look at the physical meaning of entropy and its use in the second law of thermodynamics see Ref.[17].

3 Lattice Boltzmann method

3.1 Lattice Boltzmann method algorithm

The naming convention for the lattice models is DnQm, where n is the number of spatial dimensions and m is the discrete velocities associated with that particular lattice model. The choice of these velocities m is not arbitrary as they need to satisfy certain symmetry properties[15]. Throughout this thesis we will be working with the D3Q27 model. A visualization of the model can be seen in Figure(2.1)[21, 20].

In this sub-section, the final discretized Boltzmann equation is presented along with the BGK collision operator and the entire lattice Boltzmann algorithm is summarized. The detailed space time discretization of the Boltzmann equation is not presented in this thesis. Interested readers are advised to see



(a) The D2Q9 model[20].

(b) The D3Q27 model[21].

Figure 2.1: Visualization of lattice model.[20, 21].

Ref.[22]. The final form of the discretized equations, without the external force term, along with the BGK collision operator is[2]:

$$f_i(\vec{x_i} + \vec{c_i}\Delta t, t + \Delta t) = f_i(\vec{x_i}, t) + \omega(f_i^{eq}(\vec{x_i}, t) - f_i(\vec{x_i}, t)), \tag{2.13}$$

where $i = 0, \ldots, m-1$ and m depends on the lattice model DnQm being used, Δt is the lattice time step and hereafter it is normalized in the formulation as $\Delta t = 1$. The variable $\vec{c_i}$ represents the discrete lattice velocities and the variable $\omega \in (0, 2)$ is the relaxation parameter. The template for $\vec{c_i}$ for the case of a D3Q27 lattice model is given in Table(2.1)[23]. The discretized version of the local equilibrium f_i^{eq} can be seen inside the BGK collision operator in Eq.(2.13). Its value can be approximated by considering the Taylor expansion of the equilibrium distribution function in the limit of a low Mach number[2]:

$$f_i^{eq} = \rho w_i(1 + 3\vec{c_i} \cdot \vec{u_i} + 4.5(\vec{c_i} \cdot \vec{u_i})^2 - 1.5(\vec{u_i} \cdot \vec{u_i})), \tag{2.14}$$

where $i = 0, \ldots, m-1$ and m depends on the lattice model DnQm being used and w_i are the weighting factors associated with the discrete lattice velocities $\vec{c_i}$. The weighting factors w_i for the case of the D3Q27 lattice model are given in the Table(2.1).

lattice Direction i	Discrete Velocity $\vec{c_i}$	Weighting factor w_i
0	$(0, 0, 0)$	$\frac{8}{27}$
$1, \ldots, 6$	$(\pm 1, 0, 0)$	$\frac{2}{27}$
$7, \ldots, 18$	$(\pm 1, \pm 1, 0)$	$\frac{1}{54}$
$19, \ldots, 26$	$(\pm 1, \pm 1, \pm 1)$	$\frac{1}{216}$

Table 2.1: Weighting factors w_i and discrete lattice velocities $\vec{c_i}$ associated with the D3Q27 model[23].

The macroscopic variables namely density ρ and velocity $\vec{u}$ also come up in the Eq.(2.14). These variables can be defined using the conservation of mass and momentum.

Conservation of mass[2]:

$$\rho = \sum_{i=0}^{m-1} f_i, \tag{2.15}$$

Conservation of momentum[2]:

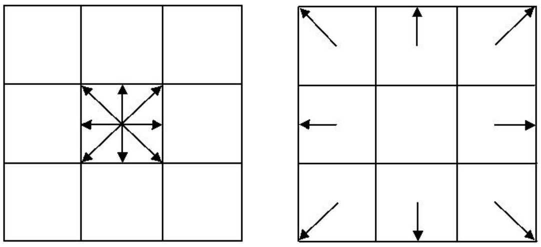
$$\rho\vec{u} = \sum_{i=0}^{m-1} f_i\vec{c_i}, \tag{2.16}$$

where m is the number of the discrete lattice velocities $\vec{c_i}$ depending on the type of lattice model DnQm being used.

Having sufficiently defined all the variables inside the Eq.(2.13),we can now proceed in solving this equation in a two step procedure which would involve collision and streaming.

1. Collision:
$$\tilde{f}_i(\vec{x_i}, t + \Delta t) = f_i(\vec{x_i}, t) + \omega(f_i^{eq}(\vec{x_i}, t) - f_i(\vec{x_i}, t)), \tag{2.17}$$
where for our case $i = 0, \ldots, m-1$ where m depends on the lattice model DnQm being used. In the collision step, particles collide with one another and a new distribution function is calculated. The equilibrium distribution function used in the collision step is taken from Eq.(2.14). The macroscopic density and velocity are conserved but the distribution functions are modified during the collision step.

2. Stream:
$$f_i(\vec{x_i} + \vec{c_i}\Delta t, t + \Delta t) = \tilde{f}_i(\vec{x_i}, t + \Delta t), \tag{2.18}$$
where for our case $i = 0, \ldots, m - 1$ where m depends on the lattice model DnQm being used. In the streaming step, the particles are advected towards their neighbouring cells depending upon their velocity directions[24]. The stream step can be implemented in two ways. Either by pushing the distribution function from one cell on to its neighbours or by having a pull step in each cell which pulls every distribution function pointing towards it[24]. As we see from Figure(2.2), the magnitude of the distribution functions during the streaming step remains the same and according to their directions they move towards their neighboring node.



(a) Distribution before the streaming step.

(b) Distribution after the streaming step.

Figure 2.2: Visualization of the streaming step using the D2Q9 lattice model.

Using multi-scale Chapman-Enskog techniques as exhibited in Ref.[25, 2] we can obtain a value for kinematic viscosity of the D3Q27 lattice model[2]:

$$\nu = c_s^2\left(\frac{1}{\omega} - \frac{1}{2}\right), \tag{2.19}$$

where $c_s = \frac{1}{\sqrt{3}}$ is the speed of sound inside the lattice[22]. It is important to note that c_s does not represent the physical speed of sound in the simulated fluid but the speed at which information travels within the lattice[24]. The lattice model remains stable only for positive viscosity[2]. More precisely, as $\omega \to 2$ the lattice becomes unstable[11]. Further analysis of error sources and the accuracy of the numerical method can be found in the Ref.[26, 27]. As a further remark it should be noted that using the lattice Boltzmann method, the pressure p can be obtained using the equation of state[2]:

$$p = c_s^2\rho, \tag{2.20}$$

where $c_s = \frac{1}{\sqrt{3}}$ is the lattice speed of sound and ρ is the macroscopic density. The algorithm for the lattice Boltzmann method can be summarized in pseudo code so far as follows:

```
\\Initialization
for all grid cells
   initialize density;
   initialize velocity;
   initialize particle distribution functions;
end for
```

```
 \\Two step procedure for solving the lattice Boltzmann equations
 for all time steps
    for all grid cells
      calculate density;
      calculate velocity;
      calculate equilibrium distribution functions;
      do collide;
      do stream;
    end for
end for
```

For the initialization block in the algorithm, the initialization of the macroscopic variables, namely density and velocity, depends on the type of problem being simulated. The particle distribution functions are normally initialized to their corresponding equilibrium distribution functions. Depending on the type of problem, however, it becomes crucial in some cases to initialize particle distribution functions with their density and velocity gradients, which are contained in the non equilibrium components of the distribution functions, as well[28]. This point will become much more clear in the next chapter. After initialization, the macroscopic variables are calculated followed by the two step procedure used to solve the lattice Boltzmann equation Eq.(2.13). The collide step is performed after which the streaming step takes place. A more detailed pseudo code can be found in Ref.[29].

4 Entropic lattice Boltzmann method

4.1 Loss of the H-Theorem and stability

The main problem with the standard lattice Boltzmann scheme is that the algorithm becomes unstable for low viscosity flows. The particle distribution functions do not remain positive throughout the simulation and hence instabilities occur[11]. The problem lies in the equilibrium distribution function Eq.(2.14) being used[18]. In standard lattice Boltzmann schemes the equilibrium distribution function follows a particular polynomial ansatz[30]. However, if we start by defining an H-functional we can use this H-functional to find the exact form of the equilibrium distribution function. Note that the H-functional is minimized, subject to constraints, at the local equilibrium. It was shown in Ref.[30] that if this approach is taken, the equilibrium distribution function never takes the form of the polynomial ansatz followed by the equilibrium distribution function of the standard lattice Boltzmann scheme. Thus, it is not possible to define an H-function on the standard lattice Boltzmann scheme with equilibrium distribution function taking the form of the polynomial ansatz given in Ref.[30]. In order to remedy this issue the entropic lattice Boltzmann scheme defines an equilibrium distribution which is the result of extremizing of a discrete H-function with the constraints of conservation of mass and momentum[11].

It still remains to be discussed how enforcing the H-Theorem would provide enhanced stability to the lattice Boltzmann schemes. It was shown in Ref.[4] that the lattice Boltzmann scheme is stable as long as the particle distribution functions remain close to their corresponding local equilibrium distribution functions. For a system that obeys the H-Theorem, we know by the second law of thermodynamics that the entropy inside that system would always evolve towards its state of maximization. By the H-Theorem(see Eq.(2.10)) this entropy maximization will occur at the local equilibrium. Thus, enforcing an H-Theorem on the lattice Boltzmann scheme would ensure that the particle distribution functions move towards or at least remain close to their local equilibrium distribution functions. Hence, ensuring stability of the scheme. A much more detailed look at the issue of stability in the lattice Boltzmann method could be found in Ref.[31].

4.2 Obtaining macroscopic variables

The steps inside the entropic lattice Boltzmann algorithm that remain unchanged from the SRT based lattice Boltzmann schemes with BGK collision models (from this point on referred to as the SRT LBM model) are the streaming step and the calculation of the macroscopic moments. The key feature of the entropic lattice Boltzmann scheme is to ensure that the growth of entropy during the simulation follows the H-Theorem (see Eq.(2.10)). The formulas for obtaining the macroscopic variables are as follows:

Conservation of mass[2]:

$$\rho = \sum_{i=0}^{m-1} f_i, \tag{2.21}$$

Conservation of momentum[2]:

$$\rho \vec{u} = \sum_{i=0}^{m-1} f_i \vec{c_i}, \tag{2.22}$$

where m is the number of the discrete lattice velocities $\vec{c_i}$ depending on the type of lattice model DnQm being used.

4.3 Discrete H-Theorem and equilibrium distribution functions

In order to enforce a discrete H-Theorem on the lattice Boltzmann method we need to define a discrete analogue of the H-functional (see Eq.(2.9)). In its most general form the discrete H-function can be written as[11]:

$$H = \sum_{i=0}^{m-1} h(f_i), \tag{2.23}$$

where m depends on the DnQm lattice model being used and $h(f_i)$ are convex functions. Since the h functions are convex, for a general variable x the first derivative of the h functions $h'(x) \geq 0$ for all $x > 0$. The discrete H-function is obtained using the Hermite quadrature formulas on the standard continuous Boltzmann's H-function[11]. For a DnQm lattice model, this yields[11]:

$$H = \sum_{i=0}^{m-1} f_i \ln\left(\frac{f_i}{w_i}\right). \tag{2.24}$$

The variable w_i again refers to the lattice weights associated with the discrete lattice velocities for the particular lattice model. Once we have obtained the discrete H-function we can use it to find the equilibrium distribution functions. Recall, that the H-function is minimized at the equilibrium distribution function. We can proceed now by extremizing Eq.(2.24) subject to the constraints of conservation of mass and momentum. Solving this problem yields the equilibrium distribution functions of the form[32]:

$$f_i^{eq} = \rho w_i \prod_{j=0}^{n-1} \left(2 - \sqrt{1+3u_j^2}\right)\left(\frac{2u_j + \sqrt{1+3u_j^2}}{1-u_j}\right)^{c_{i,j}}, \tag{2.25}$$

where $i = 0, \ldots, m-1$ and both the variables n and m depend on the lattice model DnQm being used. The variables w_i are the weighting factors, ρ is the macroscopic density, u_j is the j'th component of the macroscopic velocity $\vec{u}$ and $c_{i,j}$ refers to the j'th component of the discrete lattice velocity $\vec{c_i}$. It should be noted that while working with a 3d lattice model the exact form of the equilibrium distribution function given by Eq.(2.25) can only be used with a D3Q27 lattice model[11, 12]. The values for the weighting factors w_i and the template for the discrete lattice velocities $\vec{c_i}$ for the D3Q27 model can be seen in Table(2.1).

4.4 Relaxation parameters

The major difference in the formulation of the entropic lattice Boltzmann method when compared to the SRT LBM model comes in the equilibrium distribution functions f^{eq} and the relaxation parameters. In the SRT LBM model, there is only one relaxation parameter $\omega \in (0, 2)$ which is responsible for controlling the viscosity during simulations involving the SRT LBM model. In the entropic model, we are solving the following equation[33]:

$$f_i(\vec{x_i} + \vec{c_i}\Delta t, t + \Delta t) = f_i(\vec{x_i}, t) + \alpha\beta(f_i^{eq}(\vec{x_i}, t) - f_i(\vec{x_i}, t)), \tag{2.26}$$

where $i = 0, \ldots, m-1$ and m depends on the lattice model DnQm being used. We can see from Eq.(2.26) that the relaxation parameter ω from Eq.(2.13) is replaced by the product $\alpha\beta$ where α and β are two new relaxation parameters. Eq.(2.26) is again solved by a two step procedure involving collide and stream steps. For the entropic lattice Boltzmann method the collision step is broken down into two further steps. In the first step the particle distribution functions f_i are changed in the direction of the vector $\Delta = f^{eq} - f$, while keeping the discrete H-function (see Eq.(2.24)) constant[34]. In the second step dissipation is introduced to reduce the magnitude of the discrete H-function[34].

4.5 Entropy estimate of over-relaxation

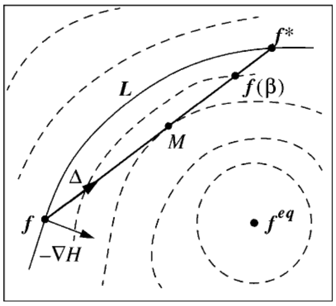


Figure 2.3: Maximal entropy estimate for relaxation[19].

In the SRT LBM model we need to worry about entropy growth during relaxation step[35]. We motivate our argument for entropy estimate by using geometric means. The following argument is based on the explanation provided in Ref.[19]. In Figure(2.3)[19] the original particle distribution function is given by f. The vector Δ points to the direction of local equilibrium. The vector $-\nabla H$ represents gradient of entropy. The contour L is where the entropy value remains constant to the one at the original particle distribution function. The dashed contours represent the local equilibrium. The point M is where H-functional is minimized for the population f. The particle distribution function $f(\beta)$ is the particle distribution function obtained after relaxation. The value of H (see Eq.(2.24)) at $f(\beta)$ is more than that at M but less than that at f. The maximum limit of relaxation is given by the conjugate particle distribution function f^*. As both f and f^* lie on the same contour they have the same entropy value. Going beyond this conjugate particle distribution function during relaxation would violate the H-Theorem (see Eq.(2.10)). Mathematically, we can now proceed by defining an entropic involution[35]:

$$f \longmapsto f^* : f^* = f + \alpha(f^{eq} - f), \tag{2.27}$$

where $\alpha > 0$. The variable α is the non trivial solution of the equation resulting from the entropy estimate of the maximal over-relaxation[35]:

$$H(f + \alpha(f^{eq} - f)) = H(f). \tag{2.28}$$

Note that the H function is evaluated using Eq.(2.24) and the equilibrium distribution function being used is given by Eq.(2.25). For all BGK type collision operator we would also have a vector Δ_{BGK} (not shown in Figure(2.3)) which points from f to f^{eq} and thus the point M will then coincide with the point f^{eq} in Figure(2.3)[19].

4.6 Controlling viscosity

After obtaining the particle distribution function f^* via the entropic involution (see Eq.(2.27)) we can complete the collision step as follows[35]:

$$f(\beta) = (1 - \beta)f + \beta f^*, \tag{2.29}$$

where $\beta \in (0, 1)$ is a relaxation parameter related to the viscosity ν. The parameter β can be defined in terms of the viscosity as[33]:

$$\beta = \frac{1}{6\nu + 1}. \tag{2.30}$$

If we expand f^* the collision step can be compactly written as:

$$\tilde{f}_i(\vec{x_i}, t + \Delta t) = f_i(\vec{x_i}, t) + \alpha\beta(f_i^{eq}(\vec{x_i}, t) - f_i(\vec{x_i}, t)), \tag{2.31}$$

where $i = 0, \ldots, m - 1$ with m depending on the lattice model DnQm being used. The streaming step looks identical for both the algorithms for the entropic and the SRT LBM model (see Eq.(2.18)). It is also important to note that the equilibrium distribution function f^{eq} being used is from Eq.(2.25). Note that by Eq.(2.19) and Eq.(2.30) we can deduce that $\omega = 2\beta$. The collision step (Eq.(2.31)) can now be rewritten in terms of the ω parameter:

$$\tilde{f}_i(\vec{x_i}, t + \Delta t) = f_i(\vec{x_i}, t) + \frac{\alpha\omega}{2}(f_i^{eq}(\vec{x_i}, t) - f_i(\vec{x_i}, t)). \tag{2.32}$$

Thus, we see from Eq.(2.32) that for $\alpha = 2$, the collision step is reduced exactly to the one being used in the SRT LBM model (see Eq.(2.17)). Since α is fixed to the value of 2 in the SRT LBM model we can see by geometric arguments using Figure(2.3) [19] why this model violates the H-Theorem (see Eq.(2.10)). The reason being that the value of $\alpha = 2$ might take us beyond the contour line of the same entropy estimate during relaxation step. The values of α are not necessarily equal to 2 for entropic lattice Boltzmann schemes and they are chosen in a way that the H-Theorem (see Eq.(2.10)) is never violated.

The kinematic viscosity can be obtained in the entropic lattice Boltzmann method using Chapman-Enskog techniques as shown in Ref.[36, 37] and it can be defined by rearranging Eq.(2.30). The apparent viscosity is defined as[38, 11]:

$$\nu_{app} = c_s^2\left(\frac{1}{\alpha\beta} - \frac{1}{2}\right), \tag{2.33}$$

where $c_s = \frac{1}{\sqrt{3}}$ is the lattice speed of sound. Eq.(2.33) is quite similar to Eq.(2.19), the only difference between them is that the ω parameter is replaced by the product $\alpha\beta$. It can also be checked that by plugging in $\alpha = 2$ in Eq.(2.33) followed by some algebraic manipulation would yield Eq.(2.30). The apparent viscosity allows us to understand how the entropic lattice Boltzmann scheme makes a compromise between accuracy and stability. We can distinguish between three cases[39]:

i) Full resolved case : In this case $\alpha = 2$ and the relaxation parameter becomes the same as the standard LBGK model.

ii) Smoother: In this case $\alpha < 2$ and smoothers are responsible for maintaining stability by increasing the apparent viscosity.

iii) Enhancer: In this case $\alpha > 2$ and enhancers are responsible for maintaining accuracy by decreasing the apparent viscosity.

4.7 Entropic lattice Boltzmann method algorithm

The pseudo code for the entropic lattice Boltzmann method algorithm for the BGK model (from this point on referred to as the entropic LBM model) so far looks like as follows[35]:

```
 \\Initialization
 for all grid cells
    initialize density;
    initialize velocity;
    initialize particle distribution functions;
 end for

\\Time loop for solving the entropic lattice Boltzmann equation
for all time steps
     for all grid cells
          Compute density;
          Compute velocity;
          Calculate $f^{eq}$;\\ $f_i^{eq} = \rho w_i \prod_{j=0}^{m-1}\left(2 - \sqrt{1+3u_j^2}\right)\left(\frac{2u_j+\sqrt{1+3u_j^2}}{1-u_j}\right)^{c_{i,j}}$
```

```
            Determine $\alpha$;\\ $H(f + \alpha(f^{eq} - f)) = H(f)$
            do collision;\\$\tilde{f}_i(\vec{x_i}, t + \Delta t) = f_i(\vec{x_i}, t) + \frac{\alpha\omega}{2}(f_i^{eq}(\vec{x_i}, t) - f_i(\vec{x_i}, t))$
            do stream;
        end for
end for
```

In the pseudo code above comments have been added to the parts which are either modified or newly added when compared to the algorithm for the SRT LBM model. It can be seen that the calculation of the equilibrium distribution function and the collision step are modified slightly. The calculation of the α parameter is the new step and it is the most crucial step in accurately implementing the entropic LBM model. A few remarks regarding the implementation of this step are needed. Firstly, it should be noted that in order to find α we need to solve Eq.(2.28), which is a non-linear equation. Thus, in practice either the Newton-Raphson method or the Bisection method or even a hybrid version using both methods is used to solve Eq.(2.28). In our implementation we have used the Newton-Raphson method. Interested readers are recommended to see Ref.[40] for a much more detailed explanation of the aforementioned root solvers for non-linear equations. Solving Eq.(2.28) for each grid point adds to the computation time of the overall algorithm. However, it is important to realize that the optimum value of α, which is found by solving Eq.(2.28), is only needed at a selected number of points where instabilities might occur during the fluid flow simulation. In order to speed up the computation time and to identify the select number of points where solving Eq.(2.28) is necessary, certain optimization techniques have been recommended in the literature (see Ref.[38, 33]). The optimization techniques used in our code will be presented below.

Asymptotic expansion for the solution of α

Firstly, measure the deviation of the particle distribution functions from the equilibrium distribution functions. In pseudo code, this can be written as:

```
function CalculateDeviation($f$,$f^{eq}$)
    for all particle distribution function directions
        if deviation > eps;\\ $\left|\frac{f_i^{eq} - f_i}{f_i}\right| > 10^{-2}$
            return deviation;
        end if
    end for
    return deviation;
end function
```

If the deviation is within a tolerance value do not solve Eq.(2.28). If the deviation is less than 10^{-6}, simply set $\alpha = 2$. If the deviation is between 10^{-2} and 10^{-6} use a series approximation as a value for α which is given by[33, 35]:

$$\alpha = 2 - \frac{4a_2}{a_1} + \frac{16a_2^2}{a_1^2} - \frac{8a_3}{a_1} + \frac{80a_2a_3}{a_1^2} - \frac{80a_2^3}{a_1^3} - \frac{16a_4}{a_1}, \tag{2.34}$$

where the variable a_n for $n \geq 1$ is defined as[33, 35]:

$$a_n = \frac{(-1)^{n-1}}{n(n+1)} \sum_{i=0}^{m-1} \frac{(f_i^{eq} - f_i)^{n+1}}{f_i^n}, \tag{2.35}$$

where m depends on the lattice model DnQm being used.

Optimal starting point for the Newton-Raphson method

Starting points are crucial for the convergence of the Newton-Raphson method. The common starting point chosen for the Newton-Raphson to solve the Eq.(2.28) is taken to be $\alpha_0 = 2$. In Ref.[38] it has been recommended to use the following starting point to save computation time:

$$\alpha_0 = \begin{cases} 2, & t = 0, \\ \alpha_{x,y,z}^{t-1}, & t > 0, \end{cases} \tag{2.36}$$

where $\alpha_{x,y,z}^{t-1}$ refers to the solution to the Eq.(2.28) obtained in the previous time step for grid point (x, y, z).

One last remark that needs to be made is the case when no non trivial solution of Eq.(2.28) exists. Some authors (see Ref.[41]) suggest using the positivity rule, which sets the value of $\alpha = \alpha_{max}$, where α_{max} is the maximum possible value of α for which the particle distribution functions remain positive even after relaxation towards local equilibrium. The parameter α_{max} can be computed as follows[37]:

$$\alpha_{max} = \min_{\substack{i=0,\ldots,m-1 \\ \Delta_i < 0}} \{\alpha \mid f_i + \alpha\Delta_i = 0\}, \tag{2.37}$$

where $\Delta_i = f_i^{eq} - f_i$ and m depends on the lattice model DnQm being used.

5 Simulation of fluid flow problems using lattice Boltzmann method

5.1 Parametrization of fluid flow problems

Parametrization of variables being used in lattice Boltzmann method based simulation is of great importance. The actual values from the physical domain need to be scaled to dimensionless values being used in the simulation. During error analysis it is important to make sure that both the analytical and the numerical solution being considered are in consistent units. That is to say either both values are being presented in physical units or both have been scaled correctly to dimensionless units. The important variables to consider are:

- fluid velocity u $[ms^{-1}]$
- fluid viscosity ν $[m^2s^{-1}]$

In our notation we will denote dimensionless variables with a star superscript (*) and the physical variables without any special superscript. Three variables are normalized to 1 during the entire simulations: lattice time step size Δt^*, lattice density ρ^*, lattice grid spacing Δx^*. This would mean:

$$\Delta t^* = \frac{\Delta t}{\Delta t} = 1,$$

$$\Delta x^* = \frac{\Delta x}{\Delta x} = 1,$$

$$\rho^* = \frac{\rho}{\rho} = 1.$$

Using these variables we can parametrize both the physical fluid velocity and viscosity to lattice variables:

$$u^* = \frac{\Delta t}{\Delta x} u, \tag{2.38}$$

$$\nu^* = \frac{\Delta t}{\Delta x^2} \nu. \tag{2.39}$$

5.2 Boundary conditions

Boundary conditions are the last pieces of the puzzle required to fully simulate fluid flow problems using algorithms based on lattice Boltzmann method. In this thesis two different types of boundary conditions are used for the upcoming numerical experiments. Both of these boundary conditions will be briefly described now.

Periodic Boundary Conditions

Periodic Boundary Conditions are most probably the simplest boundary conditions to implement. The main idea behind periodicity is displayed in Figure(2.4)[34], where periodicity in the x-direction can be seen.

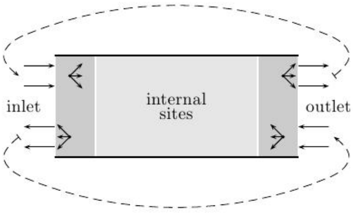


Figure 2.4: Periodic Boundary Conditions[34].

Consider a 2d computational domain $[0, N_x + 1] \times [0, N_y + 1]$, where the total number of fluid cells are $[1, N_x] \times [1, N_y]$ and there are four additional layers that make up the boundary of the domain. In terms of implementation we can proceed in the following manner. The numbering of the D2Q9 model is assumed to be the same as the one provided in Figure(2.1). Periodicity in the x-direction would mean dealing with the west and east boundary. In the east direction this would be done with[15]:

$$\begin{gathered} f_5(0, j) = f_5(N_x, j), \\ f_1(0, j) = f_1(N_x, j), \\ f_8(0, j) = f_8(N_x, j), \end{gathered}$$

where $1 \leq j \leq N_y$. In the west direction the same idea can be used[15]:

$$\begin{gathered} f_6(N_x + 1, j) = f_4(1, j), \\ f_3(N_x + 1, j) = f_5(1, j), \\ f_7(N_x + 1, j) = f_6(1, j), \end{gathered}$$

where $1 \leq j \leq N_y$. The outward particle distribution functions of the fluid cells neighbouring the east boundary of the fluid flow domain are copied to the corresponding particle distribution functions on the west boundary of the fluid flow domain[15]. Similarly, the outward particle distribution functions of the fluid cells neighbouring west boundary of the fluid flow domain are copied to the corresponding particle distribution functions on the east boundary of the fluid flow domain[15]. This idea can be generalized to implement periodicity in arbitrary spatial directions of the domain.

No Slip Boundary Conditions

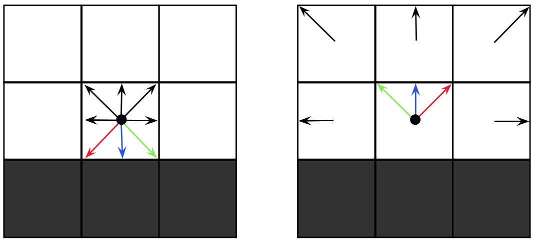
No Slip boundary conditions are used to simulate the scenario where fluid particles are stationary close to the wall[15]. One way to do this is by reflecting back the particle distribution functions to the cell from where they were coming from as depicted in Figure(2.5). This schema is referred to as the bounce back rule. The bounce back will now be described in a more mathematically precise manner. Let x represent the cell neighbouring a boundary cell. Assuming that the boundary wall is stationary, the particle distribution function at cell x would be updated during the streaming step as follows[42]:

$$f_i(x, t+1) = f_{\bar{i}}(x, t). \tag{2.40}$$

Let $\vec{c_{\bar{i}}}$ denote the discrete lattice velocity in the direction opposite to the discrete lattice velocity $\vec{c_i}$, then the particle distribution function $f_{\bar{i}}$ refers to the particle distribution function in the same direction as the discrete lattice velocity $\vec{c_{\bar{i}}}$. Note from Figure(2.5) that the particle distribution functions in cell x which are not moving in the direction of the boundary wall will be streamed as they normally are in the algorithm. For the case of a moving boundary wall the streaming step at cell x will be modified to[43]:

$$f_i(x, t+1) = f_{\bar{i}}(x, t) + 6\rho w_i \vec{c_i} \vec{u}_w. \tag{2.41}$$

The notation for $f_{\bar{i}}$ holds the same meaning as it did in Eq.(2.40). The variable ρ refers to the macroscopic density, w_i are the lattice weighting functions and $\vec{u}_w$ is the boundary wall velocity.



(a) Distribution at the boundary node before streaming.

(b) Distribution at the boundary node after streaming.

Figure 2.5: Visualization of the no slip boundary conditions using the D2Q9 lattice model.

Chapter 3

Numerical experiments with different benchmark problems

In the following sections, different numerical experiments are performed based on standard benchmark problems. The results obtained during these experiments by the entropic and the SRT LBM models are compared with particular emphasis on their stability and accuracy. In order to truly check the significance of utilizing the correct α parameter, for all set of experiments both models used the same equilibrium distribution function given by Eq.(2.25). The simulations are carried out inside the waLBerla[1](widely applicable Lattice Boltzmann solver from Erlangen) framework which is developed at the Chair for System Simulation at the Friedrich-Alexander-Universität Erlangen-Nürnberg.

1 Taylor-Green vortex

1.1 Description and setup of the problem

The Taylor-Green vortex is an initial value problem commonly used to test the accuracy of numerical methods used to simulate fluid flow. It was first mentioned in Ref.[44]. It has been extensively studied in Ref.[45]. The Taylor-Green vortex has the following analytical solution[28]:

$$u_x(x,y,t) = -\frac{1}{A}\cos(Ax)\sin(By)\exp(-2\alpha\nu t), \tag{3.1}$$

$$u_y(x,y,t) = \frac{1}{B}\sin(Ax)\cos(By)\exp(-2\alpha\nu t), \tag{3.2}$$

$$p(x,y,t) = -\frac{1}{4}\left(\frac{\cos(2Ax)}{A^2} + \frac{\cos(2By)}{B^2}\right)\exp(-4\alpha\nu t), \tag{3.3}$$

where ν is the shear viscosity and $\alpha = \frac{A^2+B^2}{2}$. The constants A and B are set to the values $A = 1$ and $B = 1$. The physical domain of the problem is $[0, 2\pi] \times [0, 2\pi]$. In Ref.[28] the Taylor-Green vortex was used to study the changes in the relative error in velocity when the particle distribution functions are initialized in different ways. Usually, the particle distribution functions are initialized to their corresponding equilibrium distribution functions. In order to avoid initialization errors Ref.[28] suggests to initialize the particle distribution functions with both their corresponding equilibrium distribution functions as well as their non-equilibrium components. In this thesis we have attempted the same experiments and compared the results obtained by the entropic model to the results obtained in Ref.[28]. We also compared the results obtained by both the entropic and the SRT LBM model to one another. For our experiments firstly, the computational domain under consideration is $[0, 30] \times [0, 30] \times [0, 30]$. Periodic boundary conditions are used in each spatial direction. The value of viscosity remains fixed at $\nu = 1$. The lattice viscosity can be set using Eq.(2.39). The x and y components of velocity are initialized by plugging in $t = 0$ to Eq.(3.1) and (3.2). The z component of velocity is initialized as $u_z = 0$. All three

[1] https://www10.informatik.uni-erlangen.de/en/Research/Projects/walberla/description.shtml

components are then scaled to lattice units using Eq.(2.38). A similar procedure is used to initialize the density, by first using Eq.(3.3) to calculate the initial pressure and then the initial density is calculated using Eq.(2.20). Appropriate scaling is then used to scale density to lattice units. Two different values were used for the time step size $\Delta t = 0.001$ and 0.025. The relative error in the velocity is calculated as follows[28]:

$$u^E = \frac{\sum_{x,y,z} |u_x - u_x^*|}{\sum_{x,y,z} |u_x^*|} + \frac{\sum_{x,y,z} |u_y - u_y^*|}{\sum_{x,y,z} |u_y^*|}, \tag{3.4}$$

where the $(^*)$ superscript represents the analytical solution for that component of velocity.

1.2 Results

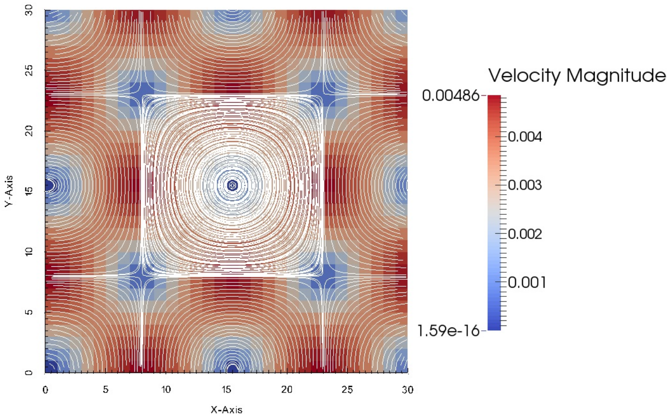


Figure 3.1: Visualization of the velocity magnitude and streamlines for the Taylor-Green vortex.

Figure(3.1) shows the streamlines and the magnitude of the velocity field of the Taylor-Green vortex flow obtained by the entropic LBM model after 10 time steps. The results are quite similar to the Figure 1a) in Ref.[28]. As the results produced by both models are quite similar we have only presented the results from the entropic LBM model in Figure(3.2). The first two plots in Figure(3.2) show the relative errors u^E for velocity obtained by the entropic LBM model after 10 time steps with two different time step sizes $\Delta t = 0.001$ and 0.025. For the first two plots the y-axis represents the value of u^E and the x-axis represents the current time step. In each plot in Figure(3.2) the green line represents the standard initialization of the particle distribution functions:

$$f_i = f_i^{eq}, \tag{3.5}$$

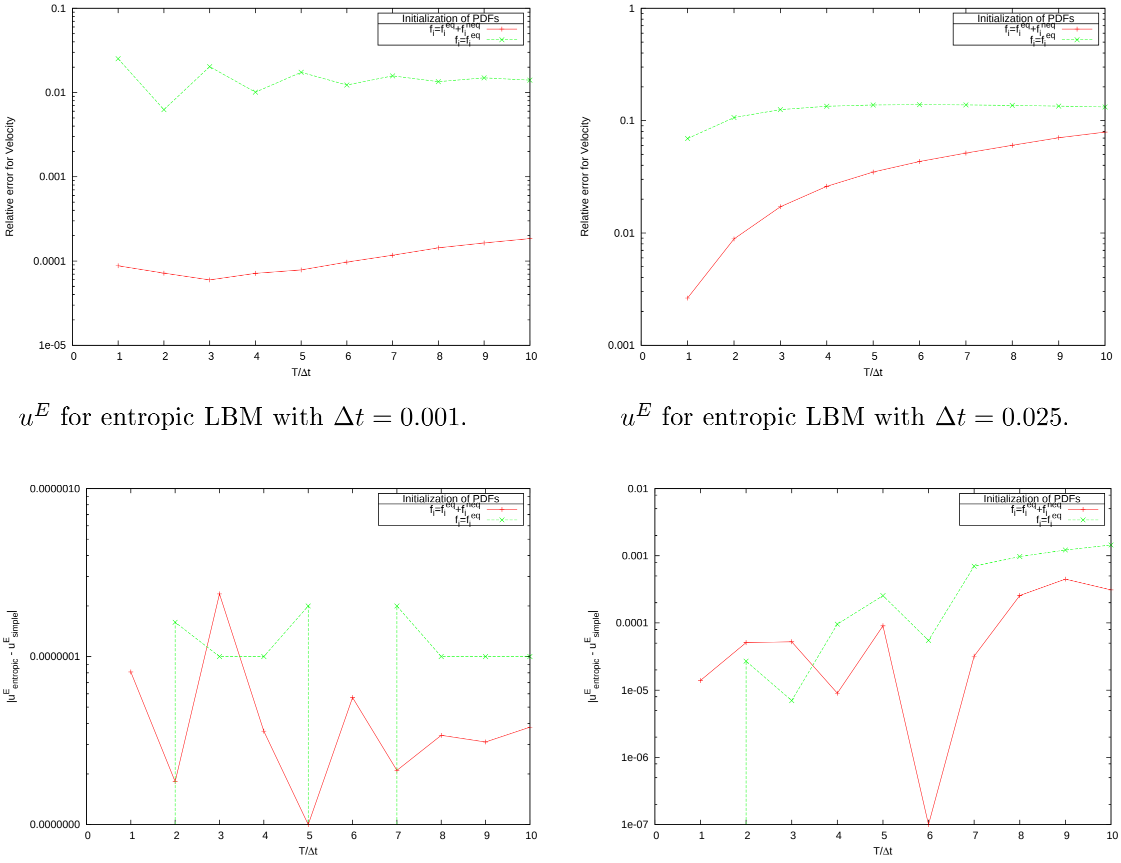
meaning that only the equilibrium components are included. The red line on the other hand represents for each plot in Figure(3.2) a different initialization of the particle distribution functions:

$$f_i = f_i^{eq} + f_i^{neq}, \tag{3.6}$$

one which includes both the equilibrium and non equilibrium components. The non equilibrium components are defined as[46]:

$$f_i^{neq} = -\frac{1}{\omega} \Delta t w_i \frac{\rho_0}{c_s^2} c_i c_i : \nabla u, \tag{3.7}$$

where $\rho_0 = 1$ and ∇u is the gradient of the velocity field defined by Eq.(3.1) and Eq.(3.2). The ":" operator is the standard double dot product. It can be seen that the relative error is reduced if the non equilibrium components are included in the initialization. The results obtained are in agreement with the results presented as Figure 2a) and 2b) in Ref.[28].



u^E for entropic LBM with $\Delta t = 0.001$.

u^E for entropic LBM with $\Delta t = 0.025$.

$\mid u^E_{entropic} - u^E_{simple} \mid$ with $\Delta t = 0.001$.

$\mid u^E_{entropic} - u^E_{simple} \mid$ with $\Delta t = 0.025$.

Figure 3.2: Relative error u^E for velocity measured for different initialization schemes using entropic LBM model with $\Delta t = 0.001$ and 0.025 for the Taylor-Green vortex problem. We have also compared the deviation in the results obtained by both the entropic and the SRT LBM model for different initialization schemes and different time step sizes.

The second set of plots shows the absolute difference in the relative errors in velocity obtained by using both the simple SRT LBM model and the entropic LBM model after 10 time steps with two different time step sizes. The y-axis represents the deviation $\mid u^E_{entropic} - u^E_{simple} \mid$ and the x-axis represents the current time step. Since the value of the viscosity is not very small, the Newton-Raphson method was not triggered during the course of the simulation and only the series approximation of the α parameter was used. Three different α values were used in the collision step: $\alpha = 1.9999, 2$ and 2.00001. Only two values of alpha were calculated which were not equal to 2. As these values are extremely close to 2 the results obtained by the entropic LBM model do not show much deviation from the results obtained using the SRT LBM model.

1.3 Conclusion

In Ref.[28] it was claimed that by initializing the particle distribution functions with their corresponding equilibrium and non equilibrium components the relative error for velocity can be reduced significantly. Similar experiments to Ref.[28] involving the Taylor-Green vortex were carried out using the entropic LBM model. The results produced are in agreement to the figures presented in Ref.[28]. For this test case we also expected to see the results from the entropic and the SRT LBM model to be quite close to one another and our experiments also validated that claim.

2 1d shock tube problem

2.1 Description and setup of the problem

The 1d shock tube problem has been used extensively in Ref.[4, 47, 48, 41, 49] to test the stability of Lattice Boltzmann schemes. The physical domain under consideration for the 1d shock tube problem is $x \in [0, 1]$. In general, the 1d shock tube problem is initialized as follows:

$$(\rho, u)|_{t=0} = \begin{cases} (\rho_L, 0), & 0 \leq x \leq 0.5, \\ (\rho_R, 0), & 0.5 < x \leq 1. \end{cases} \tag{3.8}$$

At time $t = 0$, all components of velocity u are set to zero. The density ρ on the other hand is initialized to two separate values ρ_L and ρ_R. Half of the physical domain has density $\rho = \rho_L$ and the other half has density $\rho = \rho_R$. This type of initialization for the density causes a point of discontinuity at the center of the domain. As a result a shock emerges and in this experiment we observe how this shock travels throughout the domain. Our experiments are based on the previous simulations presented in Ref.[47]. The computational domain being used is $[0, 800] \times [0, 4] \times [0, 4]$. In order to truly simulate a 1d setup we used periodic boundary conditions in the y and z-direction. Bounce back boundary conditions are used in the x-direction. The ratio between ρ_L and ρ_R is [1 : 2]. The density ρ is initialized as follows:

$$\rho(x) = \begin{cases} 1.0, & 0 \leq x \leq 400, \\ 0.5, & 401 \leq x \leq 800. \end{cases} \tag{3.9}$$

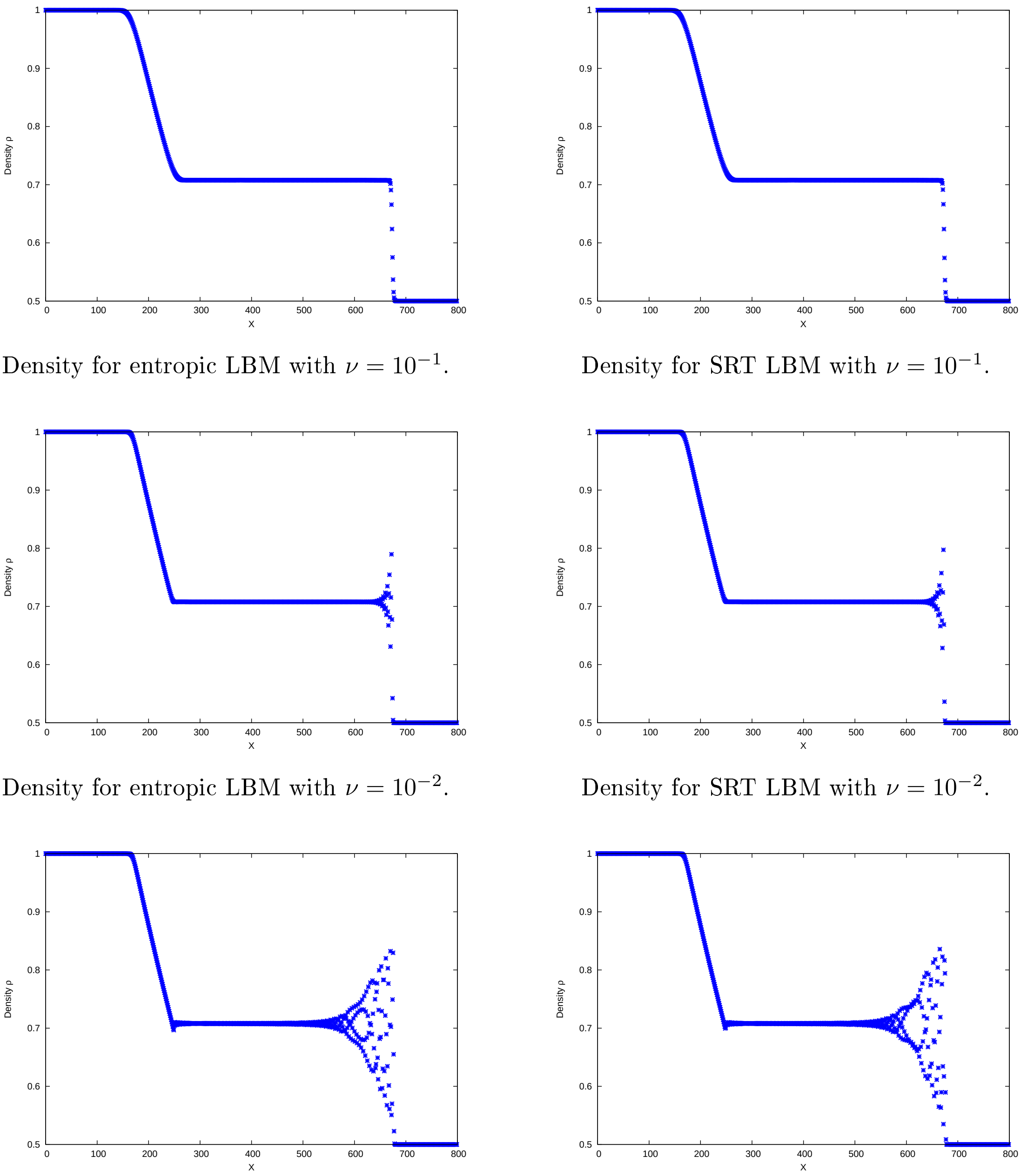
All components of velocity u are initialized to zero. This experiment is carried out for 400 time steps after which the density profile is measured. For each set of experiments the kinematic viscosity ν is changed. The values for viscosity used for these experiments are $\nu = 10^{-1}$, 10^{-2}, 10^{-3} and 10^{-9}. For each set of experiments the total entropy at each timestep is also recorded. In order to calculate total entropy inside the domain, we are summing up the value of the H function at each grid point and then using Eq.(2.11) to determine entropy. The formula used for calculating the total entropy inside the domain is:

$$S = -\sum_{x,y,z} \sum_{i} f_i \ln \left(\frac{f_i}{w_i} \right). \tag{3.10}$$

The results presented in Ref.[47] showed that both the entropic and the SRT LBM model could not fully eradicate the oscillations in their density profiles. The viscosity values used in Ref.[47] were $\nu = 3.3333 \times 10^{-2}$ and 10^{-9}. These results are contrary to what was presented in Ref.[37] and Ref.[19], where experiments were performed using $\nu = 3.3333 \times 10^{-2}$ and the entropic LBM model did not have large oscillations unlike the SRT LBM model. In Ref.[47] it was also shown that even when using low viscosity the entropic LBM model never violates the H-Theorem whereas the SRT LBM model does not preserve the H-Theorem for low viscosity flows.

2.2 Results

Figure(3.3) shows the density profiles obtained during our numerical experiments using viscosity $\nu = 10^{-1}, 10^{-2}$ and 10^{-3}. For each figure the plots on the left show the results obtained using the entropic LBM model and the plots on the right show the results obtained using SRT LBM model. In every plot the y-axis represents the density ρ and the x-axis represents the current grid point in the x-direction of the computational domain. The first set of plots depict the density profile after 400 time steps using $\nu = 10^{-1}$. The results obtained show that the density remains smooth. There are no large wiggles or oscillations found in the obtained density profiles. This was to be expected since a high value for viscosity was being used. The value of viscosity is decreased then to 10^{-2}. For this case, most of the density profile is still smooth, however the first instance of oscillations in the density profiles can be observed for these set of experiments in the interval $x \in [650, 672]$. The density profiles obtained by both models are quite similar. Very slight damping of the oscillations was attained when using the entropic LBM model. The viscosity is further reduced to 10^{-3}. For this value of viscosity, part of the density profile is still smooth. Naturally, the oscillatory behavior of the density profile has increased as a low value for viscosity



Density for entropic LBM with $\nu = 10^{-1}$.

Density for SRT LBM with $\nu = 10^{-1}$.

Density for entropic LBM with $\nu = 10^{-2}$.

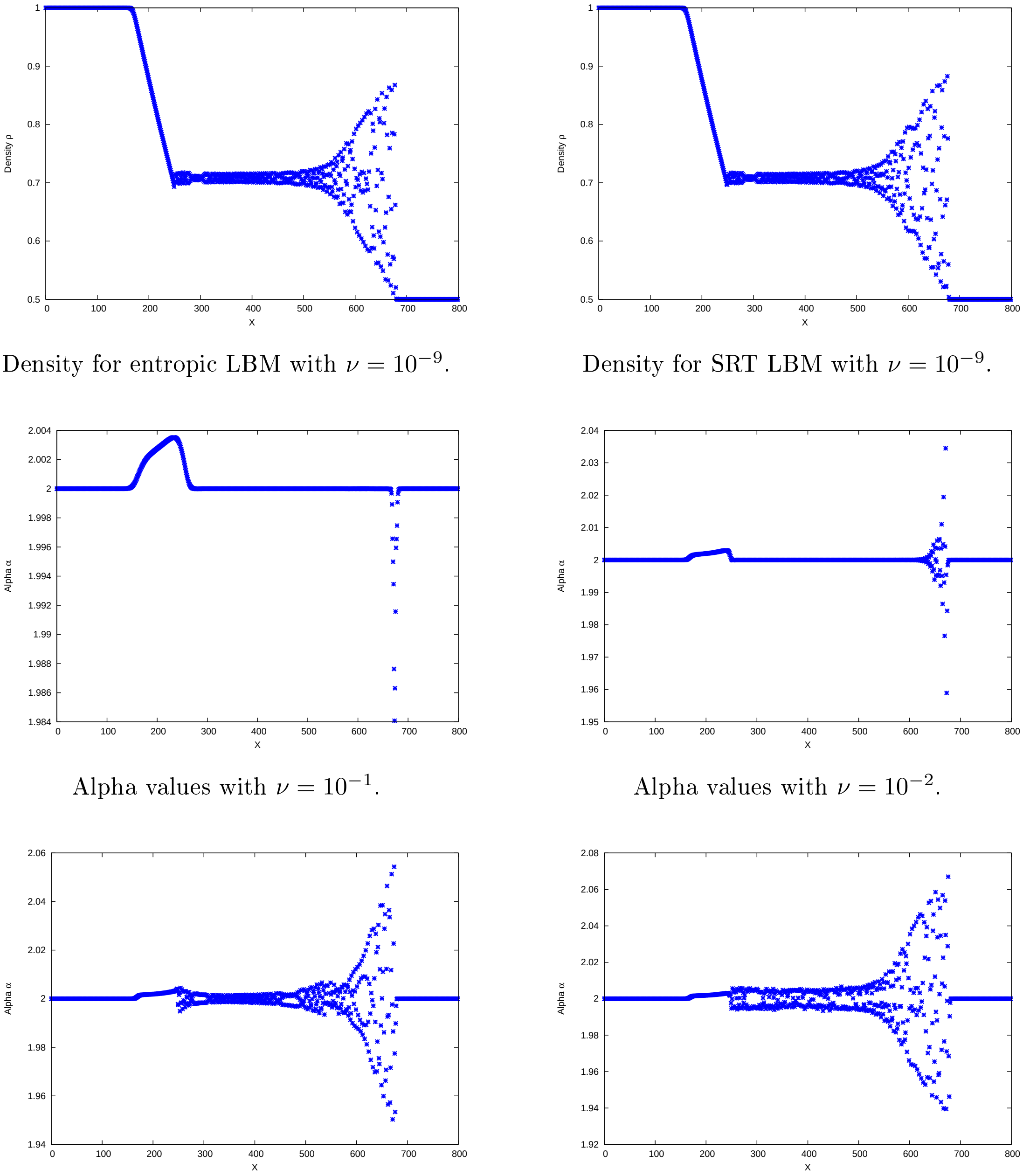
Density for SRT LBM with $\nu = 10^{-2}$.

Density for entropic LBM with $\nu = 10^{-3}$.

Density for SRT LBM with $\nu = 10^{-3}$.

Figure 3.3: Density profile measured after 400 time steps using the entropic and SRT LBM models with $\nu = 10^{-1}$, 10^{-2} and 10^{-3} for the 1d shock tube problem.

is being used. Large oscillations in particular can be seen in the interval $x \in [560, 680]$. Though, the results from both models have oscillations in the density profile it can clearly be seen that the oscillations are much more damped in the interval $x \in [580, 640]$ for the entropic LBM model. In Figure(3.4) the first two plots show the density profiles for $\nu = 10^{-9}$. If the viscosity is reduced to 10^{-9} even more oscillations are observed in the density profile. These oscillations are larger in amplitude compared to the ones attained at higher viscosity values and can be seen in the interval $x \in [500, 682]$. Even though the density profiles recorded by both models were containing oscillations, it can be observed that within the interval $x \in [600, 640]$, the oscillations have a smaller amplitude for the entropic LBM model.



Density for entropic LBM with $\nu = 10^{-9}$.

Density for SRT LBM with $\nu = 10^{-9}$.

Alpha values with $\nu = 10^{-1}$.

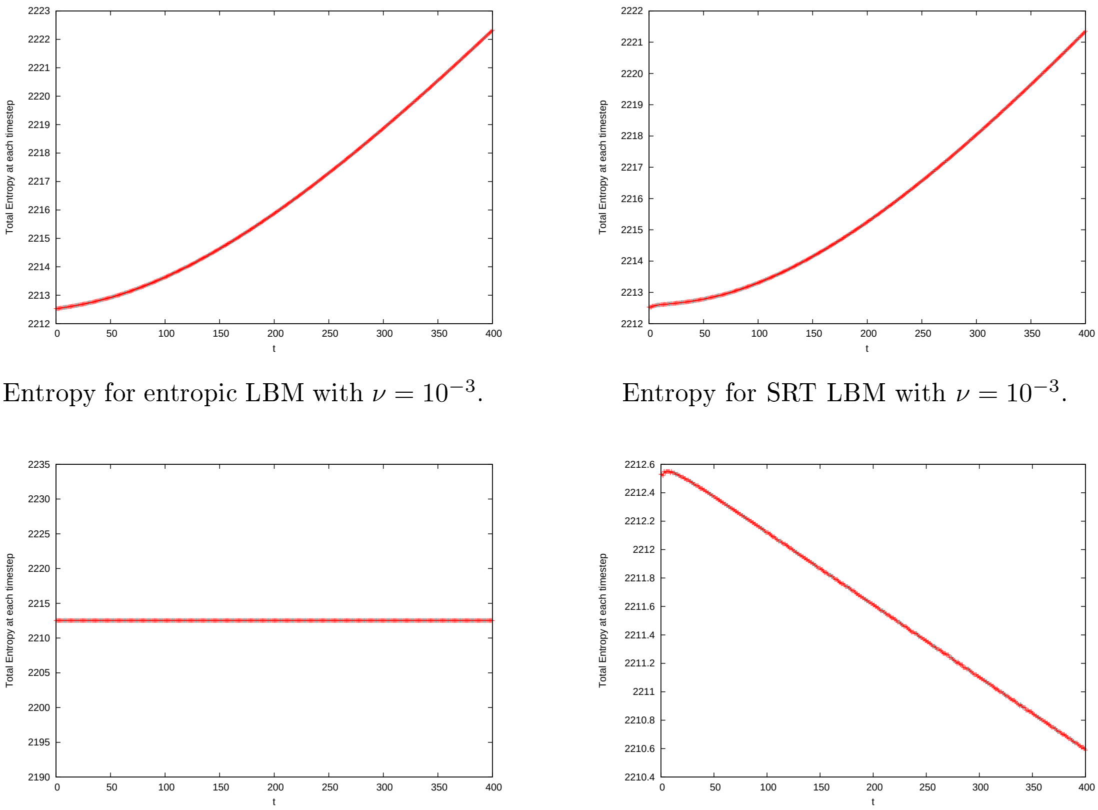
Alpha values with $\nu = 10^{-2}$.

Alpha values with $\nu = 10^{-3}$.

Alpha values with $\nu = 10^{-9}$.

Figure 3.4: Density profile measured after 400 time steps using the entropic and SRT LBM models with $\nu = 10^{-9}$ for the 1d shock tube problem. The different values of the alpha parameter calculated at the final timestep with $\nu = 10^{-1}, 10^{-2}, 10^{-3}$ and 10^{-9} are also shown.

In Figure(3.4), the final four plots show the alpha values used during collision in the final timestep for different values of viscosity. The y-axis for each plot represents the alpha value and the x-axis represents the current grid point in the x-direction of the computational domain. It was observed for $\nu = 10^{-1}$, the alpha values were in the range $\alpha \in [1.98407, 2.00352]$. The alpha parameter was calculated via the series approximation 329 times and all calculations were in the interval $x \in [133, 686]$. The Newton-Raphson method was used 8 times and the calculations took place in the interval $x \in [670, 677]$. For the case of $\nu = 10^{-2}$, the alpha values were in the range $\alpha \in [1.95893, 2.03447]$. The alpha parameter was calculated



Entropy for entropic LBM with $\nu = 10^{-3}$.

Entropy for SRT LBM with $\nu = 10^{-3}$.

Entropy for entropic LBM with $\nu = 10^{-9}$.

Entropy for SRT LBM with $\nu = 10^{-9}$.

Figure 3.5: Total entropy in the system measured after 400 time steps using entropic and SRT LBM models with $\nu = 10^{-3}$ and 10^{-9} for the 1d shock tube problem.

266 times using the series approximation within the $x \in [151, 650]$. The Newton-Raphson method was utilized 23 times and the calculations were within the interval $x \in [651, 675]$. When the viscosity was reduced to $\nu = 10^{-3}$, the alpha parameter was in the range $\alpha \in [1.95031, 2.05433]$. It was calculated 405 times using the series calculation and all calculations were recorded in the interval $x \in [156, 680]$. The Newton-Raphson method was used 120 times and these calculations were utilized in the interval $x \in [250, 677]$. Finally for $\nu = 10^{-9}$ the alpha parameter was in the range $\alpha \in [1.9395, 2.06698]$. The alpha parameter was calculated via series approximation 158 times in the interval $x \in [156, 682]$ whereas the Newton-Raphson method was utilized 369 times in the interval $x \in [250, 679]$. In Figure(3.5) the total entropy inside the system after 400 time steps is shown. The plots were made for values of viscosity for which extremely large oscillations began to occur in our experiments. For each plot in the figure the y-axis represents the total entropy inside the system and the x-axis represents the current time step. It was observed that for $\nu = 10^{-9}$ the H-Theorem is not preserved by the SRT LBM model, whereas the entropic LBM model is still able to maintain the H-Theorem.

2.3 Conclusion

The 1d shock tube experiment was performed to test the stability of the entropic and the SRT LBM model. For each experiment the value of viscosity was decreased. The results obtained regarding the density profile of the simulation showed oscillations for both models used. This is in agreement with what was presented in Ref.[47]. We also observed that the oscillations obtained using the entropic LBM model were much smaller in amplitude at certain points when compared to the oscillations obtained using the SRT LBM model. It was also recorded that even at extremely low viscosity values the entropic LBM model preserves the H-Theorem unlike the SRT LBM model.

3 Lid driven cavity flow

3.1 Description and setup of the problem

The lid driven cavity flow problem is used as a standard benchmark for testing the stability of different methods. It has been extensively tested in Ref.[50, 51]. This benchmark simulates a closed square box of length L which is completely filled with fluid being at rest at the beginning of the simulation. Except for the top wall all other walls are at rest throughout the simulation. The top wall, which is referred to as the lid, is constantly dragged across the fluid with a lid velocity u. This leads to the vortex formation inside the flow. A visualization of the setup of the lid driven cavity can be seen in Figure(3.6). Working with higher Reynolds number for this problem will lead to more intense vortex formation. The Reynolds number for this problem is defined as follows:

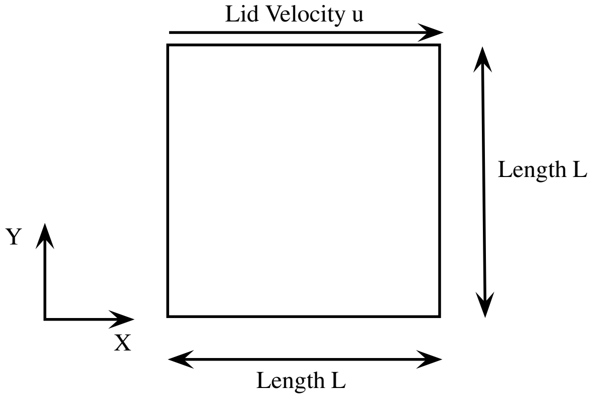


Figure 3.6: Setup of the lid driven cavity.

$$Re = \frac{uL}{\nu}, \tag{3.11}$$

where u is the lid velocity, L is the length of the lid and ν is the kinematic viscosity. The physical domain of the problem is $[0,1] \times [0,1]$. The discretized computational domain that we are using for our simulation is $[0,100] \times [0,100] \times [0,3]$. In order to truly simulate a 2d problem we use periodic boundary conditions in the z-direction. In the x and y-direction the bounce back schema is used. We have tried to replicate the experiments presented in Ref.[34]. The lid velocity is fixed to $u = (0.1, 0, 0)$ in lattice units. All components of the velocity inside the fluid are set to 0 and the macroscopic density is initialized as $\rho = 2.7$. These experiments have been carried out with Reynolds number $Re \in \{400, 1000, 2000\}$. For all set of experiments the lid velocity $u = (0.1, 0, 0)$ and the dimension of lid $L = 100$ are fixed. In order to simulate a higher Reynolds number the viscosity ν is reduced. Every simulation was run for 150000 time steps. The velocity magnitude and streamlines are then compared to the results provided in Ref.[34].

3.2 Results

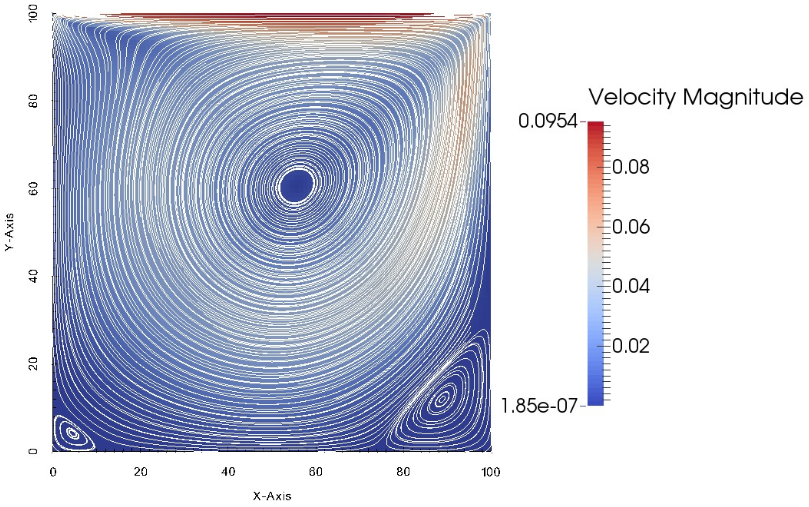


Figure 3.7: Velocity magnitude and streamlines for $Re = 400$.

As the results obtained are quite similar for both models, only the figures based on the SRT LBM model are presented. In Figure(3.7) the velocity magnitude and streamlines can be seen for the lid driven cavity flow with $Re = 400$. A primary vortex can be seen as well as two small secondary vortices at the bottom left and right corner. The position of the primary vortex is also not directly at the center of the domain but slightly towards the right hand side. This figure is in agreement with the results presented in Ref.[34] for $Re = 400$. For the entropic LBM model the values of the alpha parameter were in the range $\alpha = [1.979, 2.045]$.

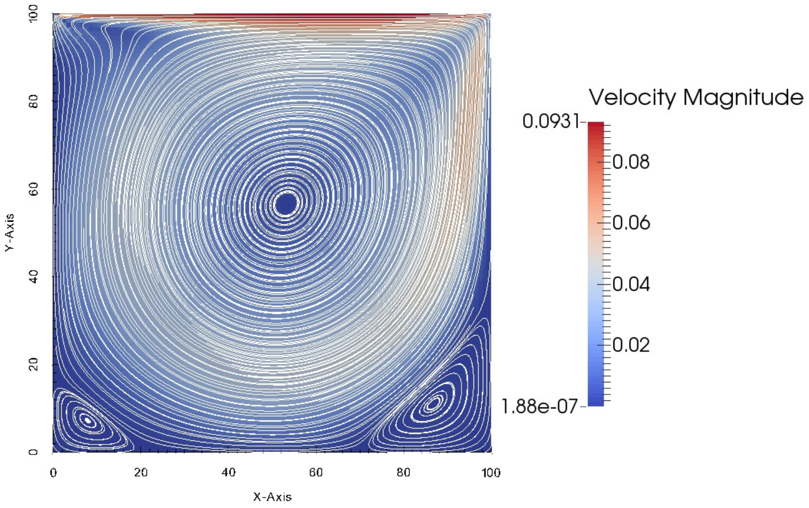


Figure 3.8: Velocity magnitude and streamlines for $Re = 1000$.

In Figure(3.8) the velocity magnitude and streamlines can be seen for the lid driven cavity flow with $Re = 1000$. Again we observe a primary vortex along with two secondary vortices at the bottom left and right corner. It can be observed that the secondary vortices are much larger in size as compared to the secondary vortices obtained for $Re = 400$. The position of the primary vortex is now slightly more towards the center of the domain. This figure is in agreement with the results presented in Ref.[34] for $Re = 1000$. For the entropic LBM model the values of the alpha parameter were in the range $\alpha = [1.979, 2.04]$

As we increased the value of the Reynolds number to 2000, we observed that the SRT LBM model no longer remained stable. Although the entropic LBM model remained stable, the results obtained were inaccurate. Note that by inaccurate we mean that the velocity magnitude and in particular the streamlines obtained did not match the results obtained when performing the same experiment but with a much larger computational domain in the x and y-direction. In an attempt to remedy the problem a new set of experiments was carried out using a smaller lid velocity $u = (0.05, 0, 0)$. Once again the SRT LBM model became unstable and the entropic LBM model remained stable even for 500000 time steps. The results obtained using the entropic LBM model however were still inaccurate. In order to further investigate the problem it was found out that after a certain number of time steps the particle distribution functions became negative. Negative particle distributions are particularly problematic for the entropic LBM model because then it is not mathematically possible to compute the H-function using Eq.(2.24) and thus the Newton-Raphson method fails. Two fail safe measures were adopted in our implementation. These involve either setting the value of $\alpha = 2$ or using the positivity rule. It was observed that solely using the positivity rule gives inaccurate results even for $Re = 1000$. Thus, perhaps the best option is to use a combination of both fail safe options in order to maintain stability and accuracy. Note that negative particle distribution functions essentially force us to use incorrect values for the alpha parameter during the collision step and this is what contributes to the inaccuracy of the results. For the entropic LBM model equipped with the positivity rule the particle distribution functions remained positive after the collision step but only became negative after streaming. This lead us to believe that the boundary conditions being used are not suitable for these set of experiments. Overall we were unsuccessful in fully reproducing the results shown in Ref.[34] where it was shown that for this grid resolution the entropic LBM model can accurately simulate flows upto $Re = 7500$, whereas the SRT LBM model can not.

3.3 Conclusion

For these sets of experiments accurate results were obtained for flows with $Re = 400, 1000$. We were unable to produce accurate results for flows with $Re > 1000$. We did observe that the entropic LBM model exhibited more stability as compared to the results produced SRT LBM model. Further investigation still needs to be made into our implementation and perhaps a different type of boundary condition needs to be utilized.

Chapter 4

Concluding remarks and outlook

In this thesis the entropic LBM model was tested for its stability and accuracy. Three different numerical tests were used for for this purpose. Our first set of experiments was based on the Taylor-Green vortex. As the analytic solution of the Taylor-Green vortex is readily available these set of experiments were geared to test the accuracy of the entropic LBM model and to make sure that our implementation particularly with regards to calculating the alpha parameter is correct. We tried to replicate the experiments performed in Ref.[28] and successfully produced results similar to the ones presented in Ref.[28]. The second set of experiments was based on the 1d shock tube. Our setup was based on the experiments performed in Ref.[47]. The results produced were quite similar to the ones presented in Ref.[47]. We observed that both the entropic and the SRT LBM model had oscillations in their density profiles. In particular for the entropic LBM model these oscillations were much more damped at certain points in the computational domain, as compared to the oscillations produced by the SRT LBM model at the same location. Finally, we performed experiments based on the lid driven cavity flow problem. The experiments were based on the setup provided in Ref.[34]. We were unable to produce accurate results for flows with $Re > 1000$ for this experiment. It was however observed that the entropic LBM model is much more stable than the SRT LBM model. We suspect that choice of boundary conditions utilized was wrong and perhaps a much more detailed investigation needs to be performed regarding this matter. The results obtained by using the entropic LBM model have been promising with regards to stability but further improvements need to be made to the present implementation of the entropic LBM model. Improved boundary conditions could be used to further enhance the stability of the model. One example of such boundary conditions is recommended in Ref.[52] and these conditions are known as the Tamm-Mott-Smith(TMS) boundary conditions. The computational run time of the implementation can also be reduced by replacing the exact entropic equilibrium distribution functions with a product-form approximation as suggested in Ref[33]. In Ref.[53, 54] it was shown via numerical experiments that the computational run time of the implementation of the entropic LBM model could be improved further by replacing iterative methods used to find the alpha parameter with direct approximation methods based on the Taylor series approximation.

List of Figures

Bibliography

[1] Milovan Peric and Joel H. Ferziger. *Computational Methods for Fluid Dynamics.* Springer Berlin Heidelberg, 2001.

[2] Shiyi Chen and Gary D. Doolen. LATTICE BOLTZMANN METHOD FOR FLUID FLOWS. *Annual Review of Fluid Mechanics*, 30(1):329–364, 1998.

[3] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94(3):511–525, 1954.

[4] Jeremy Levesly, Alexander N. Gorban, and David Packwood. A Numerical Analyst's View of the Lattice Boltzmann Method. In *Approximation Algorithms for Complex Systems*, 2011.

[5] D. D'Humières. Generalized Lattice Boltzmann Equations. *Rarefied Gas Dynamics: Theory and Simulation*, pages 450–458, 1980.

[6] I. Ginzburg, F. Verhaeghe, and D. D'Humières. Two-relaxation-time Lattice Boltzmann scheme:About parametrization, velocity, pressure and mixed boundary condtions. *Communications in Computational Physics*, 3(2):427–478, 1989.

[7] P. Lallemand and L. S. Luo. Theory of the Lattice Boltzmann method:Dispersion, dissipation , isotropy, galilean invariance and stability. *Phys. Rev. E*, 61(6):6546–6562, 2000.

[8] I. V. Karlin, A. N. Gorban, S. Succi, and V. Boffi. Maximum Entropy Principle for Lattice Kinetic Equations. *Phys. Rev. Lett.*, 81(1):6–9, 1998.

[9] I. V. Karlin and S. Succi. Equilibria for discrete kinetic equations. *Phys. Rev. E*, 58(4):R4053–R4056, 1998.

[10] I. V. Karlin, A. Ferrante, and H. C. Ottinger. Perfect entropy functions of the Lattice Boltzmann method. *EPL (Europhysics Letters)*, 47(2):182–188, 1999.

[11] Brian Keating, George Vahala, Jeffery Yepez, Min Soe, and Linda Vahala. Entropic lattice Boltzmann representations required to recover the Navier-Stokes flows. *Phys. Rev. E*, 75(3):036712, 2007.

[12] G. Vahala, B. Keating, M. Soe, J. Yepez, L. Vahala, J. Carter, and S. Ziegeler. MHD Turbulence Studies using Lattice Boltzmann Algorithms. *Communications in Computational Physics*, 4(3):624–646, 2008.

[13] Michael Griebel. *Numerical Simulation in Fluid Dynamics: A Practical Introduction.* SIAM: Society for Industrial and Applied Mathematics, 1997.

[14] Pijush.K. Kundu and Ira.M. Cohen. *Fluid Mechanics.* Academic Press, 1990.

[15] Sauro Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond.* Oxford University Press, 2001.

[16] Patrick Tabeling. *Introduction to Microfluidics.* Oxford University Press, 2005.

[17] Kerson Huang. *Statistical Mechanics.* John Wiley and Sons, 1987.

[18] Joost Geerdink. Entropic and multiple relaxation time lattice Boltzmann methods compared for time harmonic flows. Master's thesis, University of Amsterdam, 2008.

[19] Sauro Succi, Illiya V. Karlin, and Hudong Chen. Colloquium: Role of the H theorem in lattice Boltzmann hydrodynamic simulations. *Reviews of Modern Physics*, 74(4):1203–1220, 2002.

[20] A. J. Wagner and G. Kaehler. Fluctuating ideal-gas lattice Boltzmann method with fluctuation dissipation theorem for nonvanishing velocities. *Phys. Rev. E*, 87(6):063310, 2013.

[21] Sauro Succi, Nasrolalh Moradi, Andreas Greiner, and Simone Melchionna. Lattice Boltzmann modeling of water-like fluids. *Frontiers in Physics*, 2(22), 2014.

[22] Xiaoyi He and Li-Shi Luo. A priori derivation of the lattice Boltzmann equation. *Phys. Rev. E*, 55(6):R6333–R6336, 1997.

[23] P. Dellar. Lattice Boltzmann algorithms without cubic defects. *Journal of Compuational Physics*, 259(0):270–283, 2014.

[24] Jan Götz. Numerical Simulation of Bloodflow in Aneurysms using the Lattice Boltzmann Method. Master's thesis, University of Erlangen-Nuremberg, July 2006.

[25] James Buick. *Boltzmann Methods in Interfacial Wave Modelling*. PhD thesis, University of Edinburgh, 1997.

[26] Abdel Monim Mohamed Ali Mohamed Hassan Artoli. *Mesoscopic Computational Haemodynamics*. PhD thesis, University of Amsterdam, 2003.

[27] Jonas Lätt. *Hydrodynamic Limit of Lattice Boltzmann Equations*. PhD thesis, University of Geneva, 2007.

[28] P. A. Skordos. Initial and boundary conditions for the lattice Boltzmann method. *Phys. Rev. E*, 48(6):4823–4842, 1993.

[29] Nils Thuerey. A single-phase free-surface Lattice Boltzmann Method. Master's thesis, University of Erlangen-Nuremberg, June 2003.

[30] A. J. Wagner. An H -theorem for the lattice Boltzmann approach to hydrodynamics. *EPL (Europhysics Letters)*, 44(2):144, 1998.

[31] R. A. Brownlee, A. N. Gorban, and J. Levesley. Stability and stabilization of the lattice Boltzmann method. *Phys. Rev. E*, 75(3):036711, 2007.

[32] S. Ansumali, I. V. Karlin, C. E. Frouzakis, and K. B. Boulouchos. Entropic lattice Boltzmann method for microflows. *Physica A: Statistical Mechanics and its Applications*, 359(0):289 – 305, 2006.

[33] S. S. Chikatamarla, S. Ansumali, and I. V. Karlin. Entropic Lattice Boltzmann Models for Hydrodynamics in Three Dimensions. *Phys. Rev. Lett.*, 97(1):010201, 2006.

[34] Francesca Tosi. *Entropic kinetic schemes with enhanced linear stability*. PhD thesis, University of Florence, April 2007.

[35] I. V. Karlin, S. Ansumali, C. E Frouzakis, and S. S. Chikatamarla. Elements of the Lattice Boltzmann Method I: Linear Advection Equation. *Communications in Computational Physics*, 1(4):616–665, 2006.

[36] Santosh Ansumali and Illiya V. Karlin. Stabilization of the lattice Boltzmann method by the H theorem: A numerical test. *Phys. Rev. E*, 62(6):7999–8003, 2000.

[37] Santosh Ansumali and Illiya V. Karlin. Entropy function approach to the Lattice Boltzmann Method. *Journal of Statistical Physics*, 107(1-2):291–308, 2002.

[38] Francesca Tosi, Stefano Ubertini, S. Succi, and I.V Karlin. Optimization strategies for the Entropic Lattice Boltzmann Method. *Journal of Scientific Computing*, 30(3):369–387, 2007.

[39] I. V. Karlin, S. Succi, and S. S. Chikatamarla. Comment on "Numerics of the lattice Boltzmann method: Effects of collision models on the lattice Boltzmann simulations". *Phys. Rev. E*, 84(6):068701, Dec 2011.

[40] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical Mathematics*. Springer-Verlag New York, 2000.

[41] R. A. Brownlee, A. N. Gorban, and J. Levesly. *Add-ons for Lattice Boltzmann Methods: Regularization, Filtering and Limiters*, chapter Part II: Regularization, Asymptotic Analysis and Lifting of Lattice Boltzmann Methods, pages 29–51. Bentham Science Publishers, 2012.

[42] Dieter A. Wolf-Gladrow. *Lattice-Gas Cellular Automata and Lattice Boltzmann Models*. Springer, 2005.

[43] Li-Shi Luo, Wei Liao, Xingwang Chen, Yan Peng, and Wei Zhang. Numerics of the lattice Boltzmann method: Effects of collision models on the lattice Boltzmann simulations. *Phys. Rev. E*, 83(5):056710, 2011.

[44] G. I. Taylor and A. E. Green. Mechanism of the Production of Small Eddies from Large Ones. *Proc. R. Soc. Lond. A*, 158:499–521, 1937.

[45] M. E. Brachet, D. Meiron, S. Orszag, B. Nickel, R. Morf, and U. Frisch. The Taylor-Green vortex and fully developed turbulence. *Journal of Statistical Physics*, 34(5-6):1049–1063, 1984.

[46] Zhaoli Guo and T. S. Zhao. Explicit finite-difference lattice Boltzmann method for curvilinear coordinates. *Phys. Rev. E*, 67(6):066709, 2003.

[47] A. N. Gorban R. A. Brownlee and J. Levesly. Nonequilibrium entropy limiters in lattice Boltzmann methods. *Physica A*, 387(2-3):385–406, 2008.

[48] A. N. Gorban R. A. Brownlee and J. Levesly. Stable simulation of fluid flow with high-Reynolds number using Ehrenfests' steps. *Numerical Algorithms*, 45(1-4):389–408, 2007.

[49] Dave Packwood. Entropy balance and dispersive oscillations in lattice Boltzmann models. *Phys. Rev. E*, 80(6):067701, 2009.

[50] U. Ghia, K. N. Ghia, and C. T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Compuational Physics*, 48(3):387–411, 1982.

[51] J. Kim and P. Moin. Application of a Fractional -Step Method to Incompressible Navier-Stokes Equation. *Journal of Compuational Physics*, 59(2):308–323, 1985.

[52] S. S. Chikatamarla and I. V. Karlin. Entropic lattice Boltzmann method for turbulent flow simulations: Boundary conditions. *Physica A: Statistical Mechanics and its Applications*, 392(9):1925 – 1930, 2013.

[53] Ye Yu and Li Kenli. Entropic Lattice Boltzmann Method based high Reynolds number flow simulation using CUDA on GPU. *Computers and Fluids*, 88(0):241–249, 2013.

[54] T. Yasuda and N. Satofuka. An improved entropic lattice Boltzmann model for parallel computation. *Computers and Fluids*, 45(1):187–190, 2011.