LATTICE GAS AND LATTICE BOLTZMANN METHODS FOR FLUCTUATING SYSTEMS, BARRIER COATINGS, AND OVERRELAXATION

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Kyle Thomas Strand

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Physics

July 2022

Fargo, North Dakota

NORTH DAKOTA STATE UNIVERSITY

Graduate School

Title

LATTICE GAS AND LATTICE BOLTZMANN METHODS FOR FLUCTUATING SYSTEMS, BARRIER COATINGS, AND OVERRELAXATION

By

Kyle Thomas Strand

The supervisory committee certifies that this dissertation complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Prof. Alexander Wagner
Chair

Prof. Alan Denton

Prof. Andrew Croll

Prof. Nikita Barabanov

Approved:

July 22, 2022
Date

Prof. Alan Denton
Department Chair

ABSTRACT

In the field of computational fluid dynamics, lattice gas and lattice Boltzmann methods are powerful simulation methods derived from kinetic theory. These methods are renowned for their simplicity of implementation and computational speed. In recent years, lattice Boltzmann has risen in popularity for modeling hydrodynamic flows, diffusion, and more. However, a limitation of these methods is the lack of fluctuations due to the continuous nature of the model. Fluctuations arise from the discreteness found in nature, so including fluctuations presents difficulties. This dissertation explores new and novel ways of improving lattice Boltzmann and lattice gas methods.

First, we present a new derivation for a fluctuating lattice Boltzmann method in a diffusive system. Fluctuations are absent lattice Boltzmann since they were derived as a Boltzmann average of discrete lattice gases. This lattice Boltzmann method is exact and includes density dependent noise which models fluctuations to high accuracy.

Second, we extend diffusive lattice Boltzmann methods to apply to physical systems for diffusion through barrier coatings. We found that these models were able to reproduce the behavior from previous experiments and provided a simple tool for analyzing such systems. Higher order corrections to lattice Boltzmann methods are explored for extending the range for successful lattice Boltzmann implementations.

Recently, the implementation of an integer lattice gas with a Monte Carlo collision operator by Blommel *et al.* provided a template for incorporating fluctuations through the discrete nature of lattice gases. A sampling collision operator for integer lattice gases by Seekins *et al.* was able to reproduce the fluctuating diffusion equation in the Boltzmann limit similar to the diffusive fluctuating lattice Boltzmann. However, lattice gases have a more limited range of transport coefficients than lattice Boltzmann methods, since lattice Boltzmann collisions are deterministic and allow for the implementation of over-relaxation and lattice gas collisions are probabilistic and overrelaxation in a lattice gas requires a probability greater than 1.

The final section of this dissertation presents a simple method for including overrelaxation into an integer lattice gas using the sampling collision operator. It will be shown that this is possible through a permutation of occupation numbers.

ACKNOWLEDGEMENTS

The completion of this dissertation was difficult. The academic and intellectual challenges were consistent and demanding. Beyond this, there were many extremely trying hardships I experienced outside the walls of the institution. Although each of these challenges may have seemed insurmountable at the time, I am thoroughly happy to be able to present this completed manuscript. None of this would have been possible without the unwavering support of many individuals throughout this period.

First, I want to thank my advisor, Alexander Wagner. He set me down this path and provided the tools and knowledge required to complete the research. I am also very grateful that he continued to push me to be better, even when I asked him not to.

I offer deep thanks to my collaborators and friends Aaron Feickert, Reza Parsa, and Kent Ridl. The camaraderie and support from these three were an integral part of my success and I hope that they can say the same for me.

I must acknowledge my parents, Scott and Caroline Strand, who have never stopped believing and supporting me, even when I couldn't do so in myself.

I give special thanks to my longtime friend Kurt van Donselaar. We started down this path together and without his friendship and honesty, life would be extremely dull.

And finally, I must express my deepest gratitude to Clare Palmer. Her love and strength is a consistent source of motivation and self-confidence. With her, I've learned what true partnership means and I'm grateful to be able to share my life with her.

DEDICATION

I dedicate this dissertation to my children Grace and Gwendolyn. Their constant flow of inspiration gives me drive, focus, and determination.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDIX FIGURES

1. INTRODUCTION

Since its inception in the late 1980s, the lattice Boltzmann method has been consistently gaining traction as a computational method[1, 2]. With applications in fluid dynamics and beyond, the method has shown to be reliable in accuracy and computational efficiency. Lattice Boltzmann methods were originally developed from lattice gas cellular automata methods. Doing so eliminated fluctuations removing the need for averaging of multiple simulations. It also allowed for a freely chosen equilibrium distribution, which corrected some artifacts of the Boolean lattice gas methods. The number of physical systems which are successfully modeled by lattice Boltzmann methods are steadily growing and thus interest in applications of the methods is growing.

Lattice gas cellular automata (LGCA) methods were first introduced for simulating hydrodynamics by Frisch et al in 1986 [3]. This LGCA employs discrete particles on a hexagonal lattice. The particles are restricted to motion along the six nearest neighboring lattice sites with respect to a central point. LGCA treats the state of each lattice site as a Boolean operation in the manner that there exists or does not exist a particle at that lattice site. The particles will "stream" or propagate to the nearest lattice point which is determined by the velocity of the particle. The method does not allow for occupation of more than one particle per lattice site [4]. If multiple particles occupy a single point, then they will undergo a collision, which is defined by a set of rules designed to conserve mass and momentum.

Due to the discrete particle nature of LGCA, fluctuations are inherently included in the method. However, the noise associated with the discrete particles is difficult to control [5]. In principle noise can be suppressed by increasing the number of particles per lattice site, however traditional LGCA methods utilized Boolean occupation numbers. This is possible with integer valued occupation number but this drastically affects the computational performance of the method. Since LGCA methods were developed with the intention of simulating macroscopic flow of fluids, the statistical noise which arises from LGCA was considered a drawback to the method [6].

Lattice Boltzmann methods, introduced by McNamara and Zanetti in 1988 [7], were developed as an alternative to LGCA methods. Instead of employing individual particles on a lattice, lattice Boltzmann uses continuous distributions of particles. These particle distributions in lat-

tice Boltzmann can be thought of as ensemble averages of particle occupation numbers. These general distributions will remove the statistical noise which posed problems for the hydrodynamic simulation in the LGCA methods. Thus, lattice Boltzmann became a commonly used method for simulating hydrodynamics over the LGCA methods.

Since lattice Boltzmann methods remove fluctuations, this poses a problem for studying systems in which fluctuations are important. Ladd presented a method which reintroduces fluctutations to the lattice Boltzmann method by studying colloidal suspensions in lattice Boltzmann [8]. By reintroducing fluctuations, this spurred an active sub-field in lattice Boltzmann research with the purpose of correctly implementing these fluctuations. These methods are known as fluctuating Lattice Boltzmann methods.

Lattice Boltzmann has been applied to many physical systems beyond hydrodynamics including diffusion [9, 10, 11], electrostatics [12], and quantum mechanics [1]. Lattice Boltzmann is a versatile computational method which is constantly in development.

This introductory chapter will present the lattice Boltzmann method as derived from the continuous Boltzmann equation rather than as the Boltzmann limit of a lattice gas. We will then derive the hydrodynamic equations directly from the lattice Boltzmann equation. Following this, we will briefly review how fluctuations have been reintroduced into the LBM.

Since fluctuations occur more naturally in discrete systems, we return our focus back to LGCA with the goal to introduce recently developed integer lattice gases. To do this, we first introduce the original Boolean lattice gases and then show how these models have been extended to integer lattice gases.

Following the introductory chapter, we then present three published papers that form the very core of this manuscript which are followed by brief concluding remarks.

1.1. The Lattice Boltzmann Equation

In its original inception, the lattice Boltzmann equation (LBE) was intended to model the hydrodynamics equations. Unlike traditional computational fluid dynamics methods which directly solve the Navier-Stokes equation, LBE does not directly solve these equations of motion. Rather, a discretized form of the Boltzmann equation is introduced which utilizes discrete-velocity distributions of particles. It can be shown through standard kinetic theory methods that the Navier-Stokes equation can be recovered in the hydrodynamic limit. [13, 2, 14].

1.1.1. The Discrete-Velocity Distribution Functions

In order to accurately model these equations, we introduce a discrete-velocity particle distribution $f_i(\boldsymbol{x}, t)$, which we will refer to as the distribution function [13, 2]. The distribution function represents the density of particles moving with a discrete velocity set $\{\boldsymbol{v}_i\}$ at position $\boldsymbol{x}$ and time t. The velocity set $\{\boldsymbol{v}_i\}$ is a set of vectors which connect the points on the lattice in various ways and are used to determine the motion of the particles which are represented by the distribution functions. The distribution functions can be used to find macroscopic quantities of a system through weighted sums known as the velocity moments of $f_i(\boldsymbol{x}, t)$. The conserved quantities which can be defined through the velocity moments cannot be relaxed by a collision operator. For a hydrodynamic system, the density $\rho(\boldsymbol{x}, t)$ and momentum $\rho(\boldsymbol{x}, t)\boldsymbol{u}(\boldsymbol{x}, t)$ can be defined through

$$\rho(\boldsymbol{x}, t) = \sum_i f_i(\boldsymbol{x}, t) \tag{1.1}$$

$$\rho(\boldsymbol{x}, t)\boldsymbol{u}(\boldsymbol{x}, t) = \sum_i \boldsymbol{v}_i f_i(\boldsymbol{x}, t). \tag{1.2}$$

The distribution functions are discretized in position, velocity, and time, and are defined on a chosen lattice which has a lattice spacing Δx. The f_i are also defined only at discrete time steps Δt. In practice, it is common to define the lattice spacing and discrete time step as $\Delta x = 1$ and $\Delta t = 1$. The subscript i in f_i refers to a corresponding velocity from a chosen weighted set of velocity vectors $\{\boldsymbol{v}_i\}$.

The dimensionality is usually referred to by the DdQq nomenclature where d represents the total number of spatial dimensions of the lattice and q represents the total number velocities in the chosen set. Commonly employed LBM dimensions consist of D1Q3, D2Q9, D3Q15, and D3Q27, however there are more possible combinations of spatial dimensions and velocity sets which can be employed [2]. Increasing the number of spatial dimensions of the lattice and the number of velocities can increase the accuracy of the algorithm, but these increases impact the computational efficiency of the algorithm.

1.1.2. The Lattice Boltzmann Algorithm

The general lattice Boltzmann algorithm follows two steps: a collision step and a streaming step [2, 14]. The collision step is a process in which the distribution functions are locally redis-

tributed according to the rules of the collision operator. In general, the collision operator is written as a function of the set of distribution functions $\{f_i\}$ and is defined as $\Omega_i(\{f_j\})$. The streaming step is the movement of the distribution functions over the lattice according to the chosen velocity set $\{\boldsymbol{v}_i\}$. The streaming step represents the discrete advancement of time by time step Δt. Combining the collision and streaming steps, the general lattice Boltzmann equation (LBE) can be written in the form

$$f_i(\boldsymbol{x} + \boldsymbol{v}_i\Delta t, t + \Delta t) = f_i(\boldsymbol{x}, t) + \Omega_i(\{f_j\}). \tag{1.3}$$

A further requirement for the lattice Boltzmann equation is that $\boldsymbol{x} + \boldsymbol{v}_i\Delta t$ must also refer to a lattice point.

The LBE will govern the evolution of the distribution functions. The desired macroscopic quantities can be extracted from the distribution functions subject to the moments being employed. For hydrodynamics, Eqns. (1.1–1.2) can be utilized to acquire density ρ and momentum $\rho\boldsymbol{u}$ respectively.

1.1.3. The Collision Operator

In order to model the dynamics of a desired system, the distribution functions must be modified by particle collisions [5, 2, 15]. These collisions are defined by a general collision operator $\Omega_i(\{f_j\})$. The collision operator in any lattice Boltzmann implementation must not modify the conserved quantities of the system being modeled. For the case of hydrodynamics, the collision operator must not change the mass and momentum. This can be achieved by requiring

$$\sum_i \Omega_i(\{f_j\}) = 0 \tag{1.4}$$

$$\sum_i \boldsymbol{v}_i\Omega_i(\{f_j\}) = 0. \tag{1.5}$$

The collision operator can take many forms. Two commonly implemented collision operators in LBM are the multi-relaxation time (MRT) collision operator and the Bhatnagar, Gross, Krook (BGK) collision operator [16]. Both of these collision operators use the concept of particle collisions to relax the f_i to a local equilibrium distribution, f_i^0, by a characteristic relaxation time τ_i. In the case of the MRT collision operator, the subscript i refers to the specific mode being relaxed.

The following section will formally introduce the MRT collision operator and show that the BGK collision operator is a simplification of the MRT collision.

1.1.3.1. The Multi-Relaxation Time Collision Operator and Moment Space Representation

The multi-relaxation time (MRT) collision operator requires an eigenvector basis called moment space [2]. The MRT collision operator follows the idea that for each individual degree of freedom in the distribution function, the collision will relax each relevant degree of freedom towards equilibrium by a specific relaxation time τ_i, where the subscript i corresponds to the quantity of interest. The MRT collision allows for independent access to each moment. In the case of FLBM, the independent relaxation of each mode is highly desirable since the MRT collisions can offer a higher degree of accuracy and numerical stability. To gain access to the individual moments, a moment space representation is introduced. The moment space representation is a transformation of the f_i from velocity space in to moment space. The MRT collision operator will revise the lattice Boltzmann algorithm so that the collisions will take place in moment space. The f_i are first transformed into moment space where the collisions are performed. Then, the distribution functions are transformed back into velocity space where streaming can take place.

The transformation of the f_i from velocity to moment space is written

$$M^a = \sum_i m_i^a f_i \tag{1.6}$$

where M^a is the distribution function moment corresponding to a specific parameter and m_i^a is a transformation matrix. To transform back to velocity space from moment space, the back transform is

$$f_i = \sum_a n_i^a M^a. \tag{1.7}$$

In the back transform, n_i^a is also a transformation matrix. In many cases, this is equivalent to the matrix in the forward transformation, but in general $m_i^a \neq n_i^a$.

These transformations have the orthogonality relations

$$\sum_i n_i^a m_i^b = \delta^{ab}, \quad \sum_a m_i^a n_j^a = \delta_{ij}. \tag{1.8}$$

The subscripts refer to a velocity space representation and the superscripts refer to the moment space representation. These transformations allow for the free movement of the distribution functions between velocity and moment space.

The elements of the transformations matrices must be known to properly transform between f_i and M^a. As is the case in the single-relaxation time BGK collision, the collisions must not modify the conserved quantities of the system. For example, the zeroth moment which we denote as mass will give the transformation

$$M^0 = \rho = \sum_i \mathbf{1} f_i \tag{1.9}$$

which is equivalent to saying that the first row of the transformation matrix is

$$m_i^0 = (1, ..., 1) \tag{1.10}$$

for a system of n velocities. For any DdQq representation, a Gram-Schmidt orthonormalization can be performed to find the remaining elements of the transformation matrix.

If a single-relaxation time is assumed for all moments, the velocity space collision matrix takes the form

$$\Lambda_{ij} = \frac{1}{\tau}\delta_{ij}. \tag{1.11}$$

In moment space, the collision matrix is written

$$\Lambda^{ab} = \sum_{ij} m_i^a \Lambda_{ij} n_j^b = \frac{1}{\tau} m_i^a \delta_{ij} n_j^b = \frac{1}{\tau}\delta^{ab}. \tag{1.12}$$

To generalize this to a multi-relaxation time method, it is imposed that Λ^{ab} be a diagonal matrix

$$\Lambda^{ab} = \frac{1}{\tau^a}\delta^{ab} \tag{1.13}$$

where τ^a is the relaxation time which corresponds to moment a.

The MRT collision operator takes the form

$$\Omega_i(\{f_j\}) = \sum_j \Lambda_{ij}(f_j^0 - f_j)\Delta t \tag{1.14}$$

where Λ_{ij} is a collision matrix with eigenvalues given by the relaxation times. In practice, this is often performed by first transforming f_i into an eigenvector basis of Λ_{ij}, then perform the relaxation, and then back transform into the velocity space. There is a special case where the relaxation times are equal. This is known as the Bhatnagar, Gross, and Krook (BGK) collision operator [16]. For equal eigenvalues, Λ_{ij} simplifies to

$$\Lambda_{ij} = \frac{1}{\tau}\delta_{ij}. \tag{1.15}$$

Since this is a diagonal matrix containing only a single relaxation time τ, Eqn. (1.14) can be rewritten as

$$\Omega_i(\{f_i\}) = \frac{1}{\tau}(f_i^0 - f_i)\Delta t. \tag{1.16}$$

The BGK collision operator is simple and it is sufficient for many situations. Its simplicity makes it very common in practical LBM implementations.

Using the BGK operator in Eqn. (1.16), the Eqn. (1.3) can be written in the form

$$f_i(\boldsymbol{x} + \boldsymbol{v}_i\Delta t, t + \Delta t) = f_i(\boldsymbol{x}, t) + \frac{1}{\tau}(f_i^0 - f_i)\Delta t. \tag{1.17}$$

For the remainder of the manuscript, a discrete time step will be chosen such that $\Delta t = 1$. This leads to the lattice Boltzmann equation with BGK operator

$$f_i(\boldsymbol{x} + \boldsymbol{v}_i, t + 1) = f_i(\boldsymbol{x}, t) + \frac{1}{\tau}(f_i^0 - f_i). \tag{1.18}$$

The following section will derive the discrete local equilibrium distribution f_i^0 from the Maxwell-Boltzmann distribution and discuss the velocity set $\{\boldsymbol{v}_i\}$ for a chosen lattice Boltzmann implementation.

1.1.4. The Local Equilibrium Distribution

The local equilibrium distribution f_i^0 can be found by a second order expansion of the Maxwell-Boltzmann distribution,

$$f^0(\rho, \boldsymbol{u}, \boldsymbol{v}, \theta) = \frac{\rho}{(2\pi\theta)^{D/2}} e^{\frac{-(\boldsymbol{v}-\boldsymbol{u})^2}{2\theta}} \tag{1.19}$$

where $\theta = k_B T$ and D is the total number of spatial dimensions in the system [13]. Eqn. (1.19) can be factorized into

$$f^0(\rho, \boldsymbol{u}, \boldsymbol{v}, \theta) = \frac{\rho}{(2\pi\theta)^{D/2}} e^{\frac{-\boldsymbol{v}\cdot\boldsymbol{v}}{2\theta}} e^{\frac{2\boldsymbol{u}\cdot\boldsymbol{v}-\boldsymbol{u}\cdot\boldsymbol{u}}{2\theta}}. \tag{1.20}$$

This expansion is valid for small Mach numbers, which are a dimensionless quantity representing the ratio of flow velocity to the speed of sound of a local medium c_s. The Mach number is written $Ma = \frac{\boldsymbol{u}}{c_s}$. For small Ma, a small macroscopic flow can be assumed. With this assumption, the factorized form of Eqn. (1.20) is can then be expanded around $\boldsymbol{u} = 0$. This gives

$$f^0(\rho, \boldsymbol{u}, \boldsymbol{v}, \theta) = \frac{\rho}{(2\pi\theta)^{D/2}} e^{\frac{-\boldsymbol{v}\cdot\boldsymbol{v}}{2\theta}} \left[1 + \frac{\boldsymbol{v}\cdot\boldsymbol{u}}{\theta} + \frac{(\boldsymbol{v}\cdot\boldsymbol{u})^2}{2\theta^2} - \frac{\boldsymbol{u}\cdot\boldsymbol{u}}{2\theta}\right] + O(u^3), \tag{1.21}$$

where $O(u^3)$ refers to all orders of u three and higher which are neglected due to all terms of higher order containing negligible contributions to the macroscopic behavior of the system. Further discussion of this notation will be discussed in Section 1.1.5.1. This expanded form is weighted by a normalized Boltzmann factor

$$w(\boldsymbol{v}) = \frac{e^{\frac{-\boldsymbol{v}\cdot\boldsymbol{v}}{2\theta}}}{(2\pi\theta)^{D/2}}. \tag{1.22}$$

This weighted Boltzmann factor is subject to the normalization condition

$$\int w(\boldsymbol{v}) \mathrm{d}\boldsymbol{v} = 1. \tag{1.23}$$

This condition can be confirmed by exploiting Gaussian integrals. Including the weighting factor, the equilibrium distribution is

$$f^0(\rho, \boldsymbol{u}, \boldsymbol{v}, \theta) = \rho w(\boldsymbol{v}) \left[1 + \frac{\boldsymbol{u}\cdot\boldsymbol{v}}{\theta} + \frac{(\boldsymbol{u}\cdot\boldsymbol{v})^2}{2\theta^2} - \frac{\boldsymbol{u}\cdot\boldsymbol{u}}{2\theta}\right] + O(u^3). \tag{1.24}$$

This expanded form of the Maxwell-Boltzmann distribution is still in the continuous form. In order to discretize the equilibrium distribution, the discrete velocity set $\{\boldsymbol{v}_i\}$ must be considered [2]. It is not enough to simply replace the continuous particle velocity v in Eqn. (1.24) with discrete velocities v_i. This does not reproduce the correct weighting factor. The weighting factor must also be discretized to form a set of weights $\{w_i\}$ which correspond with the chosen velocity set $\{\boldsymbol{v}_i\}$.

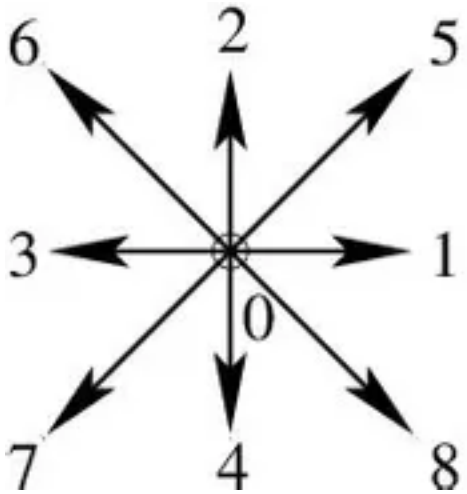


Figure 1.1. The graphical representation of the velocities which are contained in a D2Q9 model.

The weights $\{w_i\}$ are a normalized set of values which enforce the rotational isotropy of the lattice which $\{\boldsymbol{v}_i\}$ must obey. An isotropic lattice must be rotationally invariant in all directions. The correct set of weights $\{w_i\}$ will depend on the spatial dimensionality of the lattice and the choice of $\{\boldsymbol{v}_i\}$. In order to reproduce hydrodynamic behavior in two dimensions, the D2Q9 model seen in Fig. 1.1 is commonly utilized[17]. To enforce isotropy in this example, the moments of the weights are required up to fifth order [13, 2, 15]. These moments are

$$\sum_i w_i = 1 \tag{1.25}$$

$$\sum_i v_{i\alpha} w_i = 0 \tag{1.26}$$

$$\sum_i v_{i\alpha} v_{i\beta} w_i = \theta \delta_{\alpha\beta} \tag{1.27}$$

$$\sum_i v_{i\alpha} v_{i\beta} v_{i\gamma} w_i = 0 \tag{1.28}$$

$$\sum_i v_{i\alpha} v_{i\beta} v_{i\gamma} v_{i\delta} w_i = \theta^2 (\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) \tag{1.29}$$

$$\sum_i v_{i\alpha} v_{i\beta} v_{i\gamma} v_{i\delta} v_{i\nu} w_i = 0. \tag{1.30}$$

The weights of odd order in velocity are required to be zero to preserve the symmetry of the lattice. A further requirement states that all weights must be non-negative. The Einstein summation

notation has now been introduced. The Einstein notation utilizes the Greek indices to refer to spatial dimensions and repeated indices are explicitly summed over.

The conditions in Eqns. (1.25–1.30) are all required for a D2Q9 implementation of the Navier-Stokes equations. In D2Q9, the velocity set can be written

$$\{\boldsymbol{v}_i\} = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right\}. \tag{1.31}$$

Here, each vector will correspond with motion along the lattice. We can derive each weight in $\{w_i\}$ in terms of θ. We must consider each dimension individually to the use of the Einstein notation. In Eqn. (1.28), there are two constraints to consider, the first is the case in which all dimensions are the same such that

$$\sum_i v_{ix}^3 w_i = 0 \tag{1.32}$$

and the second is the case in which two dimensions are mixed in the form

$$\sum_i v_{ix}^2 v_{iy} w_i = 0. \tag{1.33}$$

These two constraints lead to the relations

$$w_1 = w_2, \quad w_3 = w_4, \quad , w_6 = w_7, \quad w_5 = w_8. \tag{1.34}$$

Now using Eqn. (1.27) for the case where there are two mixed dimensions, it follows that

$$\sum_i v_{ix} v_{iy} w_i = 0 \implies w_5 = w_6 = w_7 = w_8. \tag{1.35}$$

This means that each vector which moves diagonally on the lattice is equivalently weighted. Using Eqn. (1.29) has the constraint

$$\sum_i v_{ix}^2 v_{iy}^2 w_i = \theta^2. \tag{1.36}$$

Using the relations in Eqn. (1.35), it follows that

$$w_5 = w_6 = w_7 = w_8 = \frac{\theta^2}{4}. \tag{1.37}$$

Returning to Eqn. (1.27) and using these definitions, it can be shown that

$$w_1 = w_2 = w_3 = w_4 = \frac{\theta - \theta^2}{2} \tag{1.38}$$

which correspond to the vectors which move along the x or y directions alone. With these defined, Eqn. (1.25) can be examined and the final weight w_0 can be found which takes the form

$$w_0 = 1 - \theta(2 - \theta). \tag{1.39}$$

This weight corresponds to the rest vector.

In practice, the lattice temperature θ must be known. Examining Eqn. (1.29) reveals a case where all Greek indexed spatial coordinates would be of the same dimension. For example, when all indices correspond to the x-dimension, it would be expected that this moment would be equivalent to Eqn. (1.27). Using this fact, Eqn. (1.29) becomes

$$\sum_i v_{ix}^4 w_i = 3\theta^2. \tag{1.40}$$

Now since $v_{ix}^2 = v_{ix}^4$, it is clear that in order to recover Eqn. (1.27), it must be the case that $\theta = 1/3$. By this fact, this value for θ is required for an isotropic lattice.

With all the weights derived, the full set of weights $\{w_i\}$ can be constructed. For the D2Q9 velocity set, the set of weights is

$$\{w_i\} = \left\{1 - \theta(2 - \theta), \frac{\theta - \theta^2}{2}, \frac{\theta - \theta^2}{2}, \frac{\theta - \theta^2}{2}, \frac{\theta - \theta^2}{2}, \frac{\theta^2}{4}, \frac{\theta^2}{4}, \frac{\theta^2}{4}, \frac{\theta^2}{4}\right\} \tag{1.41}$$

For this D2Q9 implementation requiring $\theta = 1/3$ [2], the set of weights is

$$\{w_i\} = \left\{\frac{4}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}\right\}. \tag{1.42}$$

With a complete set of weights, the discrete equilibrium distribution function can be written as

$$f_i^0(\rho, \boldsymbol{u}, \boldsymbol{v}_i, \theta) = \rho w_i \left[1 + \frac{\boldsymbol{u} \cdot \boldsymbol{v}_i}{\theta} + \frac{(\boldsymbol{u} \cdot \boldsymbol{v}_i)^2}{2\theta^2} - \frac{\boldsymbol{u} \cdot \boldsymbol{u}}{2\theta}\right]. \tag{1.43}$$

The following section will utilize the local equilibrium distribution in connection with the lattice Boltzmann in Eqn. (1.18) to derive various equations of motion based on different choices of the equilibrium distribution. .

1.1.5. Deriving Equations of Motion from the Lattice Boltzmann Equation - The Hydrodynamic Limit

The macroscopic equations of motion of a system which is being modeled can be derived from the lattice Boltzmann equation in Eqn. (1.18). We expand the LBE up to the desired order to obtain a partial differential equation for the local equilibrium distribution f_i^0 which only depends on the conserved quantities.

1.1.5.1. Neglecting Higher Order Derivative Terms in the Hydrodynamic Limit

In the next section we introduce a Taylor expansion of the discrete Boltzmann equation to derive partial differential equations that are the governing evolution equations for fluid flow. An important facet of the hydrodynamic limit is that we neglect higher order derivatives from the Taylor expansion which have negligible contributions to the macroscopic behavior of the system. This runs contrary to the typical understanding of the nature of partial differential equations being dominated by their highest order. For hydrodynamic systems, the opposite of this is accurate in that the lowest orders of the partial differential equation tend to dominate the system.

A physical justification for this is that, in hydrodynamic systems, short wavelengths will equilibrate quickly, where long wavelengths take a much longer time to reach equilibrium. When examining a system in the hydrodynamic limit, the higher order derivatives in the expansion correspond to shorter wavelengths. With this in mind, the longer wavelengths (lower order derivatives) end up dominating the dynamics of the system. Since the lower order derivatives corresponding to the long wavelengths dominate, the higher order derivative terms become negligible and are considered to go to zero and thus vanish from the equation [18, 19]. This notion is explored further in section 3.4.

A common justification for neglecting higher order terms is related to how the derivatives in the expansion, $\Delta t(\partial_t + v_i\partial_\alpha)^n$, scale with the Knudsen number. The following derivation is given by Kruger *et al.* [2]. The Knudsen number is written

$$\mathrm{Kn} = \frac{l_{mfp}}{l} \tag{1.44}$$

where l_{mfp} is the mean free path and l is a macroscopic length scale. The Knudsen number can also be related similarly to time using the speed of sound of the material, c_s. c_s is on the same order of the average particle speed of the gas [20], so the average time between collisions is

$$\mathcal{T}_{mfp} = O\left(\frac{l_{mfp}}{c_s}\right). \tag{1.45}$$

An acoustic time scale can be defined as

$$\mathcal{T}_{c_s} = \frac{l}{c_s}, \tag{1.46}$$

which is the time it takes for an acoustic effect to be felt across the length scale l. These relations show that

$$\mathrm{Kn} = \frac{l_{mfp}}{l} = O\left(\frac{\mathcal{T}_{mfp}}{\mathcal{T}_{c_s}}\right). \tag{1.47}$$

Since the distribution functions are relaxed towards equilibrium through the collision process and it takes relatively few collisions to reach equilibrium, we know that the relaxation time τ is on the order of $\mathcal{T}_{mfp}$ such that

$$\tau = O(\mathcal{T}_{mfp}) \tag{1.48}$$

and

$$\Delta t = O(\tau) \tag{1.49}$$

through the discretization process of the method. Combining all of this together, we can examine the order of each derivative in the Knudsen number by

$$O(\Delta t\partial_t) \sim O\left(\frac{\tau}{\mathcal{T}_{c_s}}\right) \sim O\left(\frac{\mathcal{T}_{mfp}}{\mathcal{T}_{c_s}}\right) \sim O(\mathrm{Kn}) \tag{1.50}$$

$$O(\Delta t v_i \partial_\alpha) \sim O\left(\frac{\tau c_s}{l}\right) \sim O\left(\frac{\mathcal{T}_{mfp}}{\mathcal{T}_{c_s}}\right) \sim O(\mathrm{Kn}). \tag{1.51}$$

With this result, we have shown that the order of the derivatives from the Taylor expansion are of the same order of the Knudsen number such that

$$O(\partial^n) = O(\mathrm{Kn}^n), \tag{1.52}$$

thus satisfying the assumption that derivatives of third order and higher can be neglected, keeping only the two lowest orders of the Knudsen number.

1.1.5.2. Taylor Expansion of the Lattice Boltzmann Equation

The first step in deriving the equations of motion for a system is to take a Taylor expansion of the left hand side of Eqn. (1.18) to the second order [13] which gives

$$(\partial_t + v_{i\alpha}\partial_\alpha) f_i + \frac{1}{2}(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i + O(\partial^3) = \frac{1}{\tau}(f_i^0 - f_i). \tag{1.53}$$

In order to properly reproduce the equations of motion, it is required to write the distribution functions f_i in terms of the equilibrium distribution f_i^0. This can be done by solving Eqn. (1.53) for f_i giving

$$f_i = f_i^0 - \tau\left[(\partial_t + v_{i\alpha}\partial_\alpha) f_i + \frac{1}{2}(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i\right] + O(\partial^3). \tag{1.54}$$

In this form, there are still remaining f_i terms. To remedy this, one can substitute Eqn. (1.54) into itself until there are only f_i^0 terms remaining to the degree of choice. In this application, the desired degree is second order, so all third order and greater terms will be neglected.

$$\begin{aligned} f_i = f_i^0 - \tau\Bigg\{ & (\partial_t + v_{i\alpha}\partial_\alpha)\left[f_i^0 - \tau\left((\partial_t + v_{i\alpha}\partial_\alpha) f_i + \frac{1}{2}(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i\right)\right] \\ & + \frac{1}{2}(\partial_t + v_{i\alpha}\partial_\alpha)^2\left[f_i^0 - \tau\left((\partial_t + v_{i\alpha}\partial_\alpha) f_i + \frac{1}{2}(\partial_t + v_{i\alpha}\partial_\alpha)^2\right)\right]\Bigg\} + O(\partial^3). \end{aligned} \tag{1.55}$$

Any term which is operated upon by $(\partial_t + v_{i\alpha}\partial_\alpha)^3$ or of higher degree can be neglected and immediately falls into $O(\partial^3)$. This leaves

$$f_i = f_i^0 - \tau\left[(\partial_t + v_{i\alpha}\partial_\alpha)f_i^0 - \tau(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i + \frac{1}{2}(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i^0\right] + O(\partial^3). \tag{1.56}$$

At this point, there is still a term which still contains f_i rather than f_i^0, so the substitution of Eqn. (1.54) must be performed once again giving

$$\begin{aligned} f_i = f_i^0 - \tau\Bigg\{&(\partial_t + v_{i\alpha}\partial_\alpha)f_i^0 - \tau(\partial_t + v_{i\alpha}\partial_\alpha)^2\left[f_i^0 - \tau\left((\partial_t + v_{i\alpha}\partial_\alpha)f_i + \frac{1}{2}(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i\right)\right] \\ &+ \frac{1}{2}(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i^0\Bigg\} + O(\partial^3) \end{aligned}$$

$$\begin{aligned} f_i =& f_i^0 - \tau\left[(\partial_t + v_{i\alpha}\partial_\alpha)f_i^0 - \tau(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i^0 + \frac{1}{2}(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i^0\right] + O(\partial^3) \\ =& f_i^0 - \tau(\partial_t + v_{i\alpha}\partial_\alpha) + \left(\tau^2 - \frac{1}{2}\right)(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i^0 + O(\partial^3). \end{aligned} \tag{1.57}$$

This is now a second order representation of f_i in terms of f_i^0. Eqn. (1.57) can be inserted into Eqn. (1.53) leading to

$$\begin{aligned} &(\partial_t + v_{i\alpha}\partial_\alpha)\left[f_i^0 - \tau(\partial_t + v_{i\alpha}\partial_\alpha)f_i^0 + \left(\tau^2 - \frac{1}{2}\right)(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i^0\right] \\ &+ \frac{1}{2}(\partial_t + v_{i\alpha}\partial_\alpha)^2\left[f_i^0 - \tau(\partial_t + v_{i\alpha}\partial_\alpha)f_i^0 + \left(\tau^2 - \frac{1}{2}\right)(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i^0\right] + O(\partial^3) = \frac{1}{\tau}(f_i^0 - f_i). \end{aligned} \tag{1.58}$$

Combining all the derivatives and neglecting all terms of $O(\partial^3)$, we then arrive at

$$(\partial_t + v_{i\alpha}\partial_\alpha)f_i^0 - \left(\tau - \frac{1}{2}\right)(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i^0 + O(\partial^3) = \frac{1}{\tau}(f_i^0 - f_i), \tag{1.59}$$

which is a second order partial differential equation for the local equilibrium distribution.

Traditionally, this expansion is performed due to a perturbative method known as the Chapman-Enskog expansion [1, 5, 2, 14, 15]. Similar to this derivation, Chapman-Enskog can

derive the hydrodynamic equations from the Boltzmann equation. The Taylor expansion method presented here and the Chapman-Enskog expansion have equivalent results.

1.1.5.3. Mass Conservation

The vast majority of LBM implementations are designed to conserve both mass and momentum. However, in our case, we are interested in highlighting the fundamental behaviors of fluctuations. This is most easily done using a diffusive system where mass is the only conserved quantity. Since hydrodynamic systems, which conserve both mass and momentum, often contain difficulties related to Galilean invariance related to the macroscopic velocity $\boldsymbol{u}$, we consider only diffusive systems where Galilean invariance issues are not present. A full derivation of the Navier-Stokes equations from the LBE is presented in the Appendix.

For a system which only conserves mass, the macroscopic velocity $\boldsymbol{u}$ and the lattice temperature θ are fixed constants. We choose a local equilibrium distribution from Eqn. (1.43) of

$$f_i^0 = \rho w_i, \tag{1.60}$$

which is a simplified version of Eqn. (1.43) where the macroscopic velocity $\boldsymbol{u} = 0$.

The macroscopic moments calculated from the distribution functions in Eqns. (1.1–1.2) can also be extended to the local equilibrium distributions. These moments of the equilibrium distributions can be calculated as finite sums in the same manner. If the only imposed conserved quantity is mass, the given moments for Eqn. (1.60) are written

$$\sum_i f_i^0 = \rho \tag{1.61}$$

$$\sum_i v_{i\alpha} f_i^0 = 0 \tag{1.62}$$

$$\sum_i v_{i\alpha} v_{i\beta} f_i^0 = \rho\theta\delta_{\alpha\beta}. \tag{1.63}$$

It must be noted that these moments for Eqn. (1.60) will not be the same as the moments for Eqn. (1.43).

With these moments of f_i^0, it is possible to sum over all indices i in Eqn. (1.59) by

$$\sum_i (\partial_t + v_{i\alpha}\partial_\alpha) f_i^0 - \sum_i \left(\tau - \frac{1}{2}\right)(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i^0 + O(\partial^3) = \sum_i \frac{1}{\tau}(f_i^0 - f_i). \tag{1.64}$$

By doing this and using the moments of Eqns. (1.61–1.63), it follows that

$$\partial_t \rho \left(\tau - \frac{1}{2}\right)(\partial_t^2 \rho + \rho\theta\delta_{\alpha\beta}) + O(\partial^3) = 0. \tag{1.65}$$

This equation has a similar form to the diffusion equation. The diffusion equation states

$$\partial_t \rho(\boldsymbol{x}, t) = D\partial_\alpha \partial_\beta \rho(\boldsymbol{x}, t)\delta_{\alpha\beta} \tag{1.66}$$

where D is the diffusion coefficient of the medium. This case assumes that D is constant for the total system. Eqn. (1.65) has an extraneous second order temporal derivative which is not in Eqn. (1.66). This can be rectified by taking an additional derivative of the diffusion equation. This leads to

$$\partial_t^2 \rho = D^2 \partial_\alpha \partial_\beta \partial_\gamma \partial_\delta \rho(\boldsymbol{x}, t)(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\gamma} + \delta_{\alpha\delta}\delta_{\beta\gamma}). \tag{1.67}$$

Inserting this into Eqn. (1.65) shows that the second order time derivative is equivalent to a fourth order spatial derivative. By doing this, the ∂_t^2 term is put into $O(\partial^3)$. With this substitution, the diffusion equation has been derived in the form

$$\partial_t \rho = \left(\tau - \frac{1}{2}\right)\partial_\alpha \partial_\beta \rho\theta\delta_{\alpha\beta}. \tag{1.68}$$

In the isothermal case where θ is constant, the diffusion coefficient is written

$$D = \left(\tau - \frac{1}{2}\right)\theta. \tag{1.69}$$

This diffusive lattice Boltzmann was first derived by Wolf-Gladow in 1995 [9]. The diffusive lattice Boltzmann is a simple implementation of the lattice Boltzmann algorithm since it only conserves mass.

It bears noting that the Taylor expansion method (which is perturbative by nature) is not the only way to derive the diffusion equation. Chopard *et al.* presented a non-perturbative derivation of the diffusion equation for a cellular automaton for a simple D1Q2 model [15]. This derivation recovers additional terms that are only relevant at long wavelengths.

To model hydrodynamic systems, in addition to mass conservation, momentum conservation is also required. Since this manuscript is focused on diffusive systems in which only mass conservation is needed, the derivation for the equations of motion for hydrodynamic systems is relegated to Appendix 1.

1.2. Fluctuating Lattice Boltzmann

A property of the general LBM is that, unlike the discrete particle behavior of LGCA, fluctuations are not inherently included. In LGCA methods, the discrete particle behavior gives rise to particle noise which can be thought of as fluctuations. An issue that arises from this discrete particle noise is that the noise is uncontrollable for specific particle densities. LBM resolves this issue by implementing the distribution functions. The f_i in LBM can be thought of as ensemble averages of the states of the discrete particles. Due to this fact, LBM is devoid of fluctuations. To model systems where fluctuations play an important role, these fluctuations must be reintroduced into the LBM. There is an active sub-field of lattice Boltzmann research which studies the reintroduction of fluctuations in to the method. These revised methods are known as fluctuating lattice Boltzmann methods (FLBM) [8, 21, 22].

As discussed previously, the general LBE in Eqn. (1.14) only includes collisions and streaming of the distribution functions at each time step. At its core, there is nothing which would contribute fluctuations to the method. To remedy this, the collision operator can be modified to include a fluctuating term ξ_i which has moments

$$\sum_i \xi_i = 0 \tag{1.70}$$

In moment space, these noise terms are defined in terms of a correlation function $< \xi^a \xi^b >$. This correlation function can be written terms of densities of the system and is written as

$$< \xi^a \xi^b >= \rho \frac{2\tau^a - 1}{(\tau^a)^2}(1 - \delta^{a0}\delta^{b0})\delta^{ab}, \tag{1.71}$$

which is a fully diagonal matrix representing uncorrelated noise terms. A full derivation of this correlation function will be presented in chapter 2. Eqn. (1.71) is the fluctuation dissipation theorem for a diffusive FLBM.

The MRT collision operator is desired since it allows for each degree of freedom to be relaxed by its own independent relaxation time τ_i. The fluctuating MRT collision operator takes the form

$$\Omega_i(\{f_j\}) = \sum_j \Lambda_{ij}(f_j^0 - f_j) + \xi_i \tag{1.72}$$

where Λ_{ij} is the collision matrix which contains the independent relaxation time parameters. This revision gives a new form of the LBE which is known as the fluctuating lattice Boltzmann equation:

$$f_i(\boldsymbol{x} + \boldsymbol{v}_i, t+1) = f_i(\boldsymbol{x}, t) + \sum_j \Lambda_{ij}(f_j^0 - f_j) + \xi_i. \tag{1.73}$$

With proper forms of Λ_{ij} and ξ_i, fluctuations will be reintroduced into the LBM.

1.3. Lattice Gas Methods

Although lattice Boltzmann methods have become of higher interest and practical use than lattice gas methods, there are potential benefits to utilizing a lattice gas. The discrete nature of lattice gas is that it is intrinsically noisy, which automatically includes fluctuations in the method [6]. In the case of lattice Boltzmann, fluctuations are absent and must be included in the collision operator *ad hoc*. The process of including fluctuations into a lattice Boltzmann method will be discussed in chapter 2. In recent years, there has been renewed interest in research into lattice gas methods which was initiated by Blommel *et al.* by successfully constructing a functional integer lattice gas [17]. Traditionally, lattice gas methods were implemented where single particles existed on a lattice as Boolean values in which only one particle could exist on any lattice point at any given time. The fact that only a single particle could exist at a node was a limiting factor in the method which could be resolved by extending the Boolean particles to particle distribution functions seen in lattice Boltzmann methods. Blommel's successful implementation served as proof that it was, in fact, possible to construct lattice gases which allow for any integer number of particles to exist on a lattice node at any given time.

With a new model for integer lattice gas methods, research has continued to improve computational efficiency. Seekins *et al.* were able to present a sampling collision operator which was found to be equivalent to the BGK collision operator commonly utilized in lattice Boltzmann [23]. This was a major step forward in improving the likelihood for practical implementations of the integer lattice gas method.

The following sections will present the traditional Boolean lattice gas methods and how these methods were extended into integer lattice Boltzmann methods. Then, integer lattice gas methods are introduced, followed by a derivation of the sampling collision operator which is utilized in Sec. 1.2 to achieve overrelaxation in integer lattice gas methods.

1.3.1. The Boolean Lattice Gas Method

The first lattice gas method was developed in 1973 by Hardy *et al.* on a square lattice [24]. This method had the ability to model simple gas dynamics in a two dimensional system. This model was built on a square lattice and allowed a individual particles to exist at any lattice node at a given time. The particles were allowed to propagate along the lattice links and collide with other particles in a manner that conserves mass and momentum. The square lattice for this simple model utilized what would be referred today as a D2Q4 implementation, where each particle could have one of four possible velocities which propagate the particles along the lattice links, but are not allowed to stay at rest.

However, this method was unable to effectively model the Navier-Stokes equation to reproduce fluid flows. This inability to model the Navier-Stokes equation is due to an inherent lack of symmetry in the stress tensor due to the fact that no x direction momentum can be transferred to y direction momentum and vice versa. It was not until Frisch *et al.* [3] implemented a similar model on a hexagonal lattice that fluids could be represented by these simple lattice gas models. Instead of the four velocities utilized on the square lattice, the hexagonal lattice consisted of six velocities which can propagate the particles to any of the neighboring lattice points. The implementation of the hexagonal lattice provided sufficient isotropy on the lattice to effectively simulate fluid flow [5, 1, 14]. An examples of the Boolean lattice gas velocity set and hexagonal lattice are seen in Fig. 1.2

The construction of lattice gases are similar to the lattice Boltzmann models presented previously. A major distinction of lattice gas methods from lattice Boltzmann methods is that,

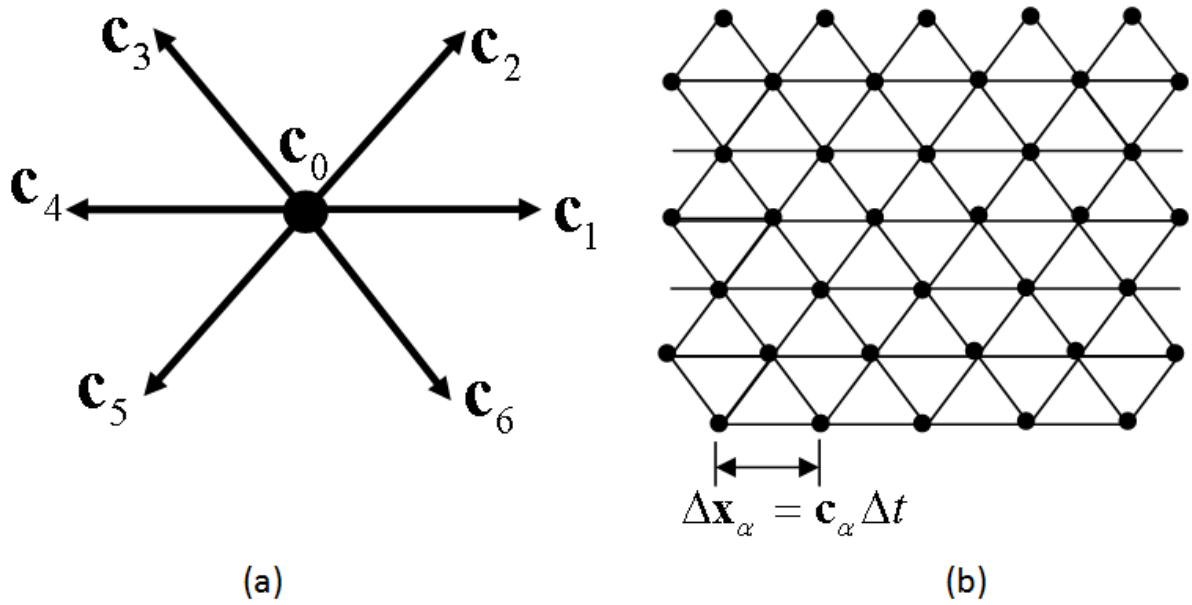


Figure 1.2. (a) Velocity vectors for a Boolean lattice gas on a heaxagonal lattice. (b) The hexagonal lattice structure used in Boolean lattice gas methods.

instead of utilizing continuous distribution functions, $f_i(\boldsymbol{x}, t)$, to represent particle configurations, lattice gas methods use discrete occupation numbers $n_i(\boldsymbol{x}, t)$ that exist on a specific lattice point. These early lattice gas models implemented Boolean occupation numbers which allowed for only one particle of a certain velocity to exist at any given node on the lattice. The macroscopic quantities of mass and momentum can be defined similarly to Eqns. (A.1-A.2) and take the form

$$N(\boldsymbol{x}, t) = \sum_i n_i(\boldsymbol{x}, t) \tag{1.74}$$

$$N(\boldsymbol{x}, t)\boldsymbol{u}(\boldsymbol{x}, t) = \sum_i \boldsymbol{v}_i f_i(\boldsymbol{x}, t), \tag{1.75}$$

where $N(\boldsymbol{x}, t)$ is the total number of particles at a given lattice node.

The Boolean lattice gas algorithm follows collision-streaming rules to govern the time-evolution of the system. The Boolean lattice gas (BLG) collision operator is discrete and probabilistic by nature. The collision step implements a collision operator $\Omega_i^{LG}(\boldsymbol{x}, t)$ which must conserve

mass and momentum

$$\sum_i \Omega_i^{LG} = 0 \tag{1.76}$$

$$\sum_i v_i \Omega_i^{LG} = 0. \tag{1.77}$$

This collision operator will redistribute the occupation numbers. This redistribution process can be represented mathematically as

$$n_i^*(\boldsymbol{x}, t) = n_i(\boldsymbol{x}, t) + \Omega_i^{LG}(\boldsymbol{x}, t) \tag{1.78}$$

where n_i^* is the post-collision occupation and the collision operator is $\Omega_i^{LG} \in \{-1, 0, 1\}$. While the collision rules implemented by the collision operator can vary widely [15], a common set of collision rules is represented in Fig. 1.3 where incoming particles to a specific lattice point are redistributed along neighboring lattice links through the collision process [25]. BLG collision rules allow for multi-body collisions to occur. Fig. 1.3 shows examples of two, three, and four particle collisions. The particles in BLG are subject to an exclusion principle which only allows for a single particle of a velocity to be present at a lattice node at a given time. Thus the equilibrium distribution is of the Fermi-Dirac variety as opposed to a Boltzmann distribution [26].

The streaming step for Boolean lattices gases works in the same manner as in lattice Boltzmann such that the post-collision particles, n_i^*, will move exactly one lattice space over a single time step, Δt, based on the velocity of the particle

$$n_i(\boldsymbol{x} + \boldsymbol{v_i}\Delta t, t + \Delta t) = n_i^*(\boldsymbol{x}, t). \tag{1.79}$$

The collision and streaming steps can be combined to define a general equation of motion for the Boolean gas method taking the form

$$n_i(\boldsymbol{x} + \boldsymbol{v_i}\Delta t, t + \Delta t) = n_i(\boldsymbol{x}, t) + \Omega_i^{LG}(\boldsymbol{x}, t). \tag{1.80}$$

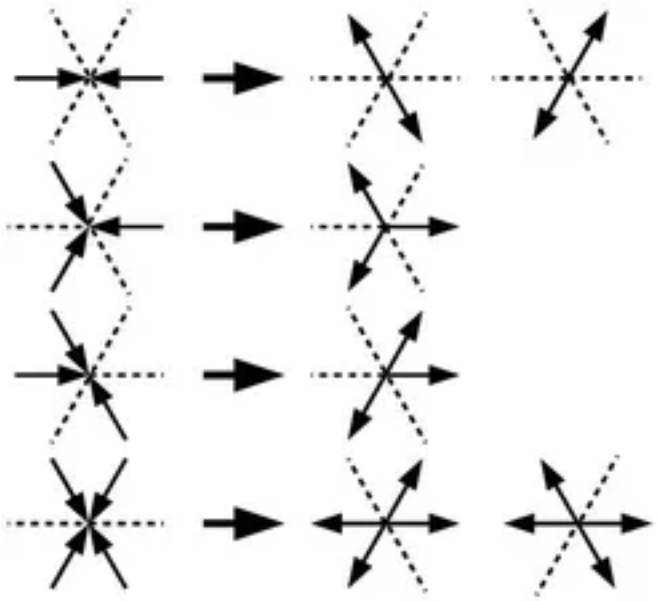


Figure 1.3. Illustration of common Boolean lattice gas collision rules in which incoming particles are redistributed to neighboring lattice nodes. These Boolean collision rules conserve both mass and momentum. Examples of two, three, and four particle collisions are shown.

It is important to note that the structure of Eqn. (1.80) is identical to that of Eqn. (1.3) except for the fact that the Boolean lattice gas utilizes the discrete Boolean occupation numbers as opposed to the continuous distribution functions.

1.3.2. Deriving Lattice Boltzmann Methods from Boolean Lattice Gases

The original lattice Boltzmann methods were derived from Boolean lattice gases as a Boltzmann average by examining non-equilibrium ensemble averages of the discrete particle occupation numbers. The particle distribution functions can be defined as

$$f_i(\boldsymbol{x}, t) = \langle n_i(\boldsymbol{x}, t) \rangle \tag{1.81}$$

where $\langle\rangle$ refer to a non-equilibrium ensemble average. This ensemble average connects the microscopic realizations of the occupation numbers to the continuous macroscopic quantities within a system. Thus taking a non-equilibrium average of the total number of particles at a lattice point, we acquire the particle density such that

$$\rho(\boldsymbol{x}, t) = \langle N(\boldsymbol{x}, t) \rangle. \tag{1.82}$$

The same non-equilibrium averaging can be performed on the collision operator

$$\Omega_i(\{f_i\}) = \langle \Omega_i^{LG}(\boldsymbol{x}, t) \rangle. \tag{1.83}$$

This averaged lattice gas collision operator is equivalent to the lattice Boltzmann collision operator Ω_i in Eqn. (1.3). We can then take the ensemble average of Eqn. (1.80) such that

$$\langle n_i(\boldsymbol{x} + v_i \Delta t, t + \Delta t) \rangle = \langle n_i(\boldsymbol{x}, t) \rangle + \langle \Omega_i^{LG}(\boldsymbol{x}, t) \rangle, \tag{1.84}$$

which obtains the following evolution equation in terms of the particle distribution functions

$$f_i(\boldsymbol{x} + v_i \Delta t, t + \Delta t) = f_i(\boldsymbol{x}, t) + \Omega_i(\{f_i\}). \tag{1.85}$$

This equation is equivalent to the lattice Boltzmann equation shown in Eqn. (1.3).

A significant drawback to the BLG algorithm is that Galilean invariance is violated in the hydrodynamic equations. This broken Galilean invariance is due to high order corrections to the Navier-Stokes equation. The corrections are related to the discrete nature of the lattice and velocity sets and the Galilean invariance violations arise when the macroscopic fluid velocity is no longer negligible when compared to the microscopic particle velocities [27, 28].

The following section will discuss a recent implementation of a lattice gas which allows for multiple particles of a given velocity to exist on a lattice node at any time. This integer lattice gas method brings with it renewed interest in discrete lattice gas methods which inherently include fluctuations.

1.3.3. Integer Lattice Gases

An integer lattice gas extends the traditional Boolean lattice gas method to include an integer number of particles on a certain lattice point instead of only allowing for a single particle. In other words, instead of Boolean occupation numbers as utilized in the Boolean lattice gas, the integer lattice gas method incorporates integer valued occupation numbers. The multi-particle integer lattice gas was a focus of interest in the late 1990s and there were developments relating to implementing these methods [29, 30, 31, 32, 33]. Integer lattice gases were introduced by Boghosian and Masselot [34, 35]. This article studied the thermodynamic behavior of the integer lattice gases

focusing on energy conservation. This paper also introduces a sampling method for collisions which only incorporates mass and momentum conservation. Section 1.3.3.2 presents a successful sampling collision operator for a diffusive integer lattice gas.

In 1998, Chopard *et al.* introduced another version of the integer lattice gas which utilized a collision which operates similarly to a BGK collision operator, which is commonly implemented in lattice Boltzmann methods [36, 35]. This method resulted in a collision process that employed a continuous probability distribution which was then sampled to acquire a new discrete distribution. This process violated momentum conservation. To remedy this momentum conservation violation, a random walk of the density is introduced which would proceed until the proper value of momentum is found.

Although these early integer lattice gas methods showed promise, the rise of lattice Boltzmann methods halted a large amount of the research into integer lattice gases. Since Blommel *et al.* introduced an integer lattice gas with a Monte Carlo collision operator in 2018 [17], there has been renewed interest in the practical applications for integer lattice gas methods. This revitalized method was the inspiration to explore overrelaxation in integer lattice gases in Sec. 4.3.

The idea behind Blommel's Monte Carlo integer lattice gas is that it implements a Monte Carlo based collision operator which conserves mass and momentum and explicitly recovers the equilibrium distribution for a macroscopic velocity $\boldsymbol{u} = 0$. A major difference from Boolean lattice gas methods, is that this integer version can be constructed on a square lattice, as is common for lattice Boltzmann methods. The collision-streaming process is still performed in a similar manner as other lattice gas methods. In this case, $n_i(\boldsymbol{x}, t) \in \mathbb{Z}$. Using integer-valued occupation numbers removes the exclusion principle observed in Boolean lattice gases, thus leading to an equilibrium distribution equivalent to a Boltzmann distribution where $\boldsymbol{u} = 0$ rather than the Fermi-Dirac distribution observed in Boolean lattice gases.

The integer-valued occupation numbers, $n_i(\boldsymbol{x}, t)$, work similarly to the distribution functions in lattice Boltzmann, however, these distribution functions are continuous such that $f_i(\boldsymbol{x}, t) \in \mathbb{R}$. The collision will redistribute these integer valued occupation numbers. The collision operator in this case is represented by Ξ_i which leads to an evolution equation for the model in the form

$$n_i(\boldsymbol{x} + v_i \Delta t, t + \Delta t) = n_i(\boldsymbol{x}, t) + \Xi_i(\{n_i\}). \tag{1.86}$$

The main differences between this version from Eqn. (1.80) is the fact that the occupation numbers are integers rather than Boolean valued and the construction of the collision operator, Ξ_i. The following section will develop the integer lattice gas (ILG) collision operator as presented by Blommel *et al.* [17].

1.3.3.1. Monte Carlo Collision Operator for Integer Lattice Gases

The collision operator for integer lattice gas, Ξ_i, differs from the lattice Boltzmann collision operator, Ω_i, by the fact that Ξ_i is probabilistic where Ω_i is deterministic. The ILG collision operator utilizes a collection of binary (two-particle) collisions, removing the need for multi-particle collisions as implemented in the collision rules for Boolean lattice gases.

In order to define the collision operator, Ξ_i, it is demanded that an equilibrium distribution equivalent to the lattice Boltzmann equilibrium distribution in Eqn. (1.43) given by

$$f_i^{eq}(\rho, \boldsymbol{u} = 0) = \rho w_i, \tag{1.87}$$

must be implemented. Since lattice gases inherently include fluctuations, it is imperative to discern the global equilibrium, f_i^{eq}, and the local equilibrium distribution f_i^0.

The probability that a collision between two particles with velocities v_i and v_j and resulting in velocities v_k and v_l can be written as $P_{ij\to kl}$. Detailed balance is assumed, which means that forward and backward collisions multiplied by the probabilities of these particles being in equilibrium must be equal. The probability of choosing a particle with velocity v_i at a given lattice node is w_i, which are the weights defined by the lattice and velocity set. The detailed balance condition can then be written as

$$w_i w_j P_{ij\to kl} = w_k w_l P_{kl\to ij}. \tag{1.88}$$

This gives the ratios of the forward and backward collisions in the form

$$\frac{P_{ij\to kl}}{P_{kl\to ij}} = \frac{w_k w_l}{w_i w_j}. \tag{1.89}$$

This allows for transitions where $ij = kl$, however, for real collisions where $ij \neq kl$, the transition probability is chosen as

$$P_{ij\to kl} \propto \min\left(1, \frac{w_k w_l}{w_i w_j}\right) \delta_{(v_i+v_j),(v_k+v_l)}, \tag{1.90}$$

where δ is the usual Kronecker delta. The transition probabilities must also add up to one

$$\sum_{kl} P_{ij\to kl} = 1 \tag{1.91}$$

with $P_{ij\to kl} \geq 0$. This probability can be achieved by introducing a proportionality factor, $\lambda_{ij,kl}$. This factor scales the rate at which certain collisions will occur. The factor is a constant value which is input prior to beginning a simulation. Putting this together gives the full transition probability for collisions in the form

$$P_{ij\to kl} = \begin{cases} \lambda_{ij,kl} \min\left(1, \frac{w_k w_l}{w_i w_j}\right) \delta_{(v_i+v_j),(v_k+v_l)} & ij \neq kl \\ 1 - \sum_{k'l' \neq ij} P_{ij\to k'l'} & ij = kl. \end{cases} \tag{1.92}$$

We must require $\lambda_{ij,kl} = \lambda_{kl,ij}$ to guarantee that Eqn. (1.89) is satisfied. The Kronecker delta function enforces mass and momentum conservation by ensuring that only collisions which conserve mass and momentum have a non-zero probability.

As is the same for Boolean lattice gases, the total number of particles at a given lattice node is given by

$$N(\boldsymbol{x}, t) = \sum_i n_i(\boldsymbol{x}, t). \tag{1.93}$$

To perform the collision, at each lattice point, a pair of particles are chosen to collide by selecting two evenly distributed random numbers between 1 and N. The two random numbers, r_1 and r_2, are then mapped to the velocities of the chosen particles determined by

$$\begin{cases} r_1 \to s_1 \text{ for } \sum_{i=0}^{s_1-1} n_i < r_1 \leq \sum_{i=0}^{s_1} n_i \\ r_2 \to s_2 \text{ for } \sum_{i=0}^{s_2-1} n_i < r_2 \leq \sum_{i=0}^{s_2} n_i. \end{cases} \tag{1.94}$$

The two numbers, s_1 and s_2, imply picking two particles with velocities v_{s_1} and v_{s_2}.

Next, the result of the collision must be determined by picking a pair of velocities, v_{s_3} and v_{s_4} with a probability of $P_{s_1 s_2 \to s_3 s_4}$. The transition is completed by by picking an additional

random number γ, where $0 < \gamma < 1$, and choosing

$$s_1 s_2 \to s_3(k) s_4(k) \text{ for } \sum_{i=0}^{k} P_{s_1 s_2 \to s_3(i) s_4(i)} < \gamma \leq \sum_{i=0}^{k+1} P_{s_1 s_2 \to s_3(i) s_4(i)}, \tag{1.95}$$

where a notation is introduced which will number pairs of velocities with an index value such that

$$\begin{aligned} s_3(i) &= i \mod V \\ s_4(i) &= [i - (i \mod V)]/V. \end{aligned} \tag{1.96}$$

The total collision step is a collection of multiple binary collisions of this type. The total number of collisions in each step is denoted by C and the index c is utilized to denote each individual binary collision. The result of any of these binary collisions are given by a random variable $\vartheta_i^c(s_1, s_2, s_3, s_4)$ and takes the form

$$\vartheta_i^c(s_1, s_2, s_3, s_4) = \delta_{i,s_3} + \delta_{i,s_4} - \delta_{i,s_1} - \delta_{i,s_2}. \tag{1.97}$$

This corresponds to the change in the total number of particles which have a velocity v_i after the collision has occurred. This process is repeated for all C collisions and we can define a full collision operator as a sum of these random variables such that

$$\Xi_i = \sum_{c=1}^{C} \vartheta_i^c(s_1, s_2, s_3, s_4). \tag{1.98}$$

The occupation numbers are updated after each individual binary collision so that ϑ_i^c results from a collision given by previous binary collisions in Eqn. (1.94).

It is interesting to note that for a D2Q9 implementation, this Monte Carlo ILG algorithm leads to a global equilibrium distribution in the form

$$f_i^{eq}(\rho, \boldsymbol{u}) = \rho \prod_{\alpha} w_{v_{i\alpha}} [1 + 3 v_{i\alpha} u_\alpha + (3 v_{i\alpha}^2 - 1)(\sqrt{1 + 3 u_\alpha^2} - 1)]. \tag{1.99}$$

This global equilibrium distribution is equivalent to an equilibrium distribution of an entropic lattice Boltzmann method [37, 38].

Although this algorithm is computationally expensive, it provides the framework for an ILG which conserves both mass and momentum. An additional drawback for ILG (and lattice gas in general) is that the the range of usable transport coefficients is small compared to LBM. A common technique in fluid dynamics to achieve low viscosities and low transport coefficients is overrelaxation. Overrelaxation is a method of relaxation in which instead of relaxing directly to the local equilibrium, the overrelaxed system will overshoot equilibrium and thus converge to equilibrium in an oscillatory manner. This is routine used in LBM due to the deterministic nature of the collision operator. However, since lattice gas collisions are probabilistic, overrelaxation was thought to be impossible due to the requirement that overrelaxed collisions would be required to occur with a probability greater than 1.

The subsequent section will introduce a sampling collision operator for diffusive systems which drastically increases computational efficiency of the ILG algorithm and serves as inspiration for investigation into the legitimacy of implementing overrelaxation in ILG.

1.3.3.2. Sampling Collision Operator for Integer Lattice Gases

The Monte Carlo integer lattice gas presented by Blommel serves as a template for successful implementation of ILG methods. However, the algorithm presented is notably inefficient, thus the range of practical applications for such an algorithm is minimal. Boghosian *et al.* [34] originally proposed a method to sample a probability distribution to perform collision in an ILG, but this sampling collision operator was never fully developed. Seekins *et al.* [23] were able to construct a sampling collision operator for a simplified diffusive system based on Blommel's Monte Carlo ILG. The sampling collision operator presented by Seekins offers a remedy to the computational efficiency issues which are present in Blommel's method. The sampling collision operator performs competitively with the corresponding fluctuating lattice Boltzmann method [39].

An integer lattice gas with a sampling collision operator will utilize the same evolution equation utilized in the Monte Carlo ILG

$$n_i(\boldsymbol{x} + v_i\Delta t, t + \Delta t) = n_i(\boldsymbol{x}, t) + \Xi_i(\{n_i\}) \tag{1.100}$$

where n_i are integer valued occupation numbers and Ξ_i is a stochastic operator that obeys all local conservation laws. In the diffusive case, the collision operator will conserve the local number of

particles, $N(\boldsymbol{x}, t)$ at a given lattice node, where

$$N(\boldsymbol{x}, t) = \sum_i n_i(\boldsymbol{x}, t). \tag{1.101}$$

In Blommel's ILG method, the collision operator was designed to conserve mass and momentum to simulate hydrodynamics. This was achieved by combining the net effect of many binary collisions. Since Seekins considered a diffusive case, only mass conservation is required. This leads to a significant simplification in developing the sampling collision operator by only needing to consider single particle collisions.

Similarly to the derivation for the Monte Carlo ILG, detailed balance is required in equilibrium seen in Eqn. (1.88). Since only unary collisions need to be considered for the diffusive case, the detailed balance condition simplifies to the form

$$N w_i P_{i\to j} = N w_j P_{j\to i} \tag{1.102}$$

and this gives the ratio of forward and backward collisions as

$$\frac{P_{i\to j}}{P_{j\to i}} = \frac{w_j}{w_i}. \tag{1.103}$$

This condition leads to transition probabilities

$$P_{i\to j} = \begin{cases} \lambda_{ij} \min\left(1, \frac{w_j}{w_i}\right) & i \neq j \\ 1 - \sum_{i,j(i\neq j)} P_{i\to j} & i = j \end{cases} \tag{1.104}$$

where Eqn. (1.102) implies $\lambda_{ij} = \lambda_{ji}$. For the highest level of efficiency for the algorithm, λ_{ij} should be as large as possible which will give the best acceptance rate for each collision. For maximal efficiency, we can choose

$$\lambda_{ij} = \max(w_i, w_j), \tag{1.105}$$

which leads to

$$P_{i\to j} = w_j. \tag{1.106}$$

This satisfies the detailed balance condition presented in Eqn. (1.102). This equates to randomly choosing a particle with velocity v_s and reassigning it with velocity v_t with probability w_t. This collision can be represented as a random variable in the form

$$\vartheta_i^c = \delta_{t(c),i} - \delta_{s(c),i}, \tag{1.107}$$

where c is the collision index and C is the total number of collisions. The collision operator can be defined as the sum of these individual collisions

$$\Xi_i^C = \sum_{c=1}^{C} \vartheta_i^c. \tag{1.108}$$

In principle, this completely defines an ILG for the fluctuating diffusion equation analogous to the hydrodynamic case presented by Blommel. Since the computational efficiency of these algorithms scale with the number of collisions, the idea is to replace the collision operator in Eqn. (1.108) with a collision operator which samples the post collision distribution from an appropriate distribution which can then perform all individual collisions in a single step.

When considering many collisions, a unique equilibrium probability for the set of occupation numbers, $\{n_i\}$, can be found. Since Eqn. (1.101) gives the total number of particles at a given lattice node, the mean of the local equilibrium distributions is Nw_i. The probability for a set of occupation numbers to exist is given by the multinomial distribution

$$P(\{n_i\}) = \begin{cases} N! \prod_{i=1}^{Q} \frac{w_i^{n_i}}{n_i!} & \text{if } \sum_i n_i = N \\ 0 & \text{otherwise.} \end{cases} \tag{1.109}$$

This is equivalent to the combinatorial problem of cumulative occurrences of N trials with Q outcomes with probability w_i.

First, we consider the collision operator Ξ_i^C in the limit of $C \to \infty$, which samples directly out of the multinomial distribution. Using a software package such as GSL [40], a set of Q random numbers can be sampled with the probability from Eqn. (1.109). This redistribution of N particles

onto Q bins can be then be sampled with probability w_i using

$$\begin{pmatrix} \hat{n}_0 \\ \vdots \\ \hat{n}_Q \end{pmatrix} = \begin{pmatrix} (X^N_{w_0,\ldots,w_Q})_0 \\ \vdots \\ (X^N_{w_0,\ldots,w_Q})_Q \end{pmatrix}, \tag{1.110}$$

where $(X^N_{w_0,\ldots,w_Q})_i$ is the ith component of the multinomial sample and $\hat{n}_i$ is the redistributed occupation number. This ensures that the collision conserves the total number of particles such that

$$\sum_i \hat{n}_i = N. \tag{1.111}$$

The collision operator, Ξ_i, can also be defined using a binomial sampling algorithm by letting X^N_p be a binomially distributed random number with probability

$$P(X^N_p = n) = \binom{N}{n} p^n (1-p)^{N-n}. \tag{1.112}$$

Since there are N particles available, an occupation number associated to velocity v_0 can be chosen such that each particle has a probability w_0 to be assigned to n_0. This is written as

$$\hat{n}_0 = X^N_{w_0}. \tag{1.113}$$

This leaves

$$\tilde{N}_1 = N - n_0 \tag{1.114}$$

particles available. Then n_1 particles from the remaining $\tilde{N}_1$ particles are chosen. Since there are fewer particles available, the probability to assign a particle to n_1 is increased to

$$\tilde{w}_1 = \frac{w_1}{1 - w_0}. \tag{1.115}$$

With this increased probability, we choose

$$\hat{n_1} = X^{\tilde{N}_1}_{\tilde{w}_1}. \tag{1.116}$$

This process is continued for the remaining occupation numbers and can be defined as

$$\tilde{N}_i = \tilde{N}_{i-1} - n_{i-1}, \tag{1.117}$$

with a normalized probability of

$$\tilde{w}_i = \frac{w_i}{1 - \sum_{i=1}^{i-1} w_i}. \tag{1.118}$$

The left-over occupation numbers are then sampled as

$$\hat{n}_i = X_{w_i}^{\tilde{N}_{i-1}}. \tag{1.119}$$

Since $\sum_i w_i = 1$, the probability for the final weight becomes $\tilde{w}_Q = 1$, such that all remaining particles will be assigned to the occupation number n_Q. Thus, this algorithm conserves the total number of particles.

For the case of a finite number of random collisions, C, the particles will have a finite probability that they will not experience a collision. This differs from the case where $C \to \infty$, where all particles undergo a collision and the fraction of particles experiencing a collision goes to one. By defining the expectation value for the fraction of collided particles as ω, the sampling collision operator is given by

$$\Xi_i^{\omega \to 1} = \hat{n}_i - n_i. \tag{1.120}$$

By forcing a fixed collision probability on each particle during a time step Δt, each individual particle will undergo a collision with probability ω. If a given lattice site has N particles, then an average of $N\omega$ collisions will occur. The number of particles undergoing a collision is represented by a binomial distribution in the form

$$P(N^c) = \begin{pmatrix} N \\ N^c \end{pmatrix} w^{N^C} (1-\omega)^{N-N^C}. \tag{1.121}$$

The number of particles associated with velocity v_i undergoing collisions is chosen by

$$n_i^{\omega} = X_{\omega}^{n_i}. \tag{1.122}$$

Then, the sampling algorithm is enacted in the same manner, except that there are now only $N^c = \sum_i n_i^\omega$ particles left to be sampled. The redistributed particles from Eqn. (1.119) can be defined as $\hat{n}_i^\omega$, which then gives the collision operator the form

$$\Xi_i^\omega = \hat{n}_i^\omega - n_i^\omega. \tag{1.123}$$

For $\omega = 1$, the sampling collision operator from Eqn. (1.120) is recovered. Thus, a sampling collision operator for any number of collisions has been defined. This sampling collision operator was the basis and inspiration for investigating the implementation of overrelaxation for ILG which is presented in Chapter 4.

1.4. Outline of the Dissertation

This dissertation first presents a FLBM for the diffusion equation in Chapter 2. This FLBM method is self-contained and presents a density dependent and mode decoupled fluctuation dissipation theorem. Following this, chapter 3 presents an application for diffusive lattice Boltzmann for fluid transport through barrier coatings is presented. In this section, a higher order scheme for lattice Boltzmann with the intention of error correction is presented. Finally in chapter 4.3, with the revitalization of lattice gas methods from Blommel's Monte Carlo integer lattice gas and Seekins' sampling collision operator for integer lattice gases, a method for implementing overrelaxation in integer lattice gases is provided.

2. FLUCTUATING LATTICE BOLTZMANN METHOD FOR THE DIFFUSION EQUATION[1]

We derive a fluctuating lattice Boltzmann method for the diffusion equation. The derivation removes several shortcomings of previous derivations for fluctuating lattice Boltzmann methods for hydrodynamic systems. The comparative simplicity of this diffusive system highlights the basic features of this first exact derivation of a fluctuating lattice Boltzmann method.

2.1. Introduction

This paper introduces a new lattice Boltzmann method for the fluctuating diffusion equation. Much of this work was inspired by earlier work on fluctuating hydrodynamics which resulted in very decent, but ultimately never quite exact discrete representations. This is why we focus here on the arguably simplest lattice Boltzmann method to extend it to its fluctuating counterpart. As we show below this derivation allows us to remove certain inexact assumptions of earlier methods leading to a surprisingly exact and robust fluctuating method. Even the limit of low density, where few particles reside on each lattice site on average, is well behaved. This is unexpected since in a continuous method the noise starts to fully dominate the dynamics of the system at such low densities, and for hydrodynamics fluctuating lattice Boltzmann methods this limit is ill behaved [41].

Let us step back for a moment to consider the interplay of fluctuations and continuous methods. Materials are composed of discrete particles that follow the deterministic evolution of Newton's second law (or more fundamentally the laws of quantum mechanics). The collective evolution of many of these particles evolves through rules that can be discovered through kinetic theory. In many cases there is a fast dynamic due to the local collisions between particles and a slow dynamics of locally conserved quantities that are unchanged by the collisions. Due to the chaotic nature of the collisions the evolution of the locally conserved quantities has a deterministic part

[1]The content of this chapter is published in "Alexander J. Wagner and Kyle Strand. Fluctuating lattice Boltzmann method for the diffusion equation. Phys. Rev. E, 94(3):033302, 2016." Kyle Strand's contributions to this publication were developing the code for the computer simulations, performing analysis on the simulations, creating figures, and drafting and revising the content presented in this chapter.

(i.e. the hydrodynamic equation) and a non-deterministic part that can often be well described as an instantaneous random component. The amount of the random component will depend on the amount of coarse-graining, and it is often assumed that for macroscopic systems these fluctuations will be averaged out, leaving the fully deterministic hydrodynamic equations.

The reason for the fluctuations lies in the discrete nature of our world. It is often argued that the explanation of random motion of small particles, known as Brownian motion, in terms of individual collisions with the molecules of a fluid by Einstein[42] was the clinching argument in favor of the particulate nature of matter. But a continuous description of nature has jettisoned this fundamentally discrete nature of matter, and therefore fluctuations have to be artificially re-introduced. This is achieved through the Langevin method, where the idea is to introduce random forces such that the predicted level of fluctuation is recovered in equilibrium [43]. These fluctuations obey a fluctuation-dissipation theorem where the irreversible dissipative parts of the evolution are balanced by their fluctuating counterpart in equilibrium.

For the Navier-Stokes equation the fluctuating equivalent was derived by Landau and Lifshitz [44]. It consists of complementing the dissipative viscous stress tensor with a fluctuating stress tensor. The first fluctuating lattice Boltzmann method, introduced by Ladd in 1993 [21, 8], used this result and introduced a fluctuating stress tensor into the lattice Boltzmann method. However, this only gives correct results in the hydrodynamic limit, i.e. for small wave numbers in a Fourier representation.

Earlier work around 1970 of Fox and Uhlenbeck [45] as well as Bixon and Zwanzig [46] generalized the linear Boltzmann equation into a Langevin equation by introducing a fluctuating collision operator. Crucially this fluctuating collision operator not only introduces a fluctuating stress tensor but additional fluctuating higher order modes. The idea of using this as a basis for the development of fluctuating lattice Boltzmann even predates the approach by Ladd, and was outlined in a conference proceeding by Dufty and Ernst in 1993 [47] around the same time as Ladd introduced his method. Overcoming the difficulties with this approach was not achieved until 2005 when Adhikari et al. [22] presented a practical implementation. The key improvement introduced here was to include fluctuations not only for the stress tensor, but for all non-conserved modes that the collision operator relaxes. This significantly improved the behavior of the fluctuating lattice Boltzmann method for larger wave numbers. This approach was then re-derived by Duenweg

from considerations of a discrete lattice gas [48]. However, there were two inconsistencies in this approach, both inherited from trying to apply the Fox-Uhlenbeck approach for a linear Boltzmann equation to a BGK-collision operator. In the linear approach, the collision is assumed to bring the local distribution function closer to the global equilibrium distribution (with special projections that still ensure local conservation of conserved quantities) whereas the BGK approach consists of a relaxation towards a local equilibrium distribution, corresponding to the local conserved quantities. A key difference is that the hydrodynamic limit of the linear Boltzmann equation does not recover the non-linear term $\nabla(\rho\mathbf{uu})$ in the Navier-Stokes equation, whereas the BGK collision operator does.

The first inconsistency occurred because in the derivation the local equilibrium distribution was replaced by the global equilibrium distribution at a crucial juncture, a problem that has been addressed by Kaehler et al. [49]. The second inconsistency occurred because it was assumed that the fluctuations of the local equilibrium distribution around the global equilibrium distribution could be neglected. At its heart, the current paper deals with avoiding this incorrect assumption, albeit in the simpler case of a lattice Boltzmann method for diffusion, rather than hydrodynamics. An extension to the hydrodynamic case appears also possible, but is more technically involved and will be reserved for a later publication.

In this paper we first introduce the fluctuating lattice Boltzmann method, and derive the form of the fluctuating collision operator for the diffusion equation. As usual for fluctuating lattice Boltzmann methods since Adhikari et al. [22] this results in a multi-relaxation-time method [50]. While this method is distinct, lattice Boltzmann methods for diffusion phenomena were derived by Wolf-Gladrow [9], and can be obtained in the limit of no flow from the more popular multi-component lattice Boltzmann methods [51, 52]. A multi-relaxation time method related to the current method (albeit with a slightly different collision matrix) can be found in Li et al. [53] and Le [54] (all of these methods, however, exclude fluctuations). In the following section we show that this method does indeed recover the fluctuating diffusion equation in the hydrodynamic limit and the last section shows a number of verifications of our new method, and its fluctuating qualities. We close by giving an outline of future research directions, particularly the impact of this work on fluctuating lattice Boltzmann methods for multi-phase systems, a problem that has been previously addressed by Gross et al. [55], Thampi et al. [56] and by Belardinelli et al. [57].

2.2. Lattice Boltzmann Method

The lattice Boltzman equation is an evolution equation for densities f_i defined on a regular lattice associated with discrete velocities v_i. These densities move in the direction of their associated velocities, and the velocities are typically chosen so that the resulting position is again a lattice position. Densities that collect at one lattice site then undergo local collisions. This can be formally written as

$$f_i(\mathbf{x} + \mathbf{v}_i, t + 1) = f_i(\mathbf{x}, t) + \Omega_i. \tag{2.1}$$

Following Qian et al. [58] one typically makes the assumption that the collision is well approximated by a BGK approximation as simply approaching a local equilibrium distribution f_i^0 which is a function of the locally conserved quantities. To include fluctuations we must consider not only the dissipative relaxation towards the local equilibrium, but also an additional fluctuating term ξ_i. We can then write the full collision term as

$$\Omega_i = \sum_j \Lambda_{ij}[f_j^0 - f_j(x,t)] + \xi_i \tag{2.2}$$

where Λ_{ij} is a collision matrix. The purpose of this paper is to derive a convenient (and exact) form of the noise terms ξ_i as a function of the collision matrix Λ_{ij}.

In this paper we will focus on the simplest case of an ideal gas. For an ideal gas of discrete lattice particles associated with the discrete velocities v_i, we expect their occupation numbers n_i to follow a binomial distribution. For a large enough system this is well approximated by the Poisson distribution

$$P(n_i) = \frac{\exp(-n_i^{eq})(n_i^{eq})^{n_i}}{n_i!}. \tag{2.3}$$

Here we have defined the a priori unknown global equilibrium occupation numbers as

$$n_i^{eq} = \langle n_i \rangle, \tag{2.4}$$

where $\langle \cdots \rangle$ denotes an equilibrium average. These fluctuations are independent for different velocities and different lattice sites. We show in the appendix, eqn. (A.22) that the second moment is

given by

$$\langle n_i(\mathbf{x},t)n_j(\mathbf{x}',t)\rangle = n_i^{eq}n_j^{eq} + n_i^{eq}\delta_{ij}\delta_{\mathbf{x}\mathbf{x}'}. \tag{2.5}$$

Let us briefly consider the difference between Binomial and Poisson distributed fluctuations. Since the collisions are local, they cannot require knowledge of the system size. So we have to be able to assume a Poisson distribution here. However a finite system size will introduce a correlation between the densities, leading to a distribution that depends on the system size and is not identical to the Poisson distribution.

There is a fundamental issue that the lattice Boltzmann densities are real numbers, not integers, so strictly they cannot be Poisson distributed. Also there is no accepted generalization of the Poisson distribution to continuous variables. Simply re-interpreting the factorial using a Γ-function will alter the moments of the distribution function. We will see below that these difficulties become noticeable when we attempt to use the lattice Boltzmann method for very small densities where the discreteness becomes apparent.

These difficulties arise generally when one attempts to add fluctuations to a continuous theory. Einstein's famous 1905 paper [59] was celebrated for explaining Brownian motion as arising from collisions with discrete solvent molecules, thereby unambiguously showing that even apparently continuous liquids are made up of discrete atoms. Here we attempt to derive a continuous theory that mimics the discrete fluctuating dynamics on length-scales that allow for a continuous treatment.

To transfer our simple results for the discrete Poisson distribution to our lattice Boltzmann method we will simply require that the first two moments of the discrete and continuous distributions agree. So we demand that

$$\langle f_i\rangle = f_i^{eq}, \tag{2.6}$$

$$\langle f_i f_j\rangle = f_i^{eq}f_j^{eq} + f_j^{eq}\delta_{ij}. \tag{2.7}$$

The basic idea of the following derivation is now to use our knowledge of the distribution of the $\langle f_i\rangle$ to predict the required fluctuation terms ξ_i that will generate them.

As a brief aside we want to mention here that this raises an issue of the interpretation of the continuous densities f_i. They were originally derived as ensemble averages from discrete lattice gas occupation numbers [50]. As such they should not experience fluctuations. Introducing fluctuations now generates a new quantity that does fluctuate as if there were discrete particles but where the occupation numbers remain continuous. This odd construct has the advantage of conserving the freedom of using a much simpler BGK and more flexible collision operator of eqn (2.2).

Let us consider the connection between the local equilibrium distribution f_i^0 and the global equilibrium distribution f_i^{eq}. We can average the lattice Boltzmann eqn. (2.1) to get

$$\langle f_i(x+v_i,t+1)\rangle = \langle f_i(x,t)\rangle + \sum_j \Lambda_{ij}(\langle f_j^0(x,t)\rangle - \langle f_j(x,t)\rangle) + \langle \xi_i(x,t)\rangle, \tag{2.8}$$

$$f_i^{eq} = f_i^{eq} + \sum_j \Lambda_{ij}(\langle f_j^0(x,t)\rangle - f_j^{eq}) + 0 \tag{2.9}$$

where we used the fact that the expectation values are translationally invariant and that $\langle \xi_i \rangle = 0$, since the noise term is designed to only contain deviations from the mean behavior. We further demand that the collision matrix Λ_{ij} be invertible to get

$$\langle f_i^0(x,t)\rangle = f_i^{eq} \tag{2.10}$$

so the expectation values of the local equilibria are the same as the expectation values of the densities. We will use this insight below to derive the global equilibrium distribution from the imposed local equilibrium distribution of the lattice Boltzmann method.

Next we calculate the expectation value of a lattice Boltzmann equation multiplied with itself, which will give us the noise correlators $\langle \xi_i \xi_j \rangle$. We get

$$\begin{aligned} &\langle f_i(x+v_i,t+1) f_j(x+v_j,t+1)\rangle \\ =&\langle f_i(x,t) f_j(x,t)\rangle + \langle f_i(x,t)\Omega_j\rangle \\ &+ \langle \Omega_i f_j(x,t)\rangle + \langle \Omega_i \Omega_j\rangle. \end{aligned}$$

The first two terms cancel again, and we drop the spatial and temporal dependence for ease of notation. We then get, substituting the BGK collision operator (2.2),

$$\begin{aligned} 0 = &\sum_k \Lambda_{jk} \langle f_i (f_k^0 - f_k) \rangle + \sum_l \Lambda_{il} \langle (f_l^0 - f_l) f_j \rangle \\ &+ \sum_{kl} \Lambda_{ik} \Lambda_{jl} \langle (f_k^0 - f_k)(f_l^0 - f_l) \rangle + \langle \xi_i \xi_j \rangle. \end{aligned} \tag{2.11}$$

We know (or more exactly require) the moment $\langle f_i f_j \rangle$ from eqn. (2.7). All that remains is to find $\langle f_i f_j^0 \rangle$ and $\langle f_i^0 f_j^0 \rangle$.

Up to this point we have not specified what the local equilibrium distribution is. The choice of equilibrium distribution – and of the conserved quantities – determines the partial differential equations simulated in the hydrodynamic limit.

In most cases the equilibrium distribution is simply a function of the locally conserved variables, i.e. those quantities that cannot be relaxed by the collisions and are therefore the slow variables. The evolution of these slow variables is then discovered in the hydrodynamic limit.

We can define the conserved quantities as linear combinations of the f_i

$$M^c = \sum_i m_i^c f_i \tag{2.12}$$

and we then have

$$f_i^0(\{M^c\}), \tag{2.13}$$

i.e. the local equilibrium is only a function of these conserved quantities. And therefore we know the local equilibrium distribution in terms of the f_i.

Up to this point our considerations have been entirely general. To progress from here we now need to define the local equilibrium distribution function. Here we have only one locally conserved quantity, the density, defined as

$$\rho(\mathbf{x}, t) = \sum_i f_i(\mathbf{x}, t). \tag{2.14}$$

To simulate the diffusion equation we demand the following moments for the equilibrium distribution:

$$\sum_i f_i^0 = \rho, \tag{2.15}$$

$$\sum_i f_i^0 v_{i\alpha} = 0, \tag{2.16}$$

$$\sum_i f_i^0 v_{i\alpha} v_{i\beta} = \rho\theta\delta_{\alpha\beta}. \tag{2.17}$$

In section 2.3 we show that this choice of local equilibrium distribution leads to the diffusion equation in the hydrodynamic limit. Here we are interested in a two-dimensional model, but the derivations are near identical for a one or three dimensional model. The simplest two dimensional velocity set consistent with these moments is

$$\{v_i\} = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\} \tag{2.18}$$

which is often also referred to as a D2Q5 model. This leads to the equilibrium distribution

$$f_i^0 = \rho w_i \tag{2.19}$$

with weights w_i given by

$$w_0 = (1 - 2\theta), \qquad w_{1\ldots4} = \frac{\theta}{2}. \tag{2.20}$$

We can now calculate the global equilibrium distribution using eqn. (2.10):

$$\begin{aligned} f_i^{eq} &= \langle f_i^0 \rangle \\ &= \left\langle w_i \sum_j f_j \right\rangle \\ &= w_i \sum_j f_j^{eq} \end{aligned}$$

Let us define

$$\rho^{eq} = \sum_i f_i^{eq} \tag{2.21}$$

and we get

$$f_i^{eq} = \rho^{eq} w_i. \tag{2.22}$$

Here the global equilibrium function has exactly the same form as the local equilibrium function. This result does not hold generally, in particular for instance for hydrodynamics models that conserve momentum [41].

Now we can calculate the equal time correlators we needed for eqn. (2.11). We obtain

$$\begin{aligned}
\langle f_i f_j^0 \rangle &= \left\langle f_i w_j \sum_k f_k \right\rangle \\
&= \sum_k (f_i^{eq} f_k^{eq} + f_i^{eq} \delta_{ik}) w_j \\
&= f_i^{eq} \rho^{eq} w_j + f_i^{eq} w_j \\
&= (\rho^{eq} + 1) \rho^{eq} w_i w_j
\end{aligned} \tag{2.23}$$

and

$$\begin{aligned}
\langle f_i^0 f_j^0 \rangle &= \left\langle w_i \sum_k f_k w_j \sum_l f_l \right\rangle \\
&= \sum_{kl} w_i w_j (f_k^{eq} f_l^{eq} + f_k^{eq} \delta_{kl}) \\
&= (\rho^{eq} + 1) \rho^{eq} w_i w_j.
\end{aligned} \tag{2.24}$$

For this simple model both correlators are the same and we have

$$\langle f_i^0 f_j \rangle - \langle f_i f_j \rangle = \rho^{eq} (w_i w_j - w_i \delta_{ij}). \tag{2.25}$$

As a side note we would like to point out that earlier derivations of fluctuating lattice Boltzmann methods [22, 48, 49] did not know how to treat these correlators and made an assumption equivalent to neglecting the δ_{ij} term. This resulted in formal results that did not require the fluctuations on

conserved moments to vanish. This requirement was then added by hand on physical grounds, an argument which this new derivation avoids, as will become clear below.

Using this result in (2.11) we obtain

$$\begin{aligned}\langle \xi_i \xi_j \rangle = \rho^{eq} \Bigg[& \sum_{kl} \Lambda_{ik} \Lambda_{jl} (w_k w_l - w_k \delta_{kl}) \\ & - \sum_{k} \Lambda_{jk} (w_i w_k - w_i \delta_{ik}) \\ & - \sum_{l} \Lambda_{il} (w_l w_j - w_l \delta_{lj}) \Bigg] .\end{aligned}$$

This fluctuation dissipation relation is, in some sense, our answer, but it is not a nice answer. We have a full correlation matrix for the noise terms, suggesting that the noise terms are not independent.

This is not a surprise, but rather a consequence of the local conservation laws which require

$$\sum_i \xi_i = 0 \tag{2.26}$$

since the fluctuations can't create or destroy mass.

This suggests that moving to a different representation in velocity space which separates out the conserved moment may be useful. This moment representation is analogous to what has been used for deriving fluctuating hydrodynamics in the past since the paper by Adhikari [22]. Here we employ the moment transformation rather later in the derivation, which avoids having to perform a rather cumbersome Fourier transform which complicated earlier derivations.

We now define moments through a general linear transformation

$$M^a = \sum_i m_i^a f_i, \tag{2.27}$$

which will include the conserved moments mentioned in eqn. (2.12), together with a back-transform

$$f_i = \sum_a n_i^a M^a \tag{2.28}$$

so that we have the relations

$$\sum_i n_i^a m_i^b = \delta^{ab} \qquad \sum_a m_i^a n_j^a = \delta_{ij} \tag{2.29}$$

and we can move freely between f and M space. We already know that we want $M^0 = \rho = \sum_i 1 f_i$, which is equivalent to

$$m_i^0 = (1, 1, 1, 1, 1). \tag{2.30}$$

One additional consideration for choosing the moments is the collision matrix. If we use a single-relaxation time collision matrix it is always diagonal. In f-space it is given by

$$\Lambda_{ij} = \frac{1}{\tau}\delta_{ij} \tag{2.31}$$

and in moment space it is given by

$$\Lambda^{ab} = \sum_{ij} m_i^a \Lambda_{ij} n_j^b = \frac{1}{\tau}\sum_{ij} m_i^a \delta_{ij} n_j^b = \frac{1}{\tau}\delta^{ab}. \tag{2.32}$$

For a more general multi-relaxation time method we simply demand that Λ^{ab} be diagonal:

$$\Lambda^{ab} = \frac{1}{\tau^a}\delta^{ab}. \tag{2.33}$$

In moment space we then get

$$\begin{aligned}\langle \xi^a \xi^b \rangle = \rho^{eq} \Bigg[& \sum_{ijkl} m_i^a m_j^b \Lambda_{ik} \Lambda_{jl} (w_k w_l - w_k \delta_{kl}) \\ & - \sum_{ijk} m_i^a m_j^b \Lambda_{jk} (w_i w_k - w_i \delta_{ik}) \\ & - \sum_{ijl} m_i^a m_j^b \Lambda_{il} (w_l w_j - w_l \delta_{lj}) \Bigg].\end{aligned} \tag{2.34}$$

Careful inspection reveals that this simplifies if we choose

$$n_i^a = m_i^a w_i. \tag{2.35}$$

The matrix of moments m_i^a is generated by selecting physically relevant vectors 1, v_{ix}, v_{iy}, $v_{ix}v_{ix} - v_{iy}v_{iy}$, and $v_{ix}v_{ix} + v_{iy}v_{iy}$ and employing the Gram-Schmidt orthonormalization procedure with respect to the scalar product

$$\langle m^a m^b \rangle_M = \sum_i m_i^a w_i m_i^b \stackrel{!}{=} \delta^{ab}. \tag{2.36}$$

The resulting moment matrix is

$$m_i^a = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & \sqrt{\frac{1}{\theta}} & -\sqrt{\frac{1}{\theta}} & 0 & 0 \\ 0 & 0 & 0 & \sqrt{\frac{1}{\theta}} & -\sqrt{\frac{1}{\theta}} \\ 0 & \sqrt{\frac{1}{2\theta}} & \sqrt{\frac{1}{2\theta}} & -\sqrt{\frac{1}{2\theta}} & -\sqrt{\frac{1}{2\theta}} \\ -\sqrt{\frac{2\theta}{1-2\theta}} & \sqrt{\frac{1-2\theta}{2\theta}} & \sqrt{\frac{1-2\theta}{2\theta}} & \sqrt{\frac{1-2\theta}{2\theta}} & \sqrt{\frac{1-2\theta}{2\theta}} \end{pmatrix} \tag{2.37}$$

It may be noted that this matrix differs from the one presented by Li et al. [53] or Le et al. [54]. The eigenvectors of the collision matrix in these approaches differ from the current ones only by a factor and additions of conserved eigenvectors. Kaehler et al. [60] showed that this ensures that the collision terms are equivalent.

For the equilibrium distribution in moment space we obtain

$$M^{a,0} = \sum_i m_i^a f_i^0. \tag{2.38}$$

For system with mass conservation the first moment is always given by $m_i^0 = (1, 1, 1, 1, 1)$, and it can be freely inserted as a factor. This implies

$$M^{a,0} = \sum_i m_i^a f_i^0 m_i^0 = \rho \sum_i m_i^a w_i m_i^0 = \rho \delta^{a0} \tag{2.39}$$

i.e. the local equilibrium density is ρ for the first moment and vanishes for all other moments by virtue of the orthogonalization condition of the moments.

Note that this also implies

$$\sum_i m_i^a w_i = \delta^{a0}. \tag{2.40}$$

We now obtain for the last term of the noise amplitudes of eqn. (2.34)

$$\begin{aligned}
&\sum_{jk} m_j^b \Lambda_{jk} w_k m_k^0 \sum_i m_i^a w_i - \sum_{jk} m_j^b \Lambda_{ik} w_k m_k^a \\
=&\Lambda^{b0}\rho^{eq} - \Lambda^{ab} \\
=&\rho(\delta^{a0}\delta^{b0} - 1)\delta^{ab}(1/\tau^a)
\end{aligned}$$

and for the first term we get

$$\begin{aligned}
&\sum_{ik} m_i^a \Lambda_{ik} w_k m_k^0 \sum_{jl} m_j^b \Lambda_{jl} w_l m_l^0 \\
&- \sum_{ijkl} m_i^a \Lambda_{ik} w_k \sum_c m_k^c w_l m_l^c \Lambda_{jl} m_j^b \\
=&\Lambda^{a0}\Lambda^{b0} - \sum_c \Lambda^{ac}\Lambda^{bc} \\
=&(\delta^{a0}\delta^{b0} - 1)\delta^{ab}(1/(\tau^a)^2).
\end{aligned}$$

With this we get

$$\langle \xi^a \xi^b \rangle = \rho^{eq} \frac{2\tau^a - 1}{(\tau^a)^2}(1 - \delta^{a0}\delta^{b0})\delta^{ab}. \tag{2.41}$$

In this representation the noise terms are diagonal, i.e. no longer correlated, and we explicitly see that the noise amplitude for the conserved mode is zero.

However, this does not quite restrict the noise amplitudes. This is fortunate since it seems rather unsatisfactory that the noise should depend on some global average density, a bit of information that should not be available to a local collision operator. This was first argued by Kaehler et al. [49].Let us define $\langle \cdots \rangle|_\rho$ as the ensemble average over a all states for a cell with local density ρ. We can then write a local noise term of the form

$$\langle \xi^a \xi^b \rangle|_\rho = \rho \frac{2\tau^a - 1}{(\tau^a)^2}(1 - \delta^{a0}\delta^{b0})\delta^{ab}. \tag{2.42}$$

While local noise had been proposed before by Kaehler, we are now able to show that local noise is consistent with (2.41), which was not possible in earlier derivations.

$$\begin{aligned}\left\langle \langle \xi^a \xi^b \rangle|_\rho \right\rangle &= \langle \rho \rangle \frac{2\tau^a - 1}{(\tau^a)^2}(1 - \delta^{a0}\delta^{b0})\delta^{ab} \\ &= \rho^{eq} \frac{2\tau^a - 1}{(\tau^a)^2}(1 - \delta^{a0}\delta^{b0})\delta^{ab}\end{aligned}$$

which is now consistent with (2.41), and this local formulation does not require knowledge of the whole system to determine the local noise amplitude. We will show below that local noise is required to fully recover the correct behavior of the system, particularly for non-equilibrium systems with large variations in the density.

2.3. Hydrodynamic Limit

We claimed earlier that the choice of equilibrium distribution given by the moments (2.15–2.17) will lead the lattice Boltzmann method to simulate a diffusion equation. In this section we will now derive the hydrodynamic limit of the fluctuating lattice Boltzmann method and show that we recover a fluctuating diffusion equation.

This derivation follows the approach developed by Kaehler et al. [60]. We write the lattice Boltzmann eqn. (2.1) in terms of a Taylor expansion to get

$$(\partial_t + v_{i\alpha}\partial_\alpha) f_i + \frac{1}{2}(\partial_t + v_{i\alpha}\partial_\alpha)^2 f_i = \sum_j \Lambda_{ij}(f_j^0 - f_j) + \xi_i. \tag{2.43}$$

We can use this equation to express the distribution f_i in terms of the local equilibrium distribution and derivatives. Using that the collision matrix is invertible we get to first order

$$f_i = f_i^0 - \sum_j \Lambda_{ij}^{-1}(\partial_t + v_{j\alpha}\partial_\alpha) f_j + \sum_j \Lambda_{ij}^{-1}\xi_j. \tag{2.44}$$

Now we reinsert this expression into our expanded lattice Boltzmann eqn. (2.43) to get

$$\begin{aligned}
&(\partial_t + v_{i\alpha}\partial_\alpha)\left(f_i^0 + \sum_j \Lambda_{ij}^{-1}\xi_j\right) \\
&- (\partial_t + v_{i\alpha}\partial_\alpha)\sum_j \left(\Lambda_{ij}^{-1} - \frac{1}{2}\delta_{ij}\right)(\partial_t + v_{j\beta}\partial_\beta)f_j \\
=&\sum_j \Lambda_{ij}(f_j^0 - f_j) + \xi_i. \qquad (2.45)
\end{aligned}$$

We then reinsert eqn. (2.44) to also replace the remaining occurrence of f_i on the left hand side and sum the resulting equation over i to obtain the hydrodynamic equation for the evolution of the density ρ. We get

$$\begin{aligned}
&\partial_t\rho + \partial_\alpha \sum_{ij} v_{i\alpha}\Lambda_{ij}^{-1}\xi_j \\
&- \sum_i (\partial_t + v_{i\alpha}\partial_\alpha)\sum_j \left(\Lambda_{ij}^{-1} - \frac{1}{2}\delta_{ij}\right) \\
&(\partial_t + v_{j\beta}\partial_\beta)\left(f_j^0 + \sum_k \Lambda_{jk}^{-1}\xi_k\right) = 0. \qquad (2.46)
\end{aligned}$$

The collision matrix should be isotropic, i.e. relaxation should be invariant under rotation. This restricts the collision matrix (2.32) to be of the form

$$\Lambda^{ab} = \mathrm{diag}\left(\frac{1}{\tau^\rho}, \frac{1}{\tau^j}, \frac{1}{\tau^j}, \frac{1}{\tau^n}, \frac{1}{\tau^s}\right). \qquad (2.47)$$

Now let us consider $v_{i\alpha}\Lambda_{ij}^{-1}$. Noting that we constructed the moment matrix such that $v_{i\alpha} = \sqrt{\theta}m_i^{\alpha+1}$ we can write

$$\begin{aligned}
\sum_i v_{i\alpha}\Lambda_{ij}^{-1} &= \sum_i \sum_{ab} \sqrt{\theta} m_i^{\alpha+1} n_i^a \tau^a \delta^{ab} m_j^b \\
&= \sqrt{\theta}\sum_b \delta^{\alpha+1,b}\tau^b m_j^b \qquad (2.48) \\
&= \tau^j v_{j\alpha}. \qquad (2.49)
\end{aligned}$$

This shows that the velocity $v_{i\alpha}$ is a left eigenvector of the inverse collision matrix with eigenvalue τ^j. Similarly 1_i (i.e. a vector consisting entirely of ones) is also a left eigenvector with the eigenvalue τ^ρ. Note that this eigenvalue is arbitrary since the density is conserved due to eqn. (2.15). This implies that all terms containing this arbitrary factor must cancel from the hydrodynamic equations.

For the second noise term we need to consider

$$\sum_{ij} v_{i\alpha}\Lambda^{-1}_{ij}v_{j\beta}\Lambda^{-1}_{jk} = \frac{1}{\tau^j}\sum_j v_{j\alpha}v_{j\beta}\Lambda^{-1}_{jk}. \tag{2.50}$$

For our D2Q5 model $v_{ix}v_{iy} = 0$ by construction, so we only need to worry about the $v_{ix}v_{ix}$ and the $v_{iy}v_{iy}$ term in the expression above. We can write these in terms of the left eigenvectors as

$$v_{ix}v_{ix} = \frac{1}{4}\left(m_i^0 + \sqrt{8\theta}\, m_i^4 + \sqrt{\frac{1-2\theta}{2\theta}}m_i^5\right) \tag{2.51}$$

and we get

$$\sum_i v_{ix}v_{ix}\Lambda^{-1}_{ij} = \tau^\rho m_i^0 + \sqrt{8\theta}\,\tau^n m_i^4 + \sqrt{\frac{1-2\theta}{2\theta}}\tau^s m_i^5. \tag{2.52}$$

If we now define the macroscopic noise terms as

$$\eta_\alpha = \tau^j \sum_i v_{i\alpha}\xi_i = \tau^j\sqrt{\theta}\xi^{1+\alpha} \tag{2.53}$$

$$\begin{aligned}\chi_{\alpha\beta} = \delta_{\alpha\beta}\sum_i \Bigg[&\tau^s\frac{1-2\theta}{2}(v_{ix}v_{ix} + v_{iy}v_{iy}) \\ &+\tau^n(-1)^{\delta_{\alpha y}}\frac{1}{2}(v_{ix}v_{ix} - v_{iy}v_{iy})\Bigg]\xi_i.\end{aligned} \tag{2.54}$$

We now get for (2.46)

$$\begin{aligned}&\partial_t\rho + \partial_\alpha\eta_\alpha + \partial_t(\tau^\rho - \frac{1}{2})(\partial_t\rho + \partial_\beta\eta_\beta) \\ &+ \partial_\alpha(\tau^j - \frac{1}{2})\left[\partial_t\eta_\alpha + \partial_\beta(\rho\theta\delta_{\alpha\beta} + \chi_{\alpha\beta})\right] = 0.\end{aligned} \tag{2.55}$$

The third term here contains of a time derivative of the first two terms and therefore is of third order. The appearance of the irrelevant relaxation time τ^ρ implies that this term will vanish not only to third, but to all orders. It is therefore neglected. We then arrive at the macroscopic

fluctuating diffusion equation

$$\partial_t \rho + \partial_\alpha \eta_\alpha + \partial_\alpha \frac{D}{\theta}(\partial_\alpha \rho\theta + (\partial_t \eta_\alpha + \partial_\beta \chi_{\alpha\beta})) = 0 \tag{2.56}$$

where we introduced the diffusion constant

$$D = \left(\tau^j - \frac{1}{2}\right)\theta \tag{2.57}$$

Now formally it may look as if $\partial_\alpha \eta_\alpha$ was of order $O(\partial)$. However, in equilibrium this random part of the diffusion current will be on average equal to the deterministic restoring part of the diffusion current $\partial_\alpha \partial_\alpha(\rho\theta)$ and it will therefore also be of order $O(\partial^2)$. Note that this is equivalent to the way one shows that the forcing term in a lattice Boltzmann method has to be of order $O(\partial)$, see e.g. Li et al. [61]. Therefore the terms

$$\partial_t \eta_\alpha + \partial_\beta \chi_{\alpha\beta} = O(\partial^3) \tag{2.58}$$

are of third order.

We then get

$$\partial_t \rho + \partial_\alpha \eta_\alpha + \partial_\alpha \frac{D}{\theta} \partial_\alpha \rho\theta = 0. \tag{2.59}$$

For a constant temperature θ this leaves us with the standard fluctuating diffusion equation

$$\partial_t \rho + \partial_\alpha \eta_\alpha + \partial_\alpha D \partial_\alpha \rho = 0 \tag{2.60}$$

where the noise current η_α obeys

$$\langle \eta_\alpha \rangle = 0 \tag{2.61}$$

$$\begin{aligned} \langle \eta_\alpha \eta_\beta \rangle &= \tau^j \tau^j \theta \langle \xi^{1+\alpha} \xi^{1+\beta} \rangle \\ &= (2\tau^j - 1)\rho^{eq} \theta \delta_{\alpha\beta} \\ &= 2\rho^{eq} D \delta_{\alpha\beta} \end{aligned} \tag{2.62}$$

for the global noise implementation and

$$\langle \eta_\alpha \rangle = 0 \tag{2.63}$$

$$\langle \eta_\alpha \eta_\beta \rangle = 2\rho D \delta_{\alpha\beta} \tag{2.64}$$

for the local noise implementation.

2.4. Simulation Results

To fully determine the algorithm we need to briefly discuss our implementation of the noise of eqn. (2.41) and (2.42). For simplicity we implement the uncorrelated ξ^a noise terms with a flat distribution:

$$P(\xi^a) = \begin{cases} \frac{1}{2d^a} & \text{for } -d^a < \xi^a < d^a \\ 0 & otherwise \end{cases} \tag{2.65}$$

with

$$d^a = \sqrt{3\rho^{eq}\frac{2\tau^a - 1}{(\tau^a)^2}} \tag{2.66}$$

for the global noise implementation and

$$d^a = \sqrt{3\rho\frac{2\tau^a - 1}{(\tau^a)^2}} \tag{2.67}$$

for the local noise implementation. For negative densities, which are unphysical but may appear in this numerical method, this is treated as if the density was zero. This fully determines our algorithm.

Now we need to test the consistency of the proposed algorithm and evaluate its limitations. We have shown that the noise terms of eqn. (2.41) are necessary to reproduce imposed correlators of the distribution functions in equilibrium given by eqn. (2.7). However, it is not clear that this is sufficient. Past implementations for fluctuating hydrodynamics have shown noticeable deviations from this expectation [49, 62].

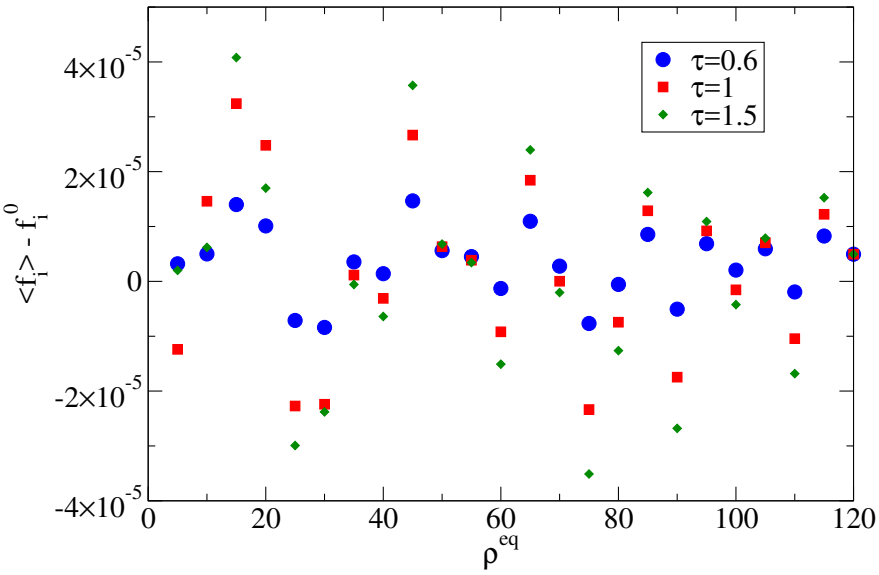


Figure 2.1. Difference of the predicted equilibrium distribution of eqn. (2.19) and the measured equilibrium distribution as a function of the average density in the range of 0 to 120 for different relaxation times. The results are for $\theta = 1/3$ averaged over 10^6 iterations and show excellent agreement. Results shown are for the local noise implementation, the global noise implementation shows similarly small errors.

Firstly we need to check that the equilibrium distribution predicted by eqn. (2.22) is indeed reproduced by our method. We compare this with a time and space average

$$\langle f_i \rangle_{exp} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{L^2} \sum_{\mathbf{x}} f_i(\mathbf{x}, t) \tag{2.68}$$

with the equilibrium distribution for different average densities in Fig. 2.1. Here we sum over the whole two dimensional lattice, which for simplicity of notation is supposed to be square and have L points in each of the two directions. We observe excellent agreement independent of the relaxation times and even for low densities.

A small note: for hydrodynamic lattice Boltzmann methods there is a difference between the effective temperature for local and global equilibrium distributions. However, this effect is closely linked to momentum conservation, which is absent in the model considered here.

Next we examine the second order moments of the distribution functions

$$d_{ij} = \frac{\langle f_i f_j \rangle - f_i^{eq} f_j^{eq}}{\sqrt{f_i^{eq} f_j^{eq}}} \stackrel{?}{=} \delta_{ij} \tag{2.69}$$

where we re-expressed eqn. (2.7) to give an expression that is intuitive to test. We numerically evaluate d_{ij} by averaging over the whole lattice and over a large number of iterations as

$$d_{ij}^{exp} = \frac{1}{T}\frac{1}{L^2}\sum_{t=1}^{T}\sum_{\mathbf{x}}\frac{f_i(\mathbf{x},t)f_j(\mathbf{x},t) - \rho^{eq}w_i\rho^{eq}w_j}{\rho^{eq}\sqrt{w_i w_j}}. \tag{2.70}$$

Results of this averaging of eqn. (2.70) were acquired for different lattice sizes over $1.7 \cdot 10^6$ iterations. Other simulation parameters were kept constant at $\tau^j = \tau^s = \tau^n = 1$, $\theta = 1/3$, $\rho^{eq} = 120$. The noise amplitude was given by eqn. (2.42). For a 3x3 lattice,

$$d_{ij}^{exp} = \begin{pmatrix} 0.96268 & -0.02643 & -0.02622 & -0.02624 & -0.02638 \\ -0.02643 & 0.98113 & -0.01867 & -0.01851 & -0.01868 \\ -0.02622 & -0.01867 & 0.98113 & -0.01770 & -0.01888 \\ -0.02624 & -0.01851 & -0.01770 & 0.98163 & -0.01894 \\ -0.02638 & -0.01868 & -0.01888 & -0.01898 & 0.98085 \end{pmatrix}.$$

For a 10x10 lattice,

$$d_{ij}^{exp} = \begin{pmatrix} 0.99655 & -0.00244 & -0.00238 & -0.00230 & -0.00235 \\ -0.00244 & 0.99828 & -0.00172 & -0.00164 & -0.00164 \\ -0.00238 & -0.00172 & 0.99835 & -0.00168 & -0.00174 \\ -0.00230 & -0.00164 & -0.00168 & 0.99854 & -0.00171 \\ -0.00235 & -0.00164 & -0.00174 & -0.00171 & 0.99833 \end{pmatrix}.$$

For a 100x100 lattice,

$$d_{ij}^{exp} = \begin{pmatrix} 0.99997 & -0.00002 & -0.00004 & -0.00002 & -0.00001 \\ -0.00002 & 0.99998 & -0.00002 & -0.00001 & -0.00002 \\ -0.00004 & -0.00002 & 0.99999 & -0.00002 & -0.00001 \\ -0.00002 & -0.00001 & -0.00002 & 0.99996 & -0.00002 \\ -0.00001 & -0.00002 & -0.00001 & -0.00002 & 0.99999 \end{pmatrix}.$$

We observe that the second equality in eqn. (2.69) is not exactly fulfilled. However, the agreement gets better the larger the lattice is. In eqn. (2.3) we assumed that the n_i are Poisson distributed, which is only true when we are dealing with an infinite system. For a finite system we would have a binomial distribution. Physically the origin of this difference is a slight correlation of the distribution function on finite lattices because of global conservation laws, similar to what Ollila et al. [63] observed for momentum conservation. In particular the total density is conserved

$$\sum_x \sum_i f_i(\mathbf{x}, t) = L^2 \rho^{eq}. \tag{2.71}$$

which implies that the f_i are correlated. The importance of this correlation then depends on the finite system size, and will vanish in the limit of large systems.

A useful way to separate out the globally conserved modes from non-conserved ones is to express the densities in terms of their Fourier modes. We define the discrete Fourier sums for the densities as

$$f_i(\mathbf{k}, t) = \sum_x \exp\left(\frac{2\pi i}{L}\mathbf{k}.\mathbf{x}\right) f_i(\mathbf{x}, t) \tag{2.72}$$

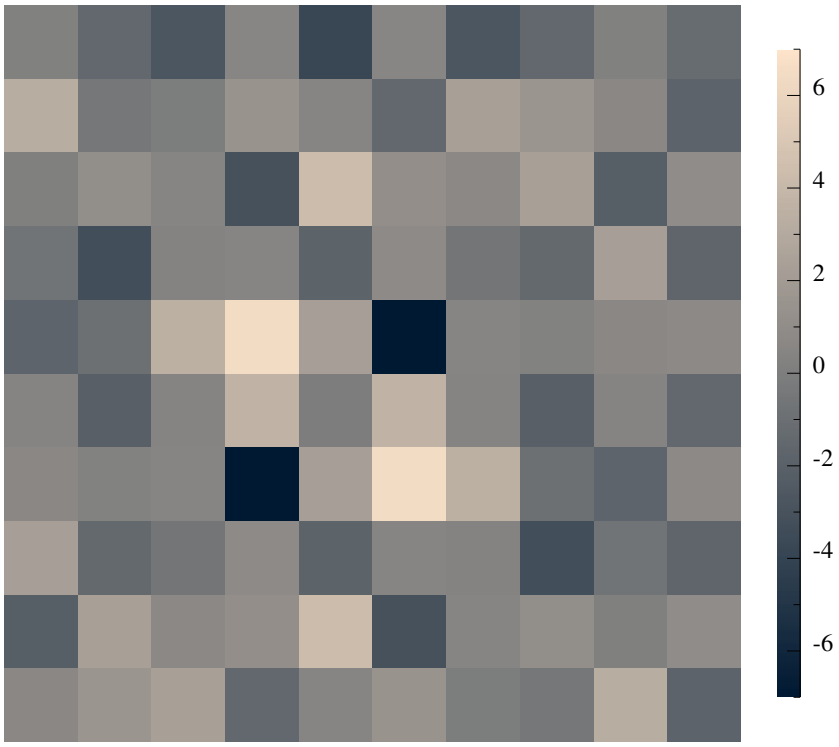


Figure 2.2. Deviation of Fourier modes from eqn. (2.76) for a lattice of $L = 10$ averaged for $T = 336 \times 10^6$ shows no discernable structure. Note that each of the $f(\mathbf{k}, t)f(-\mathbf{k}, t)$ to be averaged has values varying between about $\pm 50,000$, so the remaining scale of ± 7 after averaging is very small.

where L is the total number of lattice points in each spatial direction (assuming a square lattice for ease of notation). We then find a structure factor for the densities f_i:

$$\begin{aligned} S_{ij}(\mathbf{k}) =& \langle f_i(\mathbf{k}, t) f_j(-\mathbf{k}, t) \rangle \\ =& \sum_{\mathbf{x}} \sum_{\mathbf{x}'} \exp\left(\frac{2\pi i}{L} \mathbf{k}.(\mathbf{x} - \mathbf{x}')\right) \langle f_i(\mathbf{x}, t) f_j(\mathbf{x}', t) \rangle \\ =& \sum_{\mathbf{x}} \sum_{\mathbf{x}'} \exp\left(\frac{2\pi i}{L} \mathbf{k}.(\mathbf{x} - \mathbf{x}')\right) [f_i^{eq} f_j^{eq} + f_i^{eq} \delta_{ij} \delta_{\mathbf{x}\mathbf{x}'}] \\ =& \delta_{\mathbf{k}0} L^4 f_i^{eq} f_j^{eq} + L^2 f_i^{eq} \delta_{ij} \end{aligned} \tag{2.73}$$

which should be flat, except at $k = 0$. This result is also not expected to be entirely correct, since we have mass conservation in our finite system. In contrast to the results for the d_{ij}, however, the problem caused by mass conservation is limited to the $k = 0$ mode. We know that mass

conservation requires

$$\begin{aligned}
&\sum_{ij} f_i(\mathbf{k}=0,t) f_j(\mathbf{k}=0,t) \\
=&\sum_{ij} \sum_{x} f_i(\mathbf{x},t) \sum_{\mathbf{x}'} f_j(\mathbf{x}',t) \\
=&L^4(\rho^{eq})^2
\end{aligned} \tag{2.74}$$

whereas eqn. (2.73) predicts $L^4(\rho^{eq})^2 + L^2\rho^{eq}$. We therefore predict the full structure factor for the f_i to be

$$S_{ij}^{th}(\mathbf{k}) = \delta_{\mathbf{k}0} L^4 f_i^{eq} f_j^{eq} + (1-\delta_{\mathbf{k}0}) L^2 f_i^{eq} \delta_{ij}. \tag{2.75}$$

In Fig. 2.2 we show the deviation

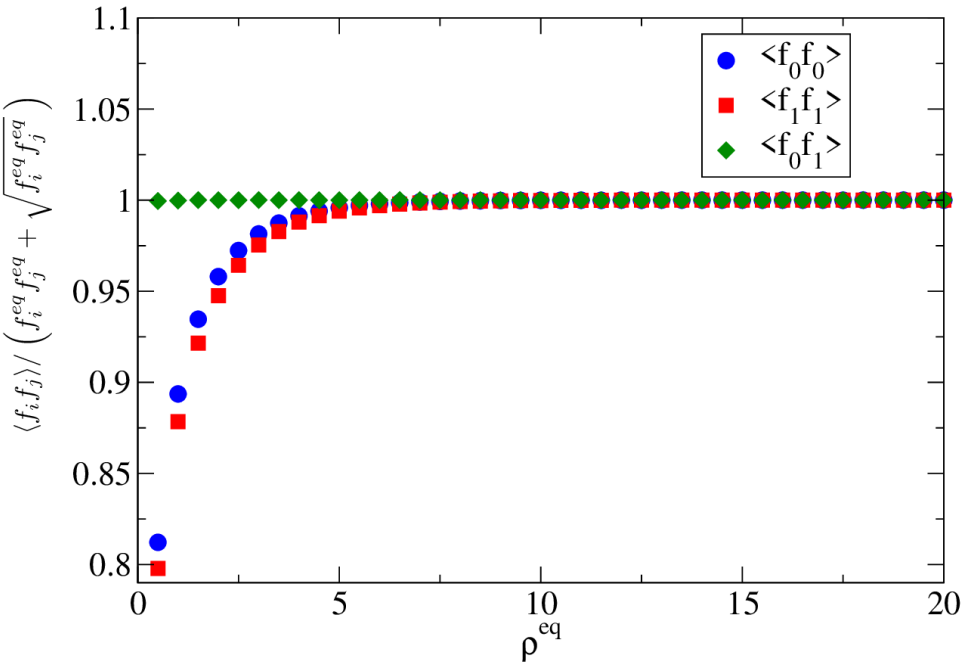
$$S_{ij}^{diff}(\mathbf{k}) = \frac{1}{T} \sum_{t=0}^{T} f_i(\mathbf{k},t) f_j(-\mathbf{k},t) - S_{ij}^{th}(\mathbf{k}) \tag{2.76}$$

and see that, unlike in previous fluctuating methods, we were unable to detect any remaining spurious structure.

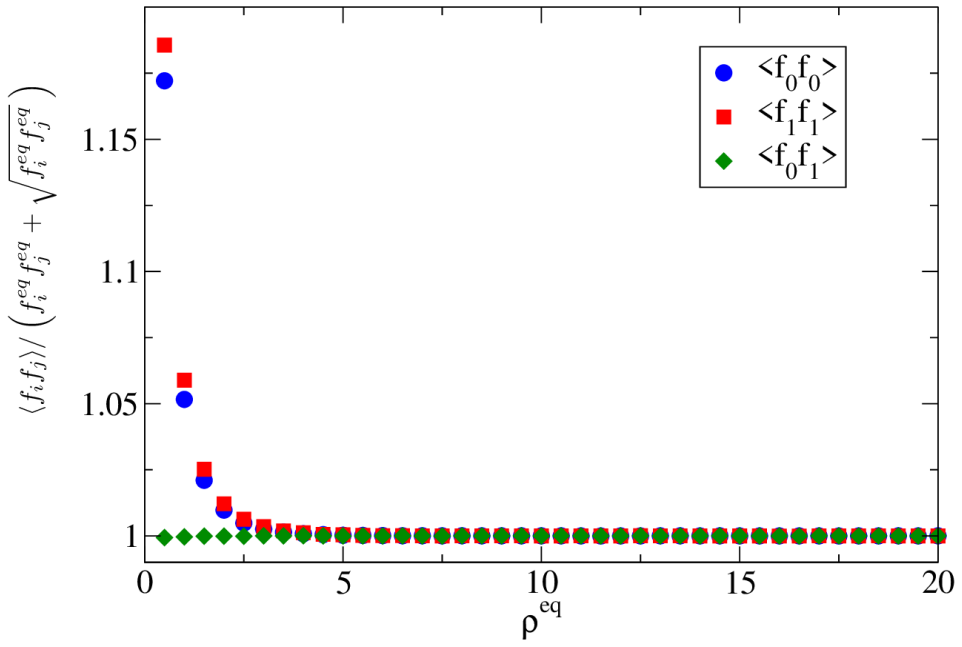
The results so far show that we are able to recover the expected averages and variances for the densities f_i, at least in the limit of infinite system size or in Fourier representation for non-zero Fourier modes. These are recovered for noise amplitudes that are dependent on a local density of eqn. (2.42) or on the global density of eqn. (2.41), and the differences between these two implementations are insignificant.

2.4.1. Limit of Low Density

As previously mentioned one might expect this method to fail in the limit of low density. When lattice Boltzmann densities sum up to only a few particles or even a fraction of a particle the corresponding Poisson distribution no longer resembles a continuous distribution. However, it is important to know the limits of a numerical method. We therefore resolved to determine the density at which our method ceases to give reasonable results.



(a)



(b)

Figure 2.3. Correlators $\langle f_i f_j \rangle$ divided by the predicted result of eqn. (2.7) as a function of the mean densities. Ideally this expression would have a constant value of 1. Unsurprisingly for low densities some deviations are observed. It is noted that the deviation for the global noise amplitude (a) sets in below $\rho^{av} \approx 7$, whereas it only sets in at about $\rho^{eq} \approx 3$ for the local noise implementation (b), and the agreement is quantitatively better.

Table 2.1. The first three moments of the distributions of Fig. 2.4.

(a) $\rho^{eq} = 30$

	$\langle f_1 \rangle$	$\langle f_1^2 \rangle$	$\langle f_1^3 \rangle$
Poisson	5.000	30.000	205
Global noise	4.997	29.971	200
Local noise	5.001	30.005	202

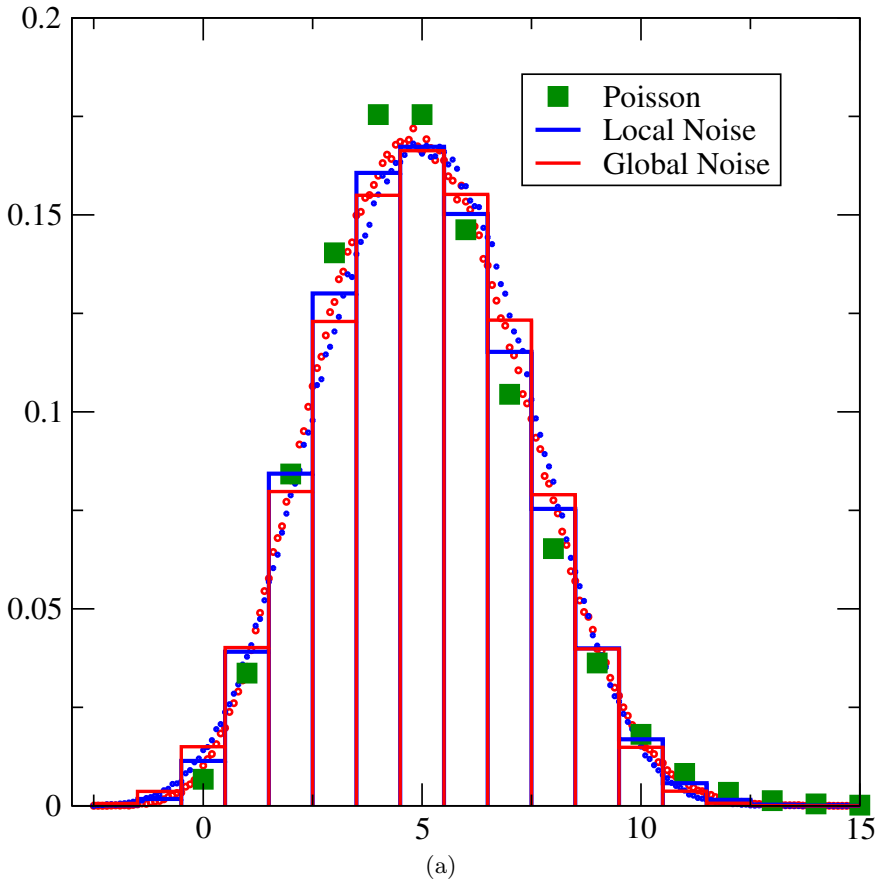
(b) $\rho^{eq} = 1$

	$\langle f_1 \rangle$	$\langle f_1^2 \rangle$	$\langle f_1^3 \rangle$
Poisson	0.1666	0.1944	0.254
Global noise	0.1711	0.1710	0.091
Local noise	0.1666	0.2071	0.164

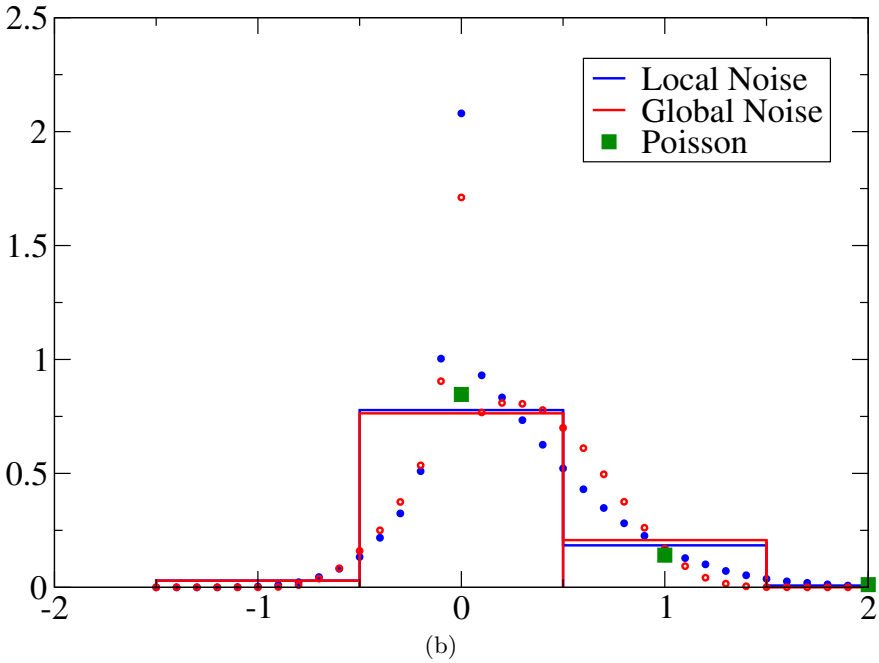
We examined the correlators $\langle f_i(x,t) f_j(x,t) \rangle$ for increasingly lower densities. The result is shown in Fig. 2.3. Even for low average densities the second moments for the density are recovered with surprising fidelity down to very small average numbers of particles. The error remains below 1% for an average density of ρ^{eq} of about 7 for the global noise implementation of eqn. (2.41), but is noticeably better for the local noise implementation of eqn. (2.42). Here the threshold of 1% deviation is reached only at a density of $\rho^{eq} \approx 3$. At this density about 5% of sites will have negative densities, where we suspended the fluctuations. It is also interesting to note that global noise leads to smaller fluctuations, whereas local noise increases the fluctuations at low densities. We did not investigate the exact reasons for these deviations since they are small, and approach a 10% error only for densities of less than one particle per lattice site, where one should no longer expect the continuous description to give good description of fluctuations.

We performed a closer comparison of the distribution of the densities. For the case of f_1 this analysis is shown in Fig. 2.4. We generated histograms with the local and global noise implementations with a resolution of 1 particle and 1/10 of a particle and compared them to the Poisson distribution.

For a total density of $\rho^{eq} = 30$ we find fairly good agreement, although the histograms do not exactly match up with each other or the Poisson distribution. Most obviously we see that there is a finite contribution from negative f_1. A direct comparison of the first three moments of these distribution functions is shown in Table 2.0a. We see that the first two moments match up nicely,



(a)



(b)

Figure 2.4. Comparison of observed distribution functions for local and global noise amplitudes to the Poisson distribution. We show the distribution for f_1 for (a) $f_1^{eq} = 5$ (for $\rho^{eq} = 30$) and (b) $f_1^{eq} = 1/6$ (for $\rho^{eq} = 1$). The small dots represent a histogram with smaller binning.

with the error being slightly smaller for the local noise implementation, which would have been expected from the results presented in Fig. 2.3.

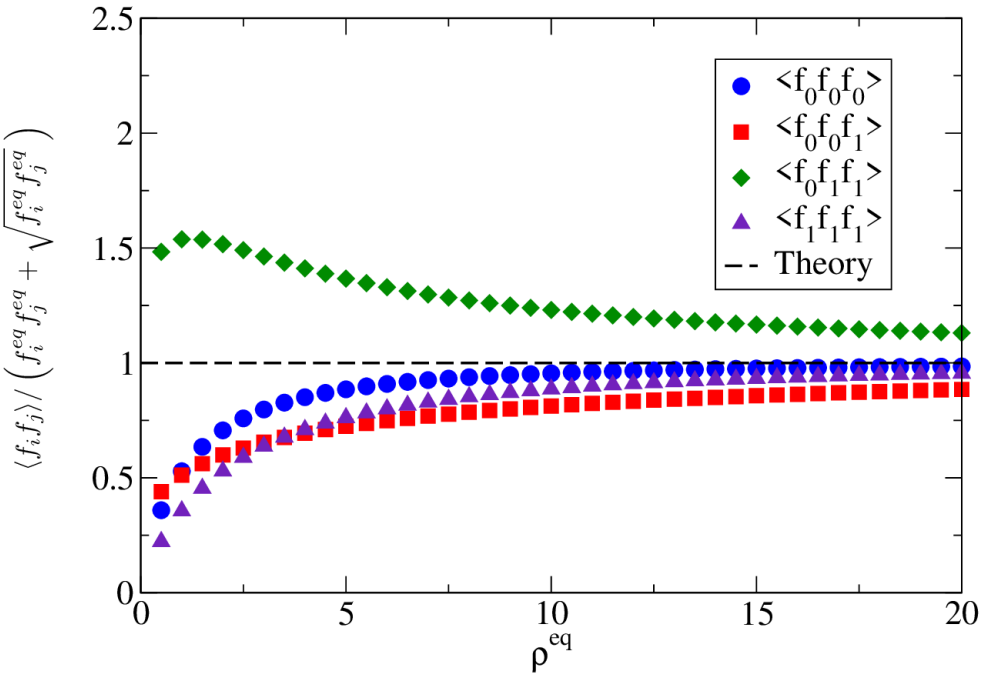
So far we have only discussed the first two moments since those were the moments that our theory intended to match. However it is reasonable to also look at the third moment, even though there is no theoretical analysis that claims these moments should match. We see in Table 2.1 that the agreement for these third moments is indeed noticeably worse, but the trend of a better fit for the local noise continues.

Secondly we examined the distribution of f_1 for only one particle per lattice site on average. This is shown in Fig. 2.4b. The Poisson distribution consists essentially only of empty lattice cells and cells with one particle. The continuous generalizations of the Poisson distribution generated by the global and local noise algorithm differ slightly from each other, but both share the curious feature of an apparent singular contribution for a distribution of exactly zero. We do not currently understand this feature. It is not due to the cessation of noise application at zero density, since this only applies to the local noise algorithm, not the global one. The matching of the moments to the Poisson distribution, shown in Table 2.0b, is no longer as good. The first moments still match, but there is a 10% deviation for the second moment in the local noise implementation, and slightly more for the global one. The third moments now vary by much more, and the match between the local noise implementation to the Poisson distribution is much better that the global noise one.

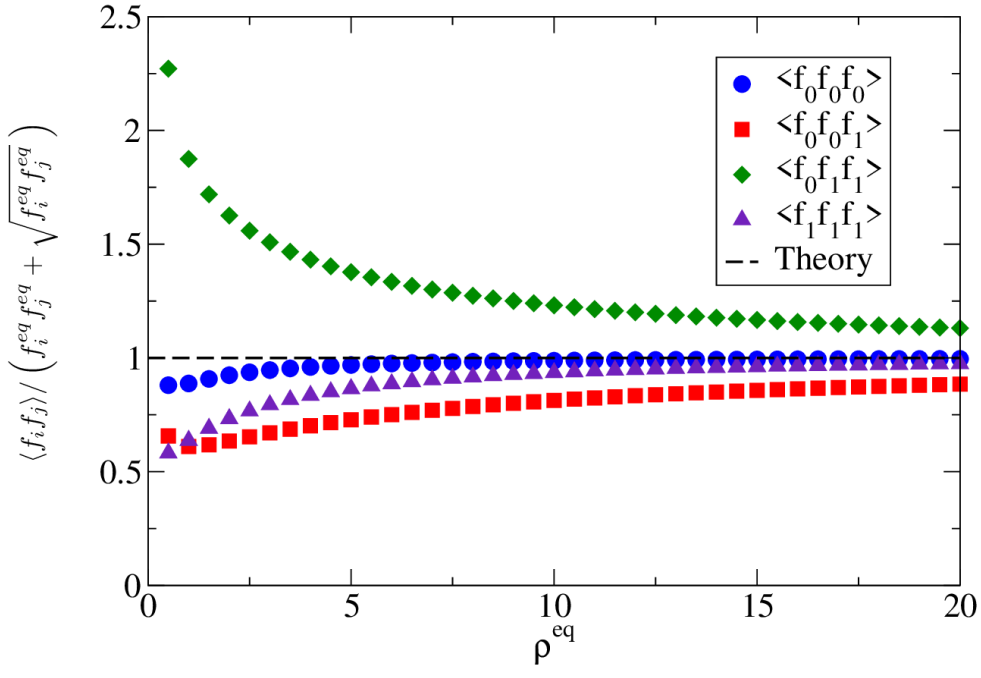
In the previous analysis we have only looked at one third order moment $\langle f_1^3 \rangle$. For completeness we also examined three other third order moments, i.e. $\langle f_0^3 \rangle$, $\langle f_0^2 f_1 \rangle$ and $\langle f_0 f_1^2 \rangle$. Eqn. (A.23) gives this prediction for these third order moments

$$\begin{aligned} &\langle f_i f_j f_k \rangle \\ \stackrel{?}{=} &f_i^{eq} f_j^{eq} f_k^{eq} + f_i^{eq} f_j^{eq} \delta_{ik} + f_j^{eq} f_k^{eq} \delta_{ij} + f_i^{eq} f_k^{eq} \delta_{jk} \\ &+ f_i^{eq} \delta_{ijk} = \zeta_{ijk}. \end{aligned} \tag{2.77}$$

The third moment of the Poisson distribution (a moment that we did not attempt to enforce) shows moderate deviations from the values expected for the Poisson distribution. This is shown in



(a)



(b)

Figure 2.5. Third moment correlators divided by the theoretical prediction ζ of eqn. (2.77) as a function of the mean densities for both global (a) and local (b) noise. The mixed terms ($\langle f_0 f_0 f_1 \rangle$, $\langle f_0 f_1 f_1 \rangle$) have more pronounced deviations from our expected value than the terms examined in Table 2.4.

Fig. 2.5. For larger densities the agreement increases, as might be expected from the central limit theorem, since the Poisson distribution becomes a Gaussian distribution for large average densities.

We conclude here that, surprisingly, the method continues to reproduce the expected fluctuations, even for very low densities, and becomes inaccurate only for average densities of about 3 particles per lattice site on average.

2.4.2. Non-Homogeneous Systems

So far we have only examined systems that are homogeneous and in equilibrium. Next we will look at two systems where the dynamics are non-homogeneous. To make the effect as abrupt as possible we examine two domains with a sharp interface where either the temperature or the mobility is different which is highly relevant for the simulation of phase-separation fronts [64, 65].

First we examined a system where two regions are held at different imposed temperatures θ_1 and θ_2. Here particles will move more quickly in the region of high θ and therefore spend less time in regions of high temperature. Averaging the hydrodynamic eqn. (2.59) we obtain the standard diffusion equation in one dimension

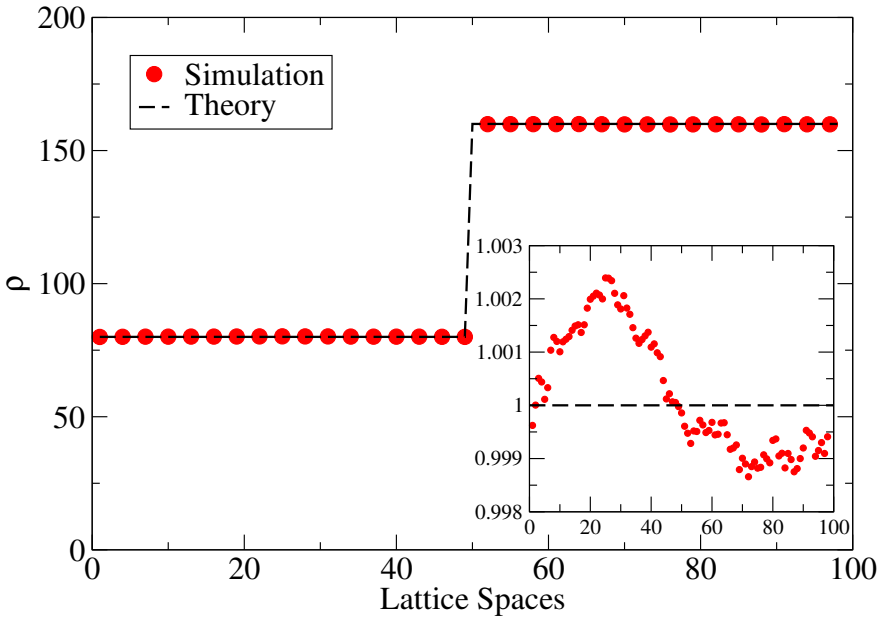
$$\partial_t \rho = \partial_x \frac{D}{\theta} \partial_x (\rho\theta). \tag{2.78}$$

Quantitatively we can look at the stationary solution of this equation which implies that the two different average densities in the two regions are given by

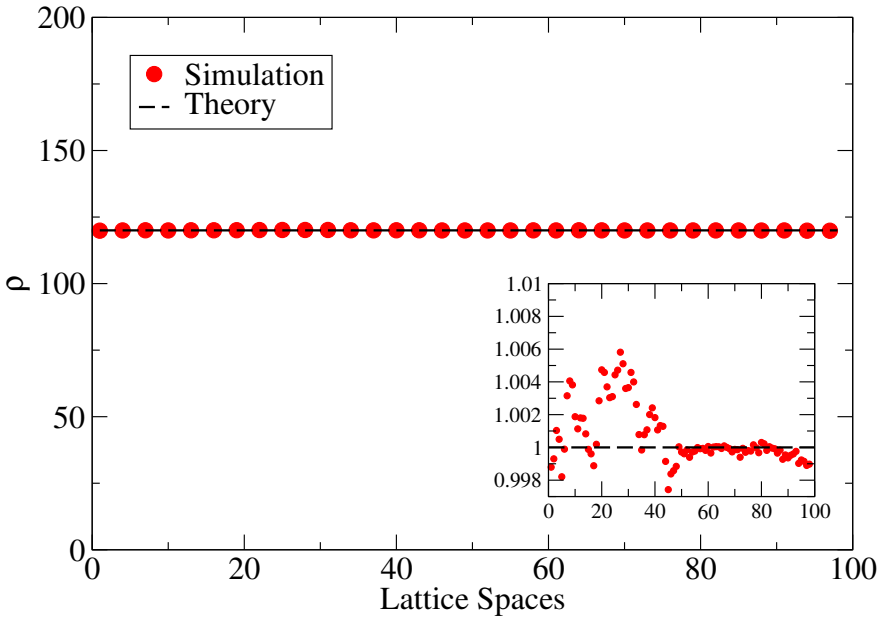
$$\langle \rho_1 \theta_1 \rangle = \langle \rho_2 \theta_2 \rangle. \tag{2.79}$$

This situation is shown in Fig. 2.6a, and we see that the average densities in the two domains are recovered without any noticeable artifacts at the sharp temperature boundary.

The second situation we consider is a system where two regions have different diffusion constants. For this system eqn. (2.78) predicts a constant density. This situation is related to the so-called Ito-Stratonovich dilemma. A single particle in this situation would undergo a random walk, and the likelihood to find it in either region (of equal size) should be the same. However, since the particle is moving more quickly in the region with high mobility the path length inside



(a)



(b)

Figure 2.6. Effect of different mobilities (a) and different temperature (b) in two different regions on the equilibrium behavior of the system. In the case of different temperatures with values $\theta_1 = 1/3$ and $\theta_2 = 1/6$ we find a density difference, whereas the case of different mobilities with diffusion constants $D_1 = 1/30$ and $D_2 = 1/3$ leaves the densities constant. The insets show the numerical solution divided by the analytical solution.

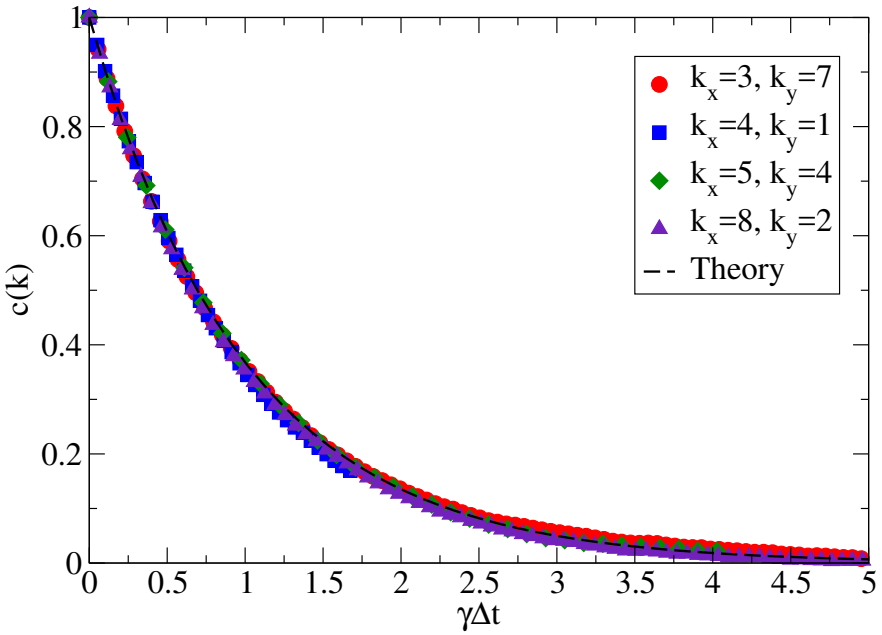


Figure 2.7. Time correlators of eqn. (2.84) for an 81x81 lattice and $\rho^{eq} = 120$, $D = 1/6$ for different k values. γ is the Fourier exponent given by Eqn. (2.85) and $\Delta t = 1$. Values have been averaged over $2 \cdot 10^6$ iterations.

this region has to be longer. This is only possible if there is an preferential reflection at the interface between the two regions into the region with the higher mobility.

The way to modify the diffusion constant in a region is through the current relaxation time τ^j, which now becomes a function of space. The results of Fig. 2.6b show that the expected behavior is recovered by our lattice Boltzmann method. We should also note that the presence of fluctuations is incidental to this problem. The same behavior persists for the deterministic diffusion equation simulated by lattice Boltzmann, but it worthwhile to note that this property is unaffected by the introduction of noise.

2.4.3. Equilibrium Dynamics

A simple heuristic argument for the time-correlation function relies on the Fourier transform of the diffusion equation. To predict the behavior of the discrete system we can start by examining the lattice Boltzmann eqn. (2.1). This representation becomes simple for the special collision operator of $\Lambda_{ij} = 1$, i.e. the case where all relaxation times are one. In that case we get

$$f_i(x + v_i, t + 1) = w_i \sum_i f_i(x, t) + \xi_i(x, t) \tag{2.80}$$

We can translate this equation by $-v_i$ and sum it over i. We then subtract the local density at time t to obtain

$$\begin{aligned}
&\rho(x,t+1)-\rho(x,t)\\
=&\sum_i w_i(\rho(x-v_i,t)-\rho(x,t))+\sum_i \xi_i(x-v_i,t).
\end{aligned} \tag{2.81}$$

For the D2Q5 model employed in our example we can then write a discrete Fourier transform of this equation to calculate the decay times for a particular Fourier mode (neglecting noise terms for now, following Forster's [66] phenomenological approach).

We believe that it is possible to perform a fully consistent analysis of time correlations, only relying on the correlation of equal time $\langle f_i(x,t)f_j(x',t)\rangle = f_i^{eq}f_j^{eq}+f_i^{eq}\delta_{ij}\delta_{xx'}$ by inserting the evolution equation. However, initial investigation of this approach show somewhat lengthy results, that would unreasonably expand this already lengthy paper.

We change our notation and write out the position vector as $\mathbf{x}=(x,y)$, as the results are not fully isotropic, and therefore cannot be written in vector form. We have

$$\begin{aligned}
&\sum_{x,y} e^{i2\pi(\frac{k_x x}{L_x}+\frac{k_y y}{L_y})}[\rho(x,y,t+1)-\rho(x,t)]\\
=&D\sum_x e^{i2\pi(\frac{k_x x}{L_x}+\frac{k_y y}{L_y})}(\rho(x+1,y,t)+\rho(x-1,y,t)\\
&+\rho(x,y+1,t)+\rho(x,y-1,t)-4\rho(x,y,t))\\
=&D(e^{i2\pi\frac{k_x}{L_x}}+e^{-i2\pi\frac{k_x}{L_x}}+e^{i2\pi\frac{k_y}{L_y}}+e^{-i2\pi\frac{k_y}{L_y}}-4)\\
&\sum_{x,y} e^{i2\pi(\frac{k_x x}{L_x}+\frac{k_y y}{L_y})}\rho(x,y,t)\\
=&2D(\cos(\pi\frac{k_x}{L_x})+\cos(2\pi\frac{k_y}{L_y})-2)\\
&\sum_{x,y} e^{i2\pi(\frac{k_x x}{L_x}+\frac{k_y y}{L_y})}\rho(x,y,t).
\end{aligned} \tag{2.82}$$

This is solved by

$$\rho(k_x,k_y,t)=\rho(k_x,k_y,0)e^{2D(\cos(2\pi\frac{k_x}{L_x})+\cos(2\pi\frac{k_y}{L_y})-2)t}. \tag{2.83}$$

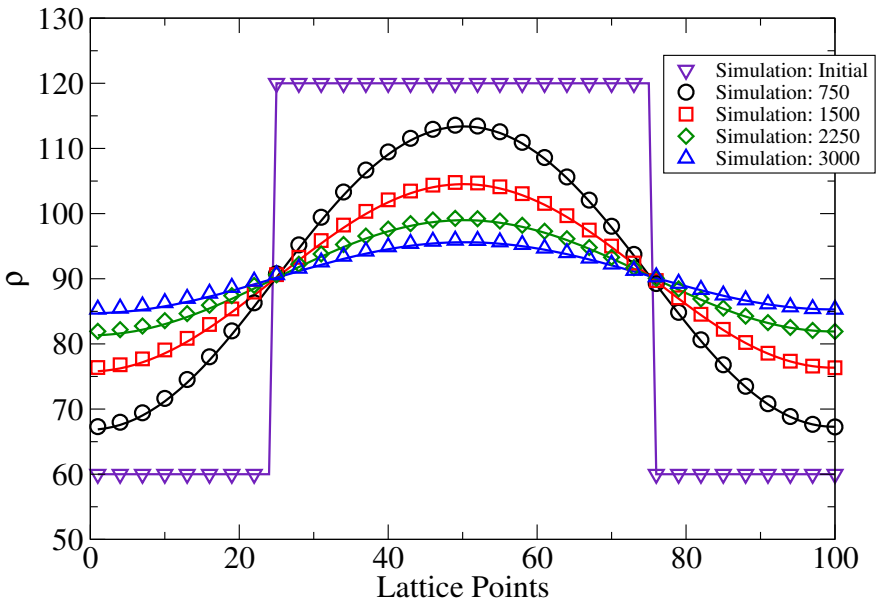


Figure 2.8. Plot of dynamics of diffusion front with initial density values of $\rho_1 = 120$ and $\rho_2 = 20$. There is good agreement between simulation and theory.

We expect the same time behavior for the correlation function

$$c(\mathbf{k}) = \frac{\langle \rho(k_x, k_y, 0)\rho(-k_x, -k_y, t)\rangle}{\langle \rho(k_x, k_y, 0)\rho(-k_x, -k_y, 0)\rangle} \stackrel{?}{=} e^{-\gamma(\mathbf{k})t} \tag{2.84}$$

with

$$\gamma(\mathbf{k}) = 2D\left[2 - \cos\left(2\pi\frac{k_x}{L_x}\right) + \cos\left(2\pi\frac{k_y}{L_y}\right)\right]. \tag{2.85}$$

This behavior is borne out, as is shown for a number of Fourier modes in Fig. 2.7. We performed additional studies with different relaxation times which all gave similarly satisfactory results.

2.4.4. Non-Equilibrium Dynamics

So far we have only examined the behavior of our system in equilibrium situations. The last test presented in this paper now focuses on an example of non-equilibrium behavior. We study a system that initially represents a step function in the x-direction and which is initially the same in the y-direction. It has a step between two densities ρ_1 and ρ_2. One can analytically solve the averaged diffusion equation (2.78) in one dimension on a domain of length L_x with periodic boundary conditions. Then for a constant diffusion constant D for a block of material with density

ρ_1 between $0.25L_x$ and $0.75L_x$ immersed in material with density ρ_2 we obtain the analytic solution

$$\rho^{th}(x,t) = \sum_{i=-\infty}^{\infty} \frac{\rho_2 - \rho_1}{2} \left[\operatorname{erf}\left(\frac{x - (i+0.25)L_x}{2\sqrt{Dt}} \right) - \operatorname{erf}\left(\frac{x - (i+0.75)L_x}{2\sqrt{Dt}} \right) \right]. \tag{2.86}$$

We set up a simulation with $L_x = 100$ and $L_y = 10,000$ and averaged over all y positions. The result of this averaging shows that our numerical solutions are in good agreement with this theoretical solution, as shown in Fig. 2.8.

All this shows, however, is that the effect of the noise averages out as expected. More interestingly we now examined the fluctuations, for which we expect

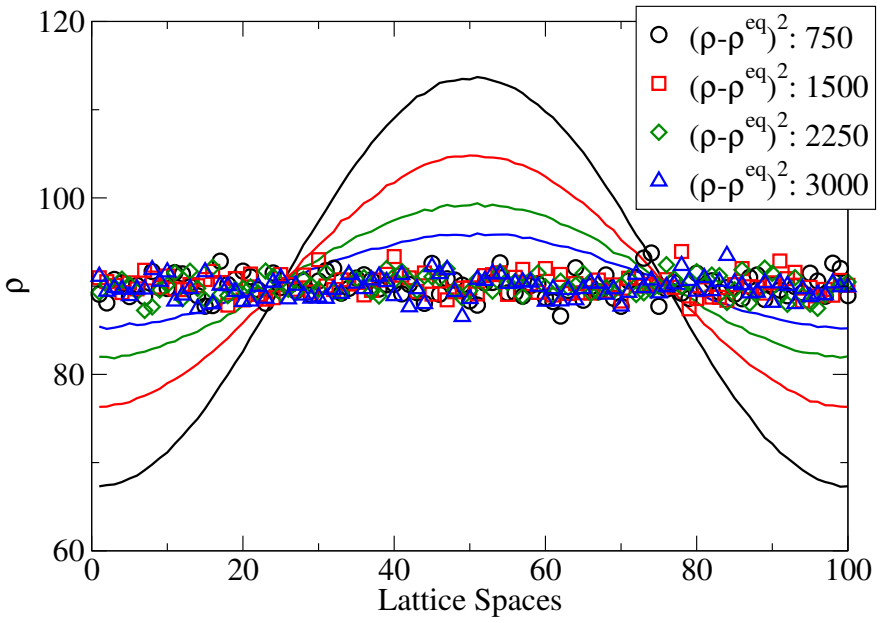
$$\langle (\rho(x,y,t) - \rho^{th}(x,t))^2 \rangle = \rho^{th}(x,t). \tag{2.87}$$

For the global noise implementation, the result is shown in Fig. 2.9a, and the fluctuations are independent of the local density. The key argument for originally proposing local noise implementations given by Kaehler et al. [49] related to such non-equilibrium situations where the local densities can vary significantly. For our local noise implementation we see that this is indeed borne out. This is shown in Fig. 2.9b, where we see excellent agreement with our theoretical prediction of eqn. (2.87).

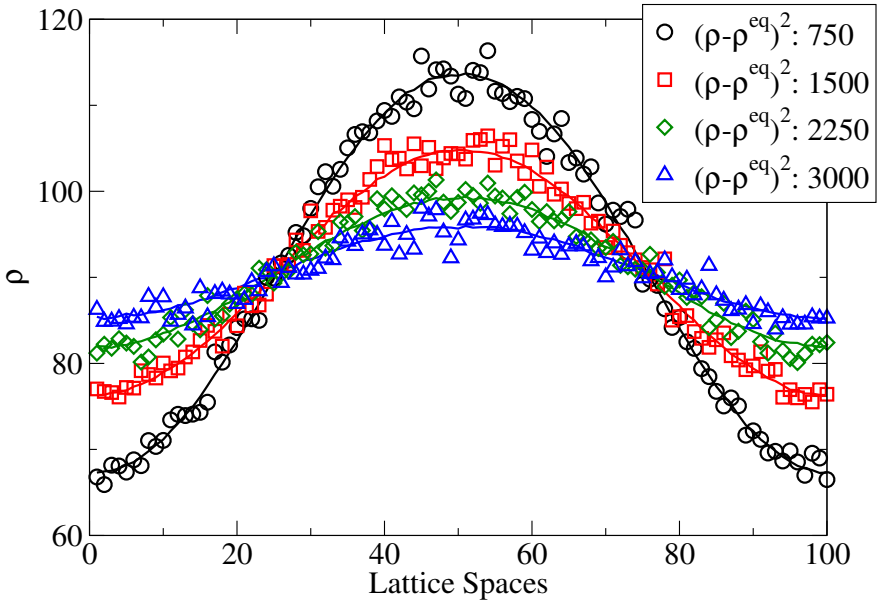
2.5. Conclusion

We presented the derivation of the simplest possible fluctuating lattice Boltzmann method and, for the first time, were able to obtain a fully consistent derivation. The new derivation allowed us to overcome several difficulties in the derivation of earlier fluctuating lattice Boltzmann methods. The lattice Boltzmann method presented in this paper simulates the fluctuating diffusion equation, but we anticipate that we will be able to use the same approach for deriving a fluctuating lattice Boltzmann method for hydrodynamic systems. This is more complex though, because the equilibrium distribution has more terms, making the calculation of eqns. (2.23) and (2.24) more cumbersome.

The consistency of the derivation is reflected in the ability of the new method to recover the expected behavior for all test cases that we have devised. In particular it is remarkable that



(a)



(b)

Figure 2.9. Plot of variance for global (a) and local (b) noise implementaions for density ρ in the form $(\rho - \rho^{eq}) = \langle\rho\rangle$, with initial density domains of $\rho_1 = 20$ and $\rho_2 = 120$. The densities are calculated on a 100x10,000 lattice. The densities are averaged over the y-direction (each x-dimensional lattice space is averaged over 10,000 y values). It is observed that applying noise to ρ^{eq} gives a near constant variance which does not vary in time.

no deviations of the fluctuation relation for any k-modes have been observed. Such deviations are common for other fluctuating methods away from the hydrodynamic regime of small k.

The first derivations of fluctuating lattice Boltzmann methods insisted that the noise amplitude be constant. It was feared that making noise depend on a local density would introduce cumbersome multiplicative noise. Kaehler et al. [49] suggested using a local noise term on physical grounds, but since the derivation required a transition to Fourier space, this non-constant noise amplitude generated insurmountable difficulties in the analysis. The new analysis presented in this paper allowed us to show that using noise amplitudes that depend on a local density is fully consistent and does not generate difficulties normally found with multiplicative noise. As should be expected the difference between using a fixed noise amplitude and one depending on local noise is most pronounced in non-equilibrium systems with large variations in the density.

A bare-bones implementation of this algorithm was published on GitHub [67] and is freely available. This implementation only contains the core algorithm, but for ease of readability the various analysis parts of the code have not been included.

In future work we will show how to extend this approach to non-ideal systems, allowing us to apply this method to examine the dynamics in the vicinity of a critical point and in meta stable systems undergoing nucleation.

3. FOURTH-ORDER ANALYSIS OF A DIFFUSIVE LATTICE BOLTZMANN METHOD FOR BARRIER COATINGS[2]

We examine the applicability of diffusive lattice Boltzmann methods to simulate the fluid transport through barrier coatings, finding excellent agreement between simulations and analytical predictions for standard parameter choices. To examine more interesting non-Fickian behavior and multiple layers of different coatings, it becomes necessary to explore a wider range of parameters. However, such a range of parameters exposes deficiencies in such an implementation. To investigate these discrepancies, we examine the form of higher-order terms in the hydrodynamic limit of our lattice Boltzmann method. We identify these corrections to fourth order and validate these predictions with high accuracy. However, it is observed that the validated correction terms do not fully explain the bulk of observed error. This error was instead caused by the standard finite boundary conditions for the contact of the coating with the imposed environment. We identify a self-consistent form of these boundary conditions for which these errors are dramatically reduced. The instantaneous switching used as a boundary condition for the barrier problem proves demanding enough that any higher-order corrections meaningfully contribute for a small range of parameters. There is a large parameter space where the agreement between simulations and analytical predictions even in the second-order form are below 0.1%, making further improvements to the algorithm unnecessary for such an application.

3.1. Introduction

Coating systems are used heavily in industry for the protection of materials and infrastructure. Common examples include the paints on cars, bicycles, and houses; the layered coating systems used on boats and airframes; and the coatings used to protect bridges. In all cases, the goal

[2]The content of this chapter has been published in "Kyle T. Strand, Aaron J. Feickert, and Alexander J. Wagner. Fourth-order analysis of a diffusive lattice Boltzmann method for barrier coatings. Phys. Rev. E, 95:063311, 2017." Kyle Strand's contributions include developing the fourth-order expansion of the lattice Boltzmann equation, performing Fourier analysis on fourth order corrections, developing code, creating figures, and drafting and revising contents appearing in this chapter.

of the coating system is to protect the underlying substrate from ingress by aggressive particulate, gaseous, or fluid materials while remaining aesthetically intact.

Crosslinked polymer networks, also called thermosets, are typically chosen in protective applications due to their net-like structure. In such a structure, precursor materials are chemically bonded through a crosslinking and curing process to form a three-dimensional structural network. This network acts as a physical and chemical barrier that attempts to prevent permeation by water, salt, particulate matter, and other environmental contaminants. Since substrates are often materials susceptible to corrosion, like aluminum or steel, it is essential that moisture not be permitted to reach the substrate in appreciable quantities.

As is known, most coatings permit, to some degree, moisture ingress [68]. This can be due to imperfections in the preparation process [69], the formation of void space during curing or cooling [70], or because of damage in service. To help detect coating formulations that may be unsuitable for use in the field, accelerated weathering testing is used to determine failure rates and modes in the lab. Much research has been devoted to the relationship between accelerated testing, comparable real-world testing, and service life, but no complete and predictive model exists that accurately correlates a coating's performance in lab testing, performance in field testing, and failure modes or lifespan that is likely to occur in service [71].

Since moisture entering the coating is conserved, the dynamics of fluid density ρ obey the continuity equation

$$\partial_t \rho + \nabla \mathbf{j} = 0, \tag{3.1}$$

where $\mathbf{j}$ is the mass current. Assuming an isotropic coating, mass current will be in the direction of negative density current. We denote the proportionality between the current and negative gradient by D, which in the simplest case is a constant. Later, we consider a more general $D(\rho)$. We therefore have $\mathbf{j} = -D\nabla\rho$. With this constitutive relation for the mass current, we recover the well-known diffusion equation.

Several methods exist to model idealized diffusion. Early work focused primarily on precise mathematical modeling and numerical solutions to boundary-matched differential equations governing diffusion [72, 73]. Modern approaches include network connectivity models [74], Monte Carlo simulations [75], and finite-element analysis [76] for more complex structures like porous

media where an effective diffusivity is desired. However, approaches dealing with pore structures may depend on the structure and porosity of the material in question, quantities that most often unknown *a priori*. Additionally, finite-element models tend to be computationally complex and often rely on commercial closed codes. As a whole, there is comparatively little known about the precise dynamics of diffusion through polymeric coatings [77].

Additionally, different approaches exist when multiple layers are considered. In the case of multiple hydrophobic barrier coatings, boundary-matching Fickian solutions can be used [73] and matched to experiment via electrochemical methods. When a base coating is hydrophilic, as is the case with some primers, an alternative approach couples Fickian diffusion for any overlying barrier coatings with the assumption of an instantaneous reservoir for the base layer [78]. In either case, different coatings in a multi-layer stackup differ in their effective diffusivity.

Any numerical technique used to model the progression of moisture in such a stackup must stably account for a wide range of diffusion constants. Since laboratory testing of candidate barrier coating systems typically includes cyclic exposure to moisture and dry ambient air over long periods of time, simulations of cyclic processes must maintain numerical stability over correspondingly longer time scales.

In this paper, we use lattice Boltzmann numerical techniques to determine the accuracy of modeling moisture ingress through a finite coating system exposed to a reservoir and adhered to an ideal substrate. Because of the necessity of modeling a wide range of saturation levels and diffusivity in the case of multi-layer systems, we analyze the error introduced in the traditional second-order approximation to the diffusion equation used in lattice Boltzmann approaches. To investigate the nature of this error, we introduce a fourth-order correction and perform a Fourier component analysis to confirm the correctness of our results. We show that the bulk of the second-order error in such a system arises from the boundary conditions used, and comment on the proper use of periodic systems to remove this error. Applications to multi-layer systems with variable diffusivity are discussed in the context of our analysis.

3.2. Lattice Boltzmann Methods

The lattice Boltzmann approach models densities f_i defined on a discrete lattice space associated with discrete lattice velocities v_i. After being displaced to a new lattice position $x + v_i$, the densities at each lattice point are redistributed in a collision step. This method has been

used extensively to model hydrodynamic behavior [3, 58, 79], diffusion [9, 51], electrostatics [12], and similar systems with high accuracy and computational efficiency. Notably, the hydrodynamic partial differential equations underlying such systems are not the starting point for the method, but rather emerge from it. Choices like the number of quantities conserved in the collision allow for the freedom of recovering the governing equations for a variety of different systems, as mentioned above.

A popular collision term defines a local equilibrium f_i^0 that only depends on the conserved quantities and then relaxes the actual density towards the local equilibrium. In this form the lattice Boltzmann equation can be written as

$$f_i(x + v_i, t + 1) = f_i(x, t) + \sum_j \Lambda_{ij} \left[f_j^0(\rho(x, t)) - f_j(x, t) \right]. \tag{3.2}$$

Here f_j^0 is the local equilibrium density, Λ_{ij} is a collision matrix, and $\rho(x, t)$ is the local density of the system, given by

$$\rho(x, t) = \sum_i f_i(x, t). \tag{3.3}$$

The form of the collision matrix allows for further control of the algorithm, but this freedom is not explored in this paper. Most examples where this freedom has shown to be useful relate to simulations of hydrodynamic systems with very low viscosity. Such low viscosities may give rise to instabilities that can be controlled by a careful choice of the collision matrix. For diffusive systems like the one considered here, the advantages of utilizing multiple relaxation times are less well established (See Ginzburg [80]), so we will employ the particularly simple collision matrix

$$\Lambda_{ij} = \frac{1}{\tau} \delta_{ij} \tag{3.4}$$

that was originally proposed by Qian [58], using a single relaxation time τ.

It is necessary to impose moments on the equilibrium distribution, following the method of [9]. While not considered here, formulations of this method in the case of multiple components [51] and multiple relaxation times [53, 54]. In particular, we impose the following (non-unique)

moments on the distribution:

$$\sum_i f_i^0 = \rho \tag{3.5}$$

$$\sum_i f_i^0 v_{i\alpha} = 0 \tag{3.6}$$

$$\sum_i f_i^0 v_{i\alpha} v_{i\beta} = \rho\theta\delta_{\alpha\beta} \tag{3.7}$$

where the Greek indices are spatial dimensions and follow the Einstein notation.

Local density conservation is assured by Eqn. (3.5), while Eqn. (3.7) introduces a spatially uniform imposed temperature θ. Following [10], a second-order Taylor approximation using this choice of moments leads to the lattice diffusion equation

$$\partial_t \rho = \nabla_\alpha \left(\tau - \frac{1}{2}\right) \nabla_\alpha(\rho\theta) \tag{3.8}$$

and, if the temperature is constant, this recovers a diffusion equation with the diffusion constant

$$D = \left(\tau - \frac{1}{2}\right)\theta. \tag{3.9}$$

Testing for coating applications usually applies moisture somewhat homogeneously on the sample, either in soak testing or weathering chambers; drying also proceeds evenly. This reduces the problem of interest to an effectively one-dimensional case. For simulations, we use the simplest one-dimensional lattice Boltzmann model with the velocities $\{v_i\} = \{0, 1, -1\}$. This one-dimensional lattice with the given velocities is known as a D1Q3 scheme. For this implementation of a diffusive system, the local equilibrium distribution can be written as

$$f_i^0 = \rho w_i, \tag{3.10}$$

where w_i are the weights related to the magnitude of the velocities $\{v_i\}$. To recover the necessary moments, the weights are

$$\begin{aligned} w_0 &= 1-\theta \\ w_1 &= \frac{\theta}{2} \\ w_2 &= \frac{\theta}{2}. \end{aligned} \tag{3.11}$$

The D1Q3 implementation then allows for a full and self-contained simulation method for a diffusive system.

3.3. Application to Water Content of Coatings

We wish to model the wetting of a single-layer coating via Fickian diffusion. Since coatings are frequently examined in the laboratory on test panels using weathering chambers that subject the coating to moisture, we will consider the case where the coating, represented by a lattice from $0 \leq x \leq L_x$, is exposed to a reservoir of varying concentration $\rho^b(t)$ at $x = 0$ and an impermeable substrate at the right end of the simulation lattice. The meaning of ρ^b is the amount of water that will be absorbed just inside the coating as it is exposed to the environment. For an immersion in water, this corresponds to the maximal water content the coating can absorb, and we scale the density so that this value corresponds to $\rho = 1$.

We must account for these two boundary conditions in our numerical simulation. We implement the source term by setting

$$f_i(0,t) = f_i^0(\rho^b(t)) \tag{3.12}$$

and by replacing the streaming step at the right end by a bounceback algorithm, where the right-moving $f_1(L_x)$ is reinserted as an f_2 in the streaming step. The result for a step function $\rho^b(t) = \Theta(t)$ in the exposure is shown in Fig. 3.1. We used a system with $L_x = 100$ lattice points, $\tau = 1$, $\theta = 0.5$, $\rho_0 = 1$, and ran the simulation for a variable number of iterations T. As expected, moisture is at first located closely to the surface and then penetrates the sample.

To verify the correctness of the simulation results, we construct an analytical solution for the concentration over time, using linear combinations of the well-known error function solution

[72]. These are solutions of the diffusion equation for the initial condition of a step function in an infinite system. If the initial step goes from $2\rho_0$ to zero, then the solution is

$$\rho^{th,1}(x,t) = \rho_0 \left(1 - \text{erf}\left(\frac{x}{\sqrt{4Dt}}\right)\right). \tag{3.13}$$

This solution has a fixed point $\rho_{\text{bath}}(0,t) = \rho_0$ at $x = 0$, which corresponds to our boundary condition. So $\rho^{th,1}(x,t)$ for $x \geq 0$ and $t > 0$ is the analytical solution for an infinite dry coating exposed to a reservoir starting at time $t = 0$. Note that the long-time behavior gives $\rho^{th,1}(x, t \to \infty) = \rho_0$ as expected.

Suppose now that we have a finite one-dimensional coating extending from $0 \leq x \leq L_x$. At $x = 0$, the coating is exposed to a reservoir with fixed concentration $\rho(x = 0, t) = \rho_0$. At $x = L_x$ is an impermeable substrate where $\nabla\rho(x = L_x, t) = 0$.

To account for the vanishing current at the substrate, we use an image source reservoir at $x = 2L_x$. This will ensure a vanishing gradient at $x = L_x$ and, by symmetry, a vanishing current. However, when the reflected concentration becomes nonzero at the reservoir again, we must subtract another image source reservoir at $x = -2L_x$ to maintain the correct boundary condition. Repeating this process infinitely, we arrive at the final solution that includes both reservoir and substrate:

$$\rho^{th}(x,t) = \rho_0 \sum_{i=0}^{\infty} (-1)^i \left[2 + \text{erf}\left(\frac{x - 2(i+1)L_x}{\sqrt{4Dt}}\right) - \text{erf}\left(\frac{x + 2iL_x}{\sqrt{4Dt}}\right)\right] \tag{3.14}$$

For practical purposes, we find ten terms of the infinite sum in Eqn. (3.14) are entirely sufficient for most cases.

It is instructive to determine the correspondence between these numerical parameters and a laboratory case. A typical barrier coating might have thickness $X = 50$ μm, diffusion constant in water $D \sim 10^{-14}$ m^2/s, and be exposed to moisture in a weathering chamber for $T = 4$ hours at a time for testing. We can introduce reduced time, length, and density scales t', x', ρ' such that

$$t = Tt' \tag{3.15}$$

$$x = Xx' \tag{3.16}$$

$$\rho = \rho_0\rho' \tag{3.17}$$

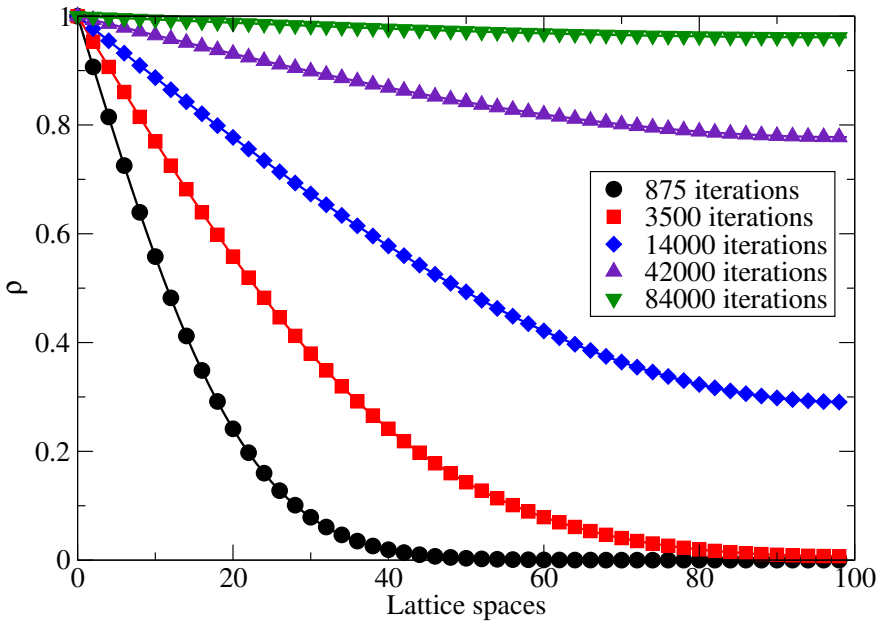


Figure 3.1. Profiles of concentration ρ at $\tau = 1$, $\theta = 0.5$, $\rho_0 = 1$ at various times (symbol), with analytical solution $\rho(x,t)$ (solid line). τ is relaxation time and θ is lattice temperature.

and $0 \leq \{t', x', \rho'\} \leq 1$. Since the unit relationship $T = X^2/D$ holds by dimensional analysis, for any given experimental setup the quantity

$$F \equiv \frac{TD}{X^2} \tag{3.18}$$

is dimensionless and we have the scaled diffusion equation $\partial_{t'} = -\nabla_{x'} F \nabla \rho'$. Using the experimental parameters suggested above gives $F = 5.76 \times 10^{-2}$. In our simulations, we use total length $X = L_x = 100$ lattice sites, reservoir concentration $\rho_0 = 1$, $\theta = 0.5$, and $\tau = 1$. Since this gives a time scale $T \approx 3500$ iterations, this means one hour of equivalent macroscopic exposure corresponds to approximately 875 simulation iterations. Further, the choice of $\tau = 1$ yields immediate relaxation of local distributions, so we would expect excellent agreement to theory.

We are now in a position to comment on the accuracy of this simulation method in comparison to the analytical solution ρ^{th} in Eqn. (3.14). For each of the exposure times in Fig. 3.1, we compute the absolute error

$$\varepsilon(x) \equiv \left| \rho(x,t) - \rho^{th}(x,t) \right| \tag{3.19}$$

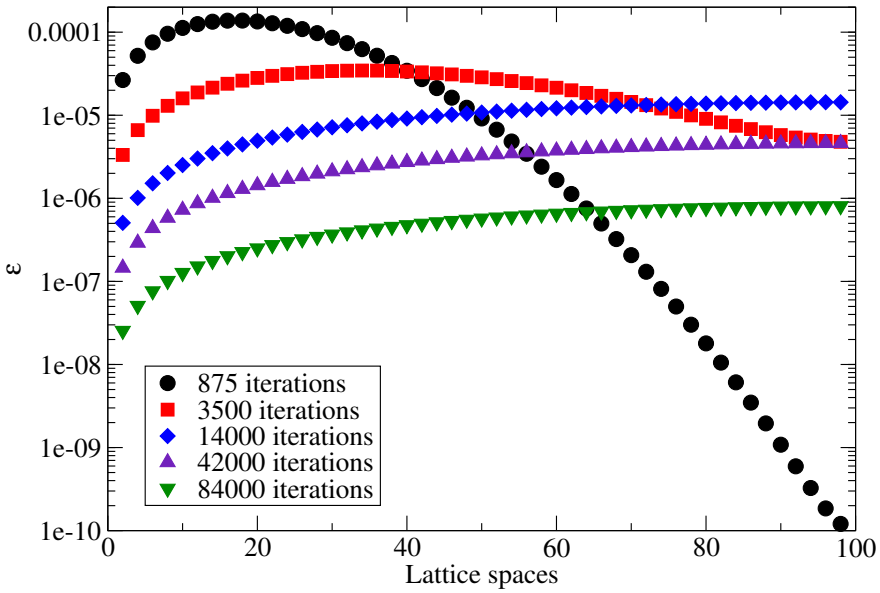


Figure 3.2. Absolute error profile ε between numerical and analytical concentration for exposure over time.

across the lattice space profile. The result is plotted logarithmically in Fig. 3.2, showing excellent agreement. It is interesting to observe how the error changes over time; initially, the error drops substantially since moisture has not yet permeated through the entire coating lattice. This tail increases as the entire lattice becomes wet, but then uniformly decays as the numerical solution approaches saturation and agrees with the corresponding analytical solution.

While this method provides efficient and stable numerical modeling of a single coating, a given coating system might consist of two or more barrier layers in a stackup, each with a different diffusion constant that permits moisture ingress and egress at different rates from its neighbors. To extend this method to the simplest multi-layer case, we might wish to model a two-layer stackup consisting of idealized barrier coatings with different physical properties. To do so, our reservoir model is modified slightly, with the outer barrier coating represented at lattice sites $0 \leq x \leq L_x/2$ and the inner barrier coating at $L_x/2 \leq x \leq L_x$. Since the diffusion constant is controlled by the parameter τ in Eqn. (3.8), the presence of two diffusion constants requires that τ be position-dependent:

$$\tau = \tau(x) \equiv \begin{cases} \tau_{\text{out}} & , \quad 0 \leq x \leq L_x/2 \\ \tau_{\text{in}} & , \quad L_x/2 \leq x \leq L_x \end{cases} \tag{3.20}$$

Table 3.1. Values of τ and θ used in simulations, with corresponding diffusion constant D and time scale T corresponding to four hours of macroscopic equivalent exposure with $F = 5.76 \times 10^{-2}$ (all in lattice units).

τ	θ	D	T
0.55	0.5	0.025	23040
0.70	0.5	0.10	5760
1.0	0.5	0.25	2304
1.5	0.5	0.50	1152
2.0	0.5	0.75	768
10.0	0.5	4.75	121

Incidentally, changing the value of θ between the two regions will lead to different maximum water uptake in the layers, an important relationship that will be explored elsewhere.

Although such a two-layer system is not investigated in this paper, it is essential to determine the range of τ values for which numerical and analytical solutions agree sufficiently over time. For efficient simulations, it is advantageous to choose τ as large as feasible, since this corresponds to a large diffusion constant and hence a shorter simulation time. For a quick initial evaluation, we run a series of lattice Boltzmann simulations with varying values of τ to the same macroscopic equivalent time of four hours of exposure. After that time, we compute the absolute error ε between numerical and analytical solutions across the entire lattice profile. Results are shown in Fig. 3.3. The choices of τ, along with the corresponding time scale T, are shown in Table 3.1.

As shown earlier, the solutions agree very well for $\tau = 1$. However, the error may be orders of magnitude larger for $\tau \neq 1$. Depending on the particular application, we may require ratios of diffusion constants that vary significantly (such as in multi-layer systems); however, the errors indicated here may cause the numerical method to appear less than ideal.

We therefore wish to examine the origin and nature of the τ-dependent error. Of note is that the derivation of the lattice diffusion equation given above (and used heavily in the literature) is done with only a second-order Taylor approximation. To determine the degree to which this approximation leads to the errors shown, we next perform a fourth-order correction to this diffusion equation.

3.4. Fourth-Order Limit of Diffusion Equation

In order to introduce a correction to the diffusion equation, we perform a Taylor expansion of the lattice Eqn. (3.2) to account for higher orders. As shown by Wagner [81], this equation

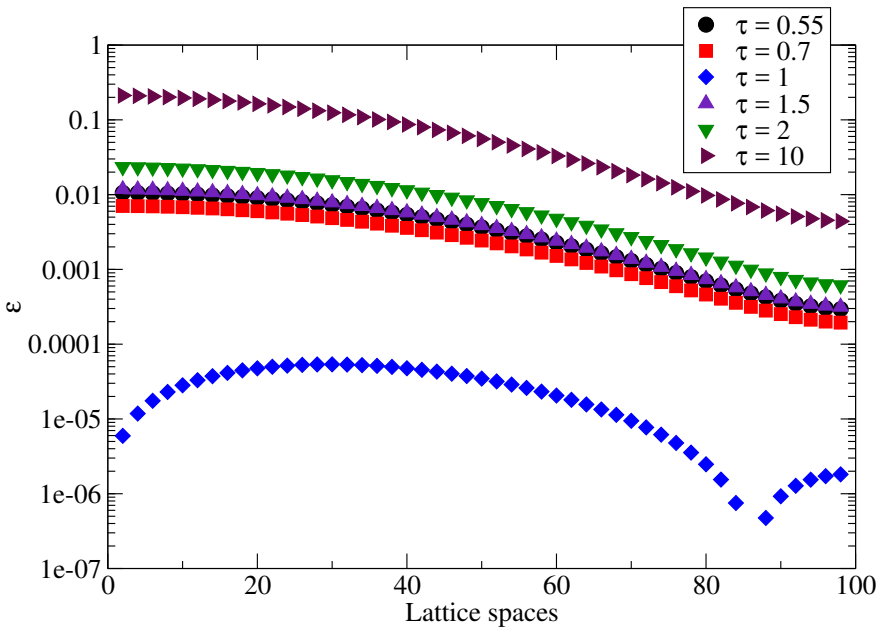


Figure 3.3. Absolute error profile ε between numerical and analytical concentration at various τ. All simulations were run to the same scaled time, corresponding to four hours.

expanded to the fourth order takes the form

$$(\partial_t + v_{i\alpha}\nabla_\alpha)f_i^0 - \left(\tau - \frac{1}{2}\right)(\partial_t + v_{i\alpha}\nabla_\alpha)^2 f_i^0 + \left(\tau^2 - \tau + \frac{1}{6}\right)(\partial_t + v_{i\alpha}\nabla_\alpha)^3 f_i^0 - \left(\tau^3 - \frac{3}{2}\tau^2 + \frac{7}{12}\tau - \frac{1}{24}\right)(\partial_t + v_{i\alpha}\nabla_\alpha)^4 f_i^0 + O(\partial^5) = \frac{1}{\tau}(f_i^0 - f_i). \quad (3.21)$$

Since we have now introduced higher-order powers into this expansion, we must utilize moments up to the fourth-order. Using the form of the equilibrium distribution in Eqn. (3.10), we calculate the higher-order moments:

$$\sum_i f_i^0 = \rho \quad (3.22)$$

$$\sum_i v_{i\alpha} f_i^0 = 0 \quad (3.23)$$

$$\sum_i v_{i\alpha} v_{i\beta} f_i^0 = \rho\theta\delta_{\alpha\beta} \quad (3.24)$$

$$\sum_i v_{i\alpha} v_{i\beta} v_{i\gamma} f_i^0 = 0 \quad (3.25)$$

$$\sum_i v_{i\alpha} v_{i\beta} v_{i\gamma} v_{i\delta} f_i^0 = \frac{\rho\theta}{3}(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) \quad (3.26)$$

Summing over all indices of Eqn. (3.21) using these revised moments, we are left with

$$\partial_t\rho - A(\tau)(\partial_t^2\rho + \nabla_\alpha\nabla_\beta\rho\theta\delta_{\alpha\beta}) + B(\tau)(\partial_t^3\rho + \partial_t\nabla_\alpha\nabla_\beta\rho\theta\delta_{\alpha\beta} + \partial_t\nabla_\alpha\nabla_\gamma\rho\theta\delta_{\alpha\gamma} + \partial_t\nabla_\beta\nabla_\gamma\rho\theta\delta_{\beta\gamma}) - C(\tau)\Bigg(\partial_t^4\rho + \partial_t^2\nabla_\alpha\nabla_\beta\rho\theta\delta_{\alpha\beta} + \partial_t^2\nabla_\alpha\nabla_\gamma\rho\theta\delta_{\alpha\gamma} + \partial_t^2\nabla_\alpha\nabla_\delta\rho\theta\delta_{\alpha\delta} + \partial_t^2\nabla_\beta\nabla_\gamma\rho\theta\delta_{\beta\gamma} + \partial_t^2\nabla_\beta\nabla_\delta\rho\theta\delta_{\beta\delta} + \partial_t^2\nabla_\gamma\nabla_\delta\rho\theta\delta_{\gamma\delta} + \nabla_\alpha\nabla_\beta\nabla_\gamma\nabla_\delta\left[\frac{\rho\theta}{3}(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma})\right]\Bigg) + O(\partial^5) = 0 \tag{3.27}$$

where we have defined the τ-dependent prefactors

$$\begin{aligned} A(\tau) &\equiv \tau - \frac{1}{2} \\ B(\tau) &\equiv \tau^2 - \tau + \frac{1}{6} \\ C(\tau) &\equiv \tau^3 - \frac{3}{2}\tau^2 + \frac{7}{12}\tau - \frac{1}{24} \end{aligned}$$

for brevity.

This form is not particularly useful since there are mixed spatial and temporal derivatives in the higher-order powers. In our one-dimensional implementation, we can drop our indices. We use the diffusion equation to write the temporal derivatives in terms of the spatial derivatives as

$$\partial_t\rho = \left(\tau - \frac{1}{2}\right)\nabla_\alpha^2\rho\theta + O(\nabla^3). \tag{3.28}$$

It immediately follows that

$$\partial_t^2\rho = \left(\tau - \frac{1}{2}\right)^2\nabla_\alpha^2\nabla_\beta^2\rho\theta^2 + O(\nabla^5). \tag{3.29}$$

We can then introduce these two substitutions into Eqn. (3.27) and we have

$$\partial_t\rho - \left(\tau - \frac{1}{2}\right)\nabla^2\rho\theta - \left(\tau - \frac{1}{2}\right)^3\nabla^4\rho\theta^2 + \left(\tau^2 - \tau + \frac{1}{6}\right)\left(\tau - \frac{1}{2}\right)3\nabla^4\rho\theta^2 - \left(\tau^3 - \frac{3}{2}\tau^2 + \frac{7}{12}\tau - \frac{1}{24}\right)\nabla^4\rho\theta = 0. \tag{3.30}$$

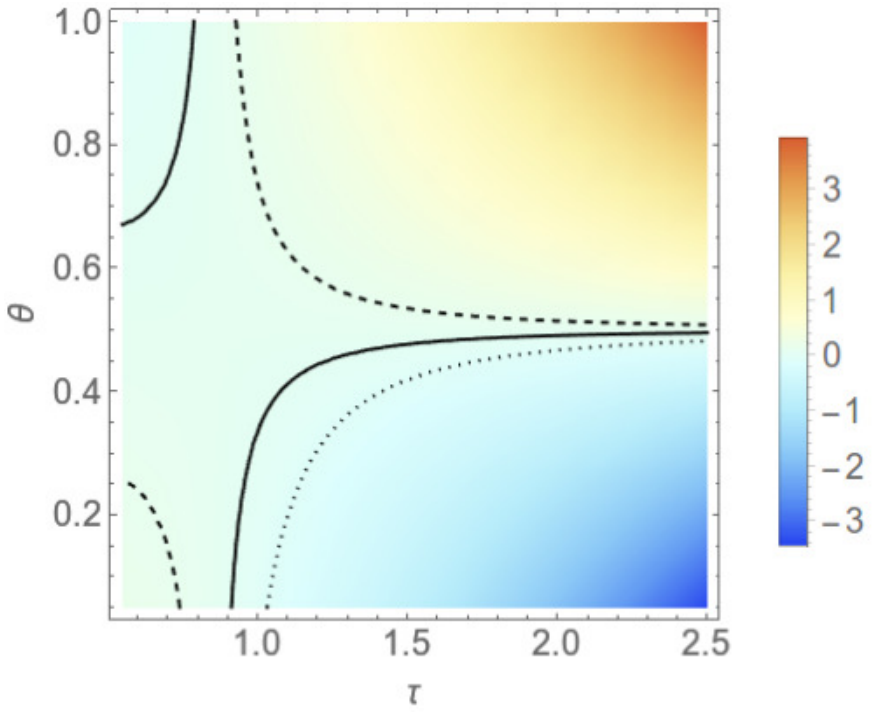


Figure 3.4. Density field representation of $\alpha(\tau,\theta)/D(\tau,\theta)$, with contour lines which indicate a values of $\alpha(\tau,\theta)/D(\tau,\theta) = 0$ (solid), $-1/\pi^2$ (dotted, predicted instability), $1/\pi^2$ (dashed, shown for symmetry). $\alpha(\tau,\theta)$ is the correction term given by Eqn. (3.32), $D(\tau,\theta)$ is the diffusion constant given by Eqn. (3.9), τ is the relaxation time, and θ is the lattice temperature.

We then obtain the form of a corrected diffusion equation

$$\partial_t \rho = D\nabla^2\rho + \alpha\nabla^4\rho \tag{3.31}$$

with corrections up to the fourth power in spatial derivatives, where we define

$$\alpha = \alpha(\tau,\theta) \equiv \left(2\tau^3\theta - \tau^3 - 3\tau^2\theta + \frac{3}{2}\tau^2 + \frac{5}{4}\tau\theta - \frac{7}{12}\tau - \frac{1}{8}\theta + \frac{1}{24}\right)\theta. \tag{3.32}$$

This definition of α represents the expected error between the second-order diffusion equation and the corrected fourth-order equation. For certain parameter values, such as $\tau = 1$ and $\theta = 1/3$, we have $\alpha = 0$, which accounts for higher accuracy observed for such parameters. We plot a density field representation of the relative error quantity $\alpha(\tau,\theta)/D(\tau\theta)$ in Fig. 3.4. We indicate a contour where this quantity vanishes, as well as additional contours whose numerical importance will be explained in later sections.

The correction term has a similar form to a surface tension term in a Cahn-Hilliard equation. In this case, positive values of α would correspond to a negative surface free energy. This implies that simulations with positive α should be unstable for high frequency perturbations. This equation can be solved in Fourier space, allowing us to verify our analytical predictions with lattice Boltzmann simulations. In the subsequent section, we perform this analysis.

3.5. Fourier Analysis of Correction Term

A Fourier transform of Eqn. (3.31) yields

$$\partial_t \widehat{\rho}(k,t,\alpha) = -Dk^2 \widehat{\rho}(k,t,\alpha) - \alpha k^4 \widehat{\rho}(k,t,\alpha). \tag{3.33}$$

Here k is any specific Fourier mode and $\widehat{\rho}(k,t)$ is the k-space density represented by

$$\widehat{\rho}(k) = \frac{1}{2\pi} \int_0^{L_x} \rho(x) e^{\frac{2\pi i k x}{L_x}} dx, \tag{3.34}$$

where L_x is the system size in the x-direction. Even though x is continuous, the finite periodicity of 2π causes k to be discrete. This allows for our system to contain a finite number of k modes which can be now examined independently. The form of Eqn. (3.33) is simple since different k modes do not couple. In k-space, the initial profile at $t = 0$ is chosen by defining $\rho(x, 0)$, which for $\widehat{\rho}(k, 0)$ gives Eqn. (3.34) and

$$\widehat{\rho}(k,t,\alpha) = \widehat{\rho}(k,0) e^{-(Dk^2 t + \alpha k^4 t)}. \tag{3.35}$$

We reproduce the uncorrected diffusion equation by setting $\alpha = 0$, obtaining

$$\widehat{\rho}(k,t,0) = \widehat{\rho}(k,0) e^{-Dk^2 t}. \tag{3.36}$$

These predictions are implemented on a discrete lattice which implies that there will be a finite number of k modes. From Eqn. (3.34), we have

$$k = \frac{2\pi}{L_x} \tag{3.37}$$

which implies a maximum allowed k mode when $k = \pi$ and a minimum lattice dimension of $L_x = 2$. In this finite system, we have the back transform

$$\rho(x,t) = \sum_k e^{ikx}\widehat{\rho}(k,t,\alpha). \tag{3.38}$$

It is now possible to verify this theoretical prediction by examining the decay of specific Fourier modes by imposing an initial profile

$$\rho(x,0) = \sin(kx). \tag{3.39}$$

Using this profile, the uncorrected and corrected k-space densities become, respectively,

$$\begin{aligned}\widehat{\rho}(k,t,0) =& \sin(kx)e^{-tDk^2}\\ \widehat{\rho}(k,t,\alpha) =& \sin(kx)e^{-t(Dk^2+\alpha k^4)}.\end{aligned} \tag{3.40}$$

In practice, we change k by varying the system size L_x. An interesting point to note is that when $\alpha < -\frac{D}{\pi^2}$, it is predicted that the numerical simulations would be unstable. This is predicted due to the fact that in Eqn. (3.40), the negative α term leads to a positive exponent and causes $\widehat{\rho}(k,t,\alpha)$ not to decay.

3.6. Numerical Verification of Correction Term

To determine the validity of the prediction for the correction term shown in Eqn. (3.32), we define a ratio between the two forms of k-space density in Eqn. (3.40) in a simple form such that

$$R(k,t,\alpha) \equiv \frac{\widehat{\rho}(k,t,0)}{\widehat{\rho}(k,t,\alpha)} = e^{\alpha k^4 t}. \tag{3.41}$$

We can use this relation to measure α from numerical simulations. We do this by initializing our probability distributions by $f_i(x,0) = f_i^0\left(\widehat{\rho}(k,0,\alpha)\right)$ and then varying L_x. Our first prediction is that $\ln(R(k,t))$ is a linear function of t. We can find α from the time evolution of the density through

$$\alpha_{\text{exp}} = \frac{1}{k^4}\frac{d}{dt}\ln R(k,t) \tag{3.42}$$

where we numerically calculate the temporal derivative using a finite difference method.

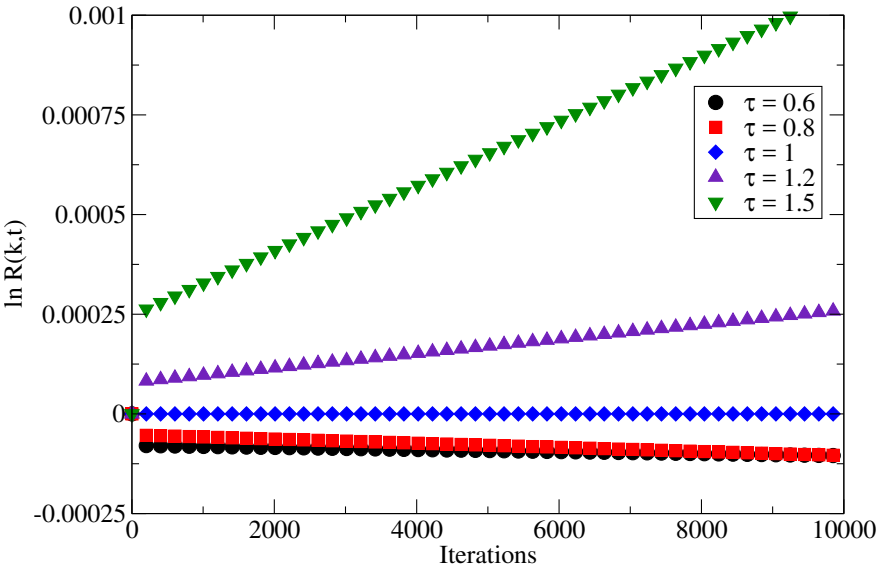


Figure 3.5. The logarithm of ratio of corrected and uncorrected k space densities $\ln R(k,t)$ as a function of discrete time steps for various values of the relaxation time τ and lattice temperature $\theta = 1/3$ and $L_x = 200$. It is observed that there is an initial offset in $\ln R(k,t)$. As the system evolves, we see that the behavior does decay as expected. Since there is this initial offset, we cannot use these early times when calculating the derivative in Eqn. (3.42).

To obtain $R(k,t)$, we initialize a lattice Boltzmann simulation with an initial density of $f_i(x,0) = f_i^0(sin(2\pi x/L_x))$. The numerical evaluation of Eqn. (3.41) using the numerical results is shown in Fig. 3.5. At $t = 0$ we have $R = 1$ by construction, but for all $\tau \neq 1$ we observe a rapid transient change which manifests itself as a near instantaneous jump in Fig. 3.5. After this transient period, the behavior of $\ln(R)$ is indeed linear, as expected. We then calculate

$$\frac{d}{dt}\ln R(k,t) \approx \frac{\ln R(k,t_2) - \ln R(k,t_1)}{t_2 - t_1}, \tag{3.43}$$

where we take t_1 when $\widehat{\rho}(k,t,\alpha) = 0.5$ and t_2 when $\widehat{\rho}(k,t,\alpha) = 0.01$ to avoid any difficulties with the offset. Eqn. (3.42) gives our correction polynomial as a function of any Fourier mode k. Using this form, we can compare our predicted correction term in Eqn. (3.32) to a numerical representation. Fig. 3.6 shows simulation data for $\frac{d}{dt}\left[\ln R(k,t)\right]$ for $\tau = 1$ and $\theta = 0.1$. We see a good fit for all k modes between simulation and the prediction in Eqn. (3.42).

We first test the prediction comparing α_{exp} in Eqn. (3.42) to our theoretical prediction for α from Eqn. (3.32). Fig. 3.7 shows a comparison between α_{exp} and our theoretical prediction for α for various values of τ and θ as a function of L_x. For this analysis, we chose a known stable value

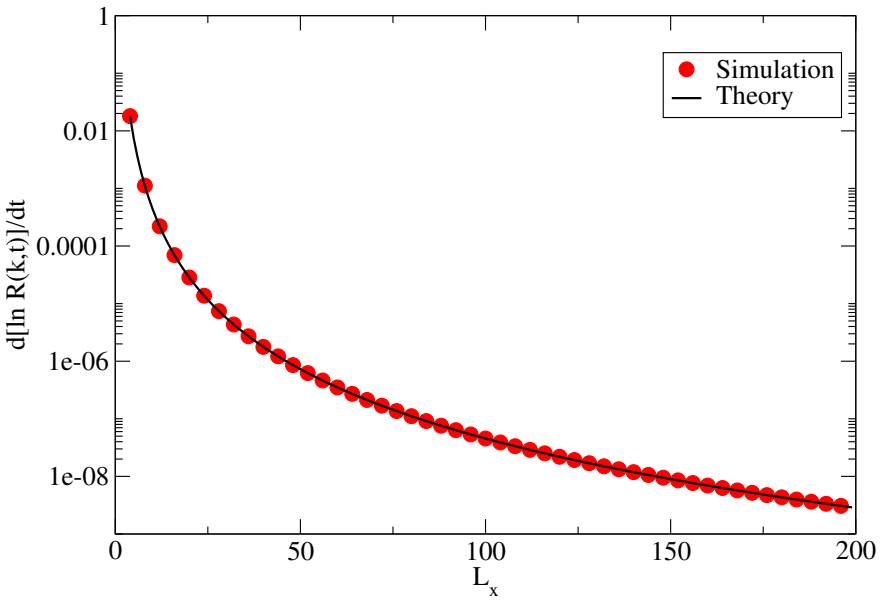


Figure 3.6. Logarithmic representation of $\frac{d}{dt}\left[\ln R(k,t)\right]$ as a function of k from simulation data for $\tau = 1$ and $\theta = 0.1$. Good agreement is observed between the simulation and the curve fit for up to $L_x = 200$.

for either τ or θ and set the other parameter as a more extreme value. For a choice of $\theta = 1/3$, we set $\tau = 0.51$ as the extreme value. In these cases, we see very good agreement between α_{exp} and our prediction. In the cases of $\theta = 1/3$ with $\tau = 1.5$ and $\theta = 0.9$ and $\tau = 1$ we observe good agreement for $L_x > 40$, but as L_x becomes smaller, deviations begin to increase. This suggests that there is a discrepancy in α_{exp} for large k modes.

In the case where $\alpha = 0$, it is interesting to note that the results match a $\frac{1}{k^6}$ rather than the predicted $\frac{1}{k^4}$ fit. This implies that there are additional correction terms which may be relevant at specific values of τ and θ. These higher-order corrections are not considered in the present analysis.

As discussed previously, Eqn. (3.40) predicts numerical instability when $\alpha < -\frac{D}{\pi^2}$. The density representation shown in Fig. 3.4 implies that this will happen as we increase τ and decrease θ to extreme values ($\tau \gtrsim 4$ and $\theta \lesssim 0.3$ simultaneously). A contour showing $\alpha(\tau,\theta)/D(\tau,\theta) = -1/\pi^2$, the start of the region of instability, is shown in that figure.

It is instructive to examine α while holding either τ or θ fixed. Setting $\theta = \frac{1}{3}$, we examine α as a function of τ alone in Fig. 3.8, which shows excellent agreement to theory over 100 independent k modes. We set $\tau = 1$ and examine α as a function of θ alone in Fig. 3.9, with similarly excellent agreement.

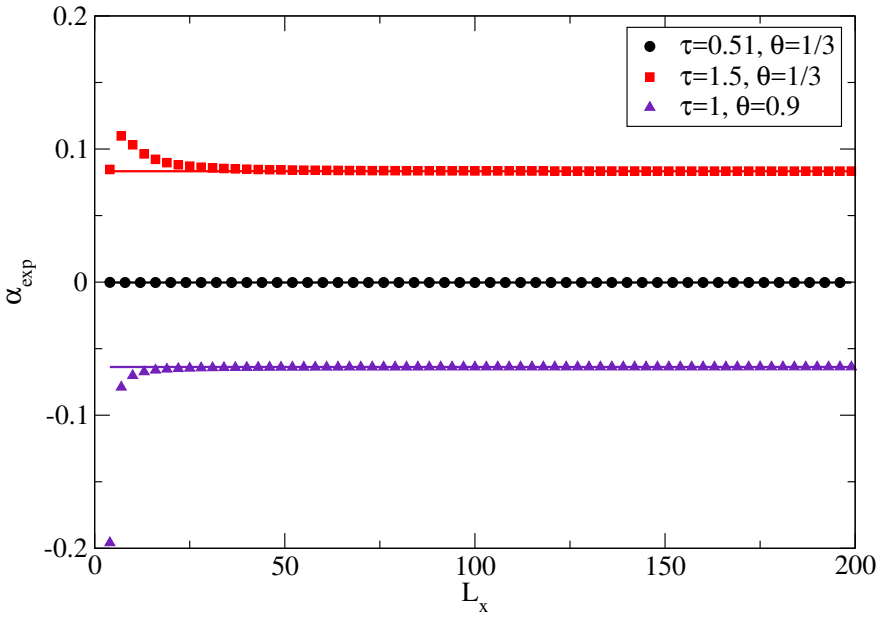


Figure 3.7. Comparison of correction terms α_{exp} (symbol) to theoretical prediction for α from Eqn. (3.32) (solid line) for various values of θ and τ as a function of L_x. It is observed that for $\tau = 0.51$ and $\theta = 1/3$ that α_{exp} matches the theoretical α well for all L_x. For sets of values $\tau = 1$ with $\theta = 0.9$ and $\tau = 1.5$ with $\theta = 1/3$, there is a good match for $L_x > 40$ but deviations are observed for small values of L_x.

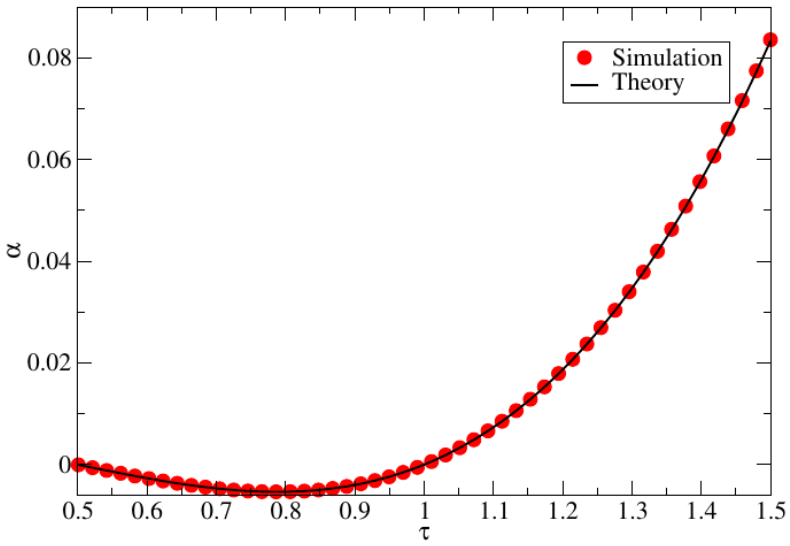


Figure 3.8. Comparison of numerical results and theoretical for correction term α as a function of the relaxation time τ, with a lattice temperature $\theta = \frac{1}{3}$. Results are collected for $L_x = 100$.

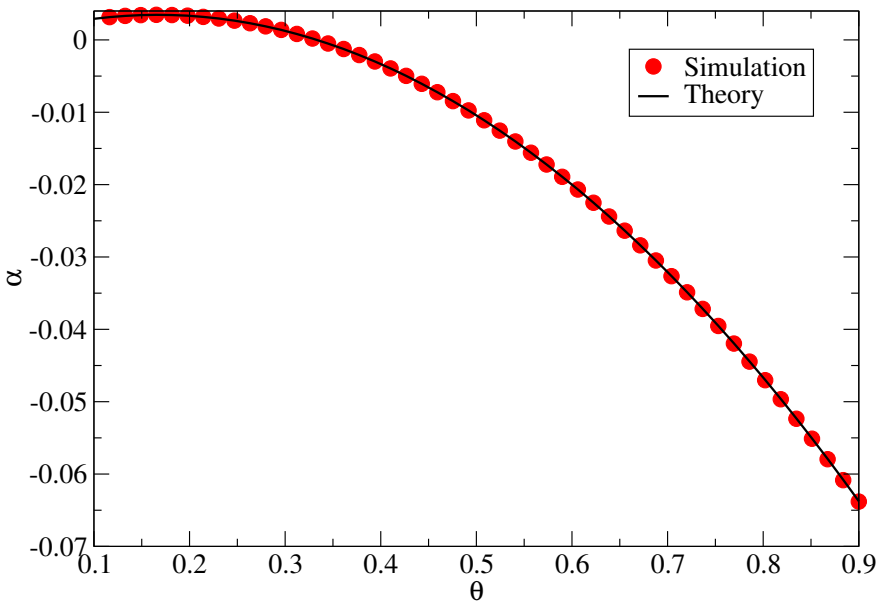


Figure 3.9. Comparison of numerical results and theoretical α of correction term as a function of the lattice temperature θ, with relaxation time $\tau = 1$. Results are collected over 100 independent k modes.

3.7. Application of Correction to Reservoir diffusion

With the fourth-order correction term in hand and its correctness assured, we next determine its applicability to our reservoir coating system. Fig. 3.10 shows the absolute error profile between lattice Boltzmann simulation results and a fourth-order corrected analytical solution. This solution is produced by first setting up an appropriate initial step function

$$\rho(x,0) = \begin{cases} 2 & , \quad L_x < x < 3L_x \\ 1 & , \quad x = L_x \text{ or } x = 3L_x \\ 0 & , \quad \text{else} \end{cases} \tag{3.44}$$

in a periodic lattice. This is entirely equivalent to the boundary conditions implied by the derivation of the second-order error function solution in Eqn. (3.14). We transform this step function into k space via a discrete Fourier transform, use the fourth-order correction to perform a time evolution, and then transform the result back into real space. Strictly speaking, the method of Eqn. (3.14) generates a continuous solution, while the Fourier transform approach yields a discrete solution. We discuss the ramifications of this difference in the Appendix and conclude that the difference

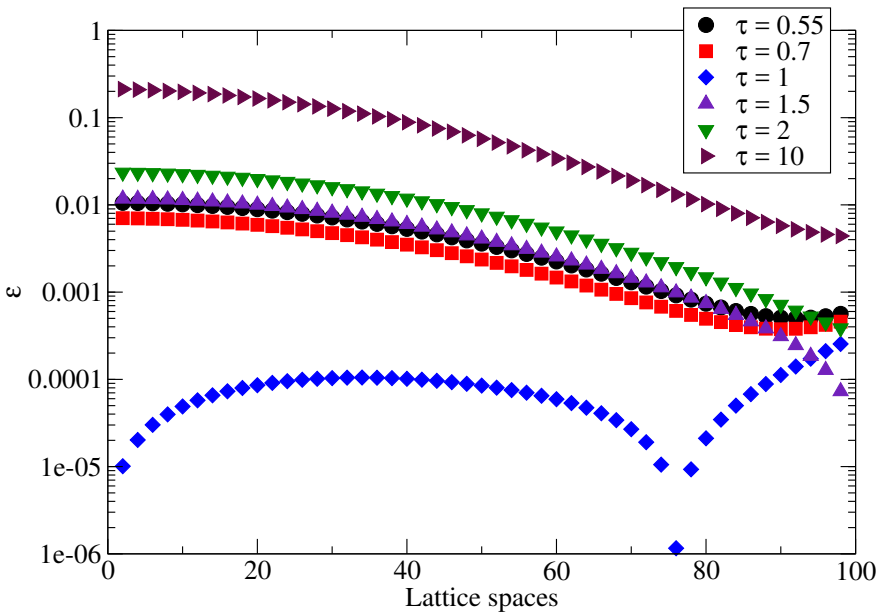


Figure 3.10. Absolute error profile ε between numerical and fourth-order Fourier analytical concentration at various relaxation times τ. All simulations were run to the same scaled time, corresponding to four hours.

in solution discretization is very small and of the same order as the error produced in our best numerical results.

We note no consistent improvement over the second-order shown in Fig. 3.3 from the introduction of the fourth-order correction. However, the magnitude of the error, especially for high values of τ, remains troubling. The nature of the finite simulation lattice is such that the boundaries are treated independently of other lattice sites. In particular, the reservoir density ρ is set manually and not strictly determined by local distributions. Since we have seen that τ-dependent errors tend to accumulate near the reservoir boundary over an order of magnitude higher than at the substrate boundary, the nature of using such a finite lattice is suspect. The case when $\tau = 1$ yielded excellent agreement throughout the finite lattice, but this is consistent with the immediate relaxation of local equilibrium distributions and does not apply to other values of τ.

This τ-dependent error is consistent with the jump observed in Fig. 3.5, where setting $f_i(x,0) = f_i^0(\rho(x))$ led to deviations. Indeed, Eqn. (3.21) implies that

$$f_i = f_i^0 - \tau(\partial_t f_i^0(\rho) + v_{i\alpha}\nabla_\alpha f_i^0(\rho)) + O(\partial^2), \tag{3.45}$$

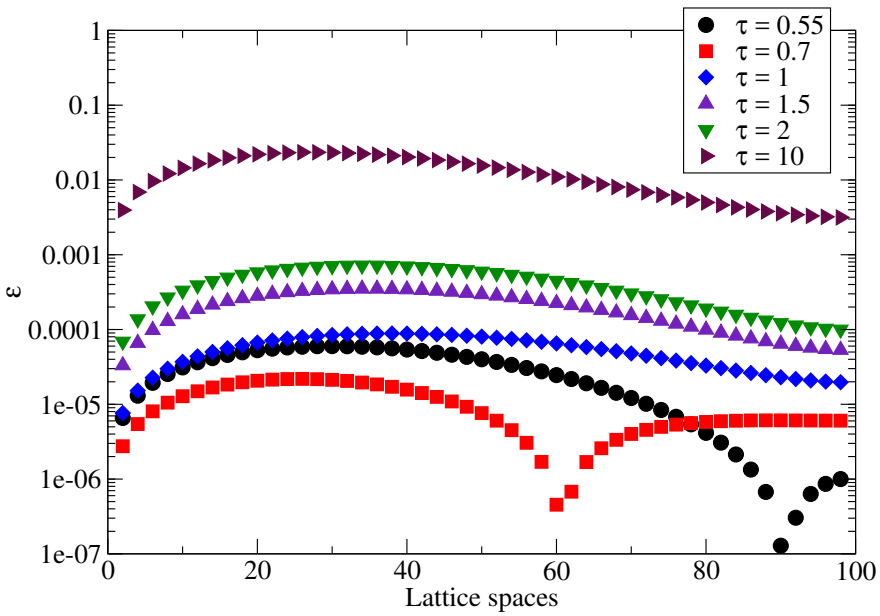


Figure 3.11. Periodic system absolute error profile ε between numerical and second-order Fourier analytical concentration at various relaxation times τ. All simulations were run to the same scaled time, corresponding to four hours.

which suggests an approach that would allow us to increase the accuracy of our boundary conditions.

In our current case, however, we can avoid the cumbersome issue of the boundary condition altogether by simply embedding the system into the periodic lattice used for establishing the initial step function condition of the analytical Fourier solution. This permits a more standard lattice Boltzmann approach that does not rely on manual density adjustment at the reservoir (here at $x = 3L_x$) and uses symmetry to establish the substrate (at $x = 4L_x$) with no bounceback. With this setup, we have removed the need for boundary conditions altogether; we therefore expect that the τ-dependent error should be substantially reduced, especially at the reservoir boundary.

We again run two sets of simulations for our range of τ values to the same scaled time, both using the periodically-embedded lattice simulation. The first set of simulations uses only the traditional second-order approximation and is shown in Fig. 3.11. The second set applies our fourth-order correction and is shown in Fig. 3.12.

As was hoped, the error at the reservoir is reduced by orders of magnitude when compared to the finite system with imposed boundaries. This confirms that the accumulated error from the finite system is due to the presence of boundary conditions that are only guaranteed to match at $\tau = 1$ when relaxation is immediate during collisions. However, contrary to expectation, there is

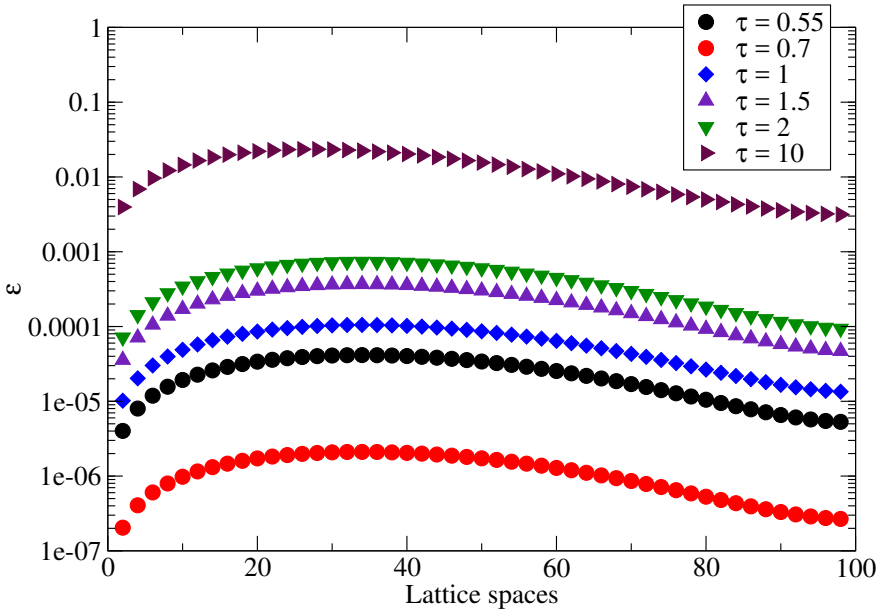


Figure 3.12. Periodic system absolute error profile ε between numerical and fourth-order Fourier analytical concentration at various relaxation times τ. All simulations were run to the same scaled time, corresponding to four hours.

almost no benefit from the fourth-order $\alpha(\tau, \theta)$ correction, even though its validity was verified via Fourier analysis.

It is natural at this point to wonder if there are any choices of parameters τ and θ for which the fourth-order correction provides substantial benefit in our reservoir problem, especially since its use in simulations incurs additional computational burden. Naturally, any such error analysis depends heavily on the particular problem of interest, and therefore on the initial profile and desired time evolution. For our system, we examine the parameter space $0.5 < \tau \leq 2.5$ and $0.1 \leq \theta \leq 1.0$. For each point in this space, we run a lattice Boltzmann periodic reservoir system simulation to the same scaled time. After this time, we compute the ratio

$$\overline{\varepsilon} \equiv \frac{\varepsilon_4(\tau, \theta)}{\varepsilon_2(\tau, \theta)}, \tag{3.46}$$

where

$$\varepsilon_{2,4} \equiv \sqrt{\frac{1}{L_x} \sum_{x=1}^{L_x} \left[\rho(x) - \rho_{2,4}(x)\right]^2} \tag{3.47}$$

is the root mean square error between numerical concentration ρ and second- or fourth-order Fourier analytical concentration $\rho_{2,4}$. If $\overline{\varepsilon} \approx 1$, there is no appreciable correction from using the fourth-order

solution; as $\bar{\varepsilon} \to 0$, the correction becomes more substantial. From a computational perspective, there is a trade-off between the computational burden of the correction and the benefit (if any) from using it. We do not comment on the appropriate balance for any particular situation.

Contrary to expectation, there are no regions of the given parameter space where $\bar{\varepsilon} < 0.9$ during long times, indicating no appreciable benefit to the correction. Further, the fourth-order analysis predicts numerical instability in the bulk region of parameter space where $\alpha < -1/\pi^2$, although the numerical simulations and second-order analysis remain stable. This is a surprising result overall: a fourth-order correction is not only unhelpful in increasing the accuracy of solutions at long times, it is often worse than the second-order approximation and predicts numerical problems incorrectly.

If we instead run the same analysis for a much shorter time (in the equivalent macroscopic system, just 3.5 seconds), the results are more promising and shown in Fig. 3.13. For small values of both τ and θ, the fourth-order correction increases accuracy by an order of magnitude. This is largely due to the fact that the fourth-order theory accurately predicts some early time oscillations at the sharp reservoir interface, as shown by the error reduction in Fig. 3.14. This discussion of higher-order effects gives the rather surprising result that for our barrier coating application, there is no noticeable improvement. This may also arise because even the second-order results are accurate enough that any resulting errors are of the same order of magnitude as the difference between continuous and discrete analytical solutions, as shown in Fig. A.1 in the Appendix.

3.8. Conclusions

In this paper we have examined whether a diffusive lattice Boltzmann method is an effective tool for examining problems related to Fickian water diffusion in barrier coatings. This validation was assisted by our ability to derive an analytical solution for a simple, but not trivial, coatings problem. In Sec. 3.3 we presented a real-space solution for the water content of a dry coating that is initially exposed to a constant moisture reservoir on the surface. A second analytical solution in terms of Fourier components was presented in Sec. 3.5 that can be used both for the standard Fickian diffusion case already examined in Sec. 3.3, as well as the more complex fourth-order diffusion equation we derived as part of a higher-order hydrodynamic limit of the lattice Boltzmann equation. The two equivalent analytical solutions differ slightly because our Fourier series corresponds to a discrete system with only a finite number of Fourier terms.

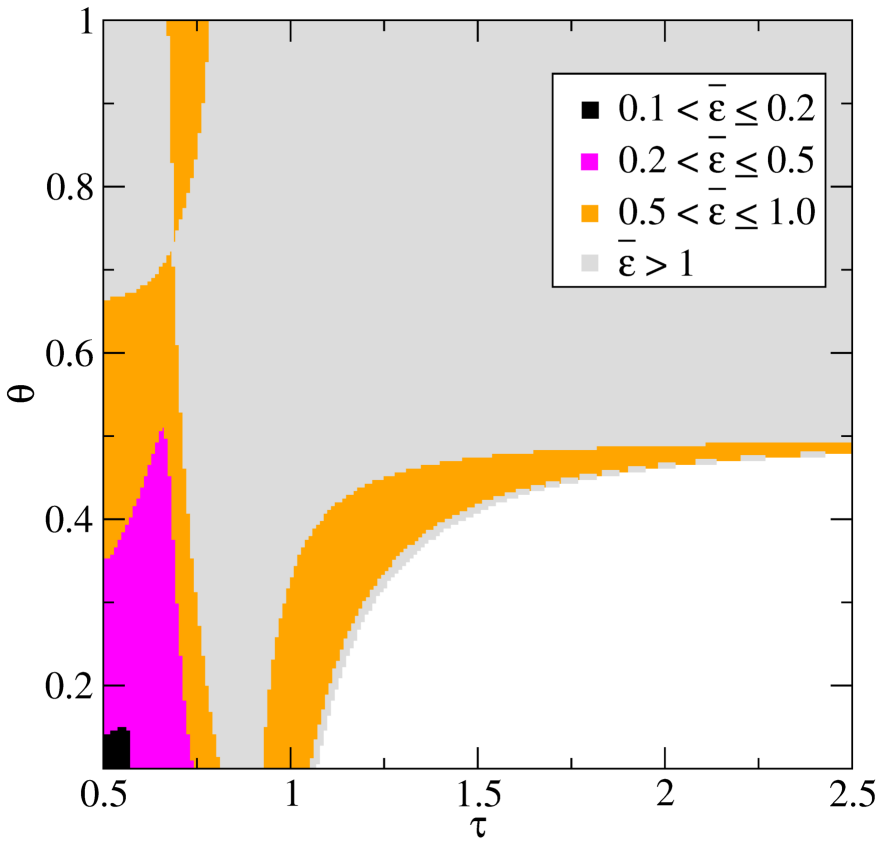


Figure 3.13. Error ratio $\bar{\varepsilon}$, indicating bands comparing the second- and fourth-order Fourier solution accuracy. All simulations were run to the same scaled time, corresponding to a macroscopic system time of 3.5 seconds.

For a simple initial implementation of the inlet boundary, we found excellent agreement only for a relaxation time $\tau = 1$. Our analysis revealed that the disagreement for $\tau \neq 1$ was caused by assuming an equilibrium distribution as the reservoir boundary condition. Eventually we were able to define a "perfect" boundary condition by doing away with the boundary altogether through an embedding of the system in a large periodic system that only requires periodic boundary conditions.

Along the way of our examination, we discovered that we can indeed identify a fourth-order accurate hydrodynamic limit of the diffusion equation. However, this higher-order correction was found to be irrelevant for the coatings problem considered here, as we could only identify a small region in parameter space where the fourth-order predictions were significantly more accurate. This may act as a cautionary tale that validating a higher-order correction does not guarantee that such predictions will always be more accurate for specific applications.

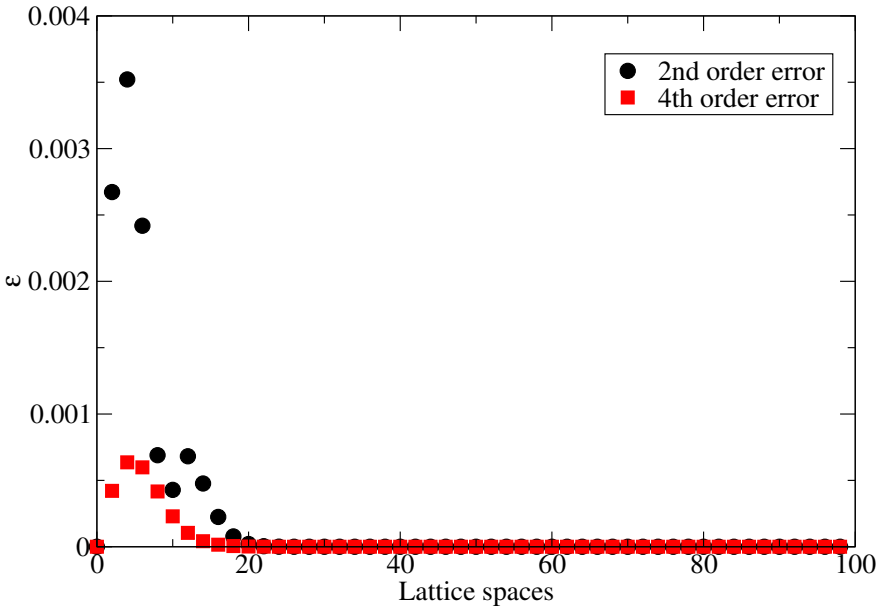


Figure 3.14. Absolute error between numerical simulation results and second-order Fourier (circles) and fourth-order Fourier (squares) analytical solutions. Simulation was run with relaxation time $\tau = 0.55$ and lattice temperature $\theta = 0.15$ to the macroscopic equivalent time of 3.5 seconds.

However, for the best cases, the numerical solutions agree with our analytical solutions almost as well as the two analytical solutions agree with each other, suggesting that the proposed method is indeed an excellent candidate to be applied to coatings problems.

4. OVERRELAXATION IN DIFFUSIVE INTEGER LATTICE GAS[3]

One of the most striking draw-backs of standard lattice gas methods over lattice Boltzmann methods is a much more limited range of transport parameters that can be achieved. It is common for lattice Boltzmann methods to use overrelaxation to achieve arbitrarily small transport parameters in the hydrodynamic equations. Here, we show that it is possible to implement overrelaxation for integer lattice gases. For simplicity we focus here on lattice gases for the diffusion equation. We demonstrate that adding a flipping operation to lattice gases results in a multi-relaxation time lattice Boltzmann scheme with overrelaxation in the Boltzmann limit.

4.1. Introduction

Lattice Boltzmann methods have emerged as a highly successful numerical method for many areas of fluid flow and beyond. However, particularly for fluctuating systems, the discrete nature of earlier lattice gas methods seems much more appropriate, since fluctuations in nature are directly related to the discreteness of matter. Original lattice gas approaches were inferior to lattice Boltzmann methods in several respects. However, recent developments in integer lattice gases by Blommel *et al.* [17] showed that many of the artifacts of traditional Boolean lattice gas methods [3, 82] could be overcome by allowing for integer occupation numbers. One of the remaining shortcomings of the integer lattice gas methods is that the resulting transport coefficients have a more limited range than the transport coefficients that can be a achieved using lattice Boltzmann methods. Lattice Boltzmann methods routinely use a collision operator that overrelaxes the local distributions which allows them to achieve arbitrarily small transport coefficients. This motivated us to further investigate the possibility of achieving overrelaxation in integer lattice gas methods.

Integer lattice gas (ILG), as developed by Blommel *et al.* [17] provided a template for extending traditional lattice gas cellular automaton methods [3, 82]. Traditionally, lattice gases

[3]The content of this chapter for is published in "Kyle Strand and Alexander J. Wagner. Overrelaxation in a diffusive integer lattice gas. Phys. Rev. E 105, L063301, 2022". Kyle Strand's contributions include developing mathematics for flipping operator, implementing flipping operator into the integer lattice gas method, developing code, performing analysis, and drafting and revising content appearing in this chapter.

only allowed a single particle per lattice node, however Chopard *et al.* presented a scheme for multi-particle ILG [15]. Im recent years, we have seen a reemergence of research in practical applications of lattice gas methods [83, 84]. Blommel presented a novel way of allowing any integer number of particles to occupy any node and be consistently re-distributed through binary collisions that conserved both mass and momentum. Although this implementation solved issues such as Galilean invariance which plagued lattice gas methods, it was computationally expensive thus limiting the practical application of such a model. Seekins *et al.* [23], inspired by Boghosian and Chopard *et al.* [34, 15, 36], presented a modification to the collision operator which, instead of following the defined collision rules step-by-step, sampled a probability distribution to arrive at the same results as presented in Blommel's original model.

The success of the Seekins' sampling collision operator in terms of computational practicality sheds new light on what could be achieved using ILG methods. One key draw-back of lattice gas methods is that they do not appear to allow for overrelaxation of the collision operator. Such overrelaxation is routinely used in lattice Boltzmann methods to achieve lower transport coefficients. This is particularly helpful for hydrodynamic simulations at high Reynolds numbers, which are helped significantly if low viscosities are possible. The idea of overrelaxation is that instead of local collisions moving a local distribution closer to local equilibrium the effect is to over-shoot this approach and land beyond the equilibrium distribution. Such overrelaxation has been shown by Bösch and Karlin to be disconnected from the standard kinetic theory domain from which lattice Boltzmann methods are typically derived [85]. More recently Pachalieva *et al.* was able to show that overrelaxation can also be obtained by a simple coarse-graining of Molecular Dynamics simulations [86], giving a more direct link between overrelaxation and a physical system. This inspired us to question the generally held idea that overrelaxation is not possible in lattice gas methods, and in particular are revisiting this question for integer lattice gas methods.

In this manuscript, we present a simple and effective method which successfully performs overrelaxation in a diffusive integer lattice gas. We extend the sampling collision operator presented by Seekins to incorporate overrelaxation through a simple permutation of particles during the collision. In section 4.2, we introduce the basics of the integer lattice gas method. Section 4.3 provides the necessary extensions for implementing overrelaxation in ILG methods. We derive the Boltzmann average for the system in section 4.4 showing its correspondence with lattice Boltzmann

methods and we show that we can derive the diffusion equation from the equation of motion for the overrelaxed integer lattice gas. In section 4.5, we verify the validity of our refined collision operator incorporating overrelaxation.

4.2. Integer Lattice Gas

We review here briefly the integer lattice Boltzmann method of Seekins *et al.* [23]. It consists of an underlying regular lattice where neighboring lattice points are connected through lattice velocities $\{v_i \Delta t\}$. With each of these lattice velocities at each lattice point we associate integer occupation numbers $n_i(x,t)$ that evolve through the lattice gas evolution equation

$$n_i(x + v_i \Delta t, t + \Delta t) = n_i(x,t) + \Xi_i(\{n_i\}). \tag{4.1}$$

Here Ξ_i is a collision operator that redistributes the particles at each lattice site. This collision operator is stochastic by nature and must obey all local conservation laws. In our case only mass is conserved, so we require

$$\sum_i \Xi_i = 0. \tag{4.2}$$

Seekins *et al.* [23] introduced a collision operator that picked a fraction ω of particles at random and redistributed them to occupation number n_i with a probability w_i, where the w_i are the familiar weight functions used in the definition of lattice Boltzmann equilibrium distributions [87]. The details of the algorithm that allows this to be done efficiently will not be discussed here, but are detailed in the publication cited above.

For this lattice gas we can derive a lattice Boltzmann average through

$$f_i(x,t) = \langle n_i(x,t) \rangle, \tag{4.3}$$

where the average $\langle \cdots \rangle$ implies a non-equilibrium average over all possible realizations of the stochastic lattice gas. The same average is applied to the lattice gas collision operator to obtain the lattice Boltzmann collision operator

$$\Omega_i = \langle \Xi_i \rangle. \tag{4.4}$$

This lattice Boltzmann collision operator was shown to be of the form

$$\Omega_i = \omega(f_i^0 - f_i) \tag{4.5}$$

where the local equilibrium distribution is given by

$$f_i^0(x,t) = \rho(x,t)w_i \tag{4.6}$$

with the local density

$$\rho(x,t) = \sum_i f_i(x,t). \tag{4.7}$$

The resulting lattice Boltzmann equation

$$f_i(x + v_i\Delta t, t + \Delta t) = f_i(x,t) + \omega[f_i^0(x,t) - f_i(x,t)], \tag{4.8}$$

can then be shown to have the diffusion equation as its hydrodynamic limit:

$$\partial_t \rho(x,t) = \nabla D \nabla \rho(x,t), \tag{4.9}$$

where the diffusion constant is given by

$$D = \left(\frac{1}{\omega} - \frac{1}{2}\right)\theta, \tag{4.10}$$

with

$$\theta = \sum_i w_i v_i^2. \tag{4.11}$$

In lattice Boltzmann simulations that are used as numerical methods in their own right, values of $\omega \in \{0, 2\}$ are routinely used, but in lattice gas implementations the definition of ω as a probability limits its range to $\omega \in \{0, 1\}$. This limits the usefulness of lattice gas methods compared to their lattice Boltzmann counterparts.

4.3. Overrelaxation in a Lattice Gas

The contribution of this paper is to show that it is indeed possible to construct lattice gas methods that can access the $\omega \in \{1, 2\}$ range often used in lattice Boltzmann approaches. This is

of interest not only for the diffusive systems considered here, but also for hydrodynamic systems, which are the main forte of lattice Boltzmann methods. In those systems the transport coefficient of interest is the viscosity, and obtaining low values for the viscosity is essential for simulations of systems with high Reynolds numbers. In this manuscript we present a proof of principle that we hope to extend to those even more important hydrodynamic LB models in the near future.

The range of $\omega \in \{1, 2\}$ is referred to as overrelaxation because the lattice Boltzmann collision operator will overshoot the local equilibrium distribution in the relaxation process. $\omega = 1$ corresponds to full relaxation, where the distributions reach local equilibrium in each step, and $\omega < 1$ implies under-relaxation. That there is a difficulty of deriving overrelaxation from, say, a continuous Boltzmann equation was shown by Bösch *et al.* [85]. But aside from its obvious practical utility it has now been shown by Pachalieva *et al.* [86] that overrelaxation in lattice Boltzmann can also be obtained by coarse-graining Molecular Dynamics simulations.

As indicated by Bösch *et al.* overrelaxation cannot be obtained through a continuous extension of the collision process. We propose here to augment the collision process with a flipping operation such that

$$F_i(n_i) = n_{-i}. \tag{4.12}$$

where we interpret negative indices such that $v_{-i} \to -v_i$. This leads to the lattice gas evolution equation

$$n_i(x + v_i\Delta t, t + \Delta t) = F_i(n_i) + \Xi_i(\{F_i(n_i)\}). \tag{4.13}$$

Heuristically such a flipping operation will send particles back along the direction they just came from, and it is reasonable to expect that this operation will completely suppress diffusion. It is therefore reasonable to expect that this operation on its own will lead to something resembling $\omega = 2$, *i.e.* full overrelaxation. A closer examination of the collision operator Eqn. 4.5 shows that this is not the full story as we will examine below.

This flipping operation is then augmented with an additional collision with a collision fraction $\omega^* \in \{0, 1\}$. In the limiting case of $\omega^* = 0$, we only apply the flipping operation, leading to the case of full overrelaxation, and the limiting case of $\omega^* = 1$ means that all particles are re-distributed, making the flipping operation moot and leading to full relaxation. In the next section we will derive

the Boltzmann limit of this augmented lattice gas and show that it indeed corresponds to a lattice Boltzmann method with overrelaxation.

4.4. Boltzmann Approximation

We will now derive the Boltzmann average of the lattice gas including the flip operator of Eqn. (4.13). This is most easily accomplished by separating the LB operation into a collision and a streaming step and then transform the collision term into a moment space where the flip-operation has a very simple interpretation. We obtain

$$f_i(x + v_i\Delta t, t + \Delta t) = F_i[f_i(x,t)] + \omega\{f_i^0(x,t) - F_i[f_i(x,t)]\}. \tag{4.14}$$

Following the procedure presented by Wagner *et al.* [39], we transform the distribution functions into moment space by defining a transformation matrix m_i^a with which we obtain occupation numbers in moment space

$$M^a = \sum_i m_i^a f_i. \tag{4.15}$$

The transformation matrix is orthogonal with respect to the Hermite norm defined through

$$\sum_i m_i^a w_i m_i^b = \delta^{ab}, \tag{4.16}$$

$$\sum_a m_i^a w_j m_j^a = \delta_{ij}. \tag{4.17}$$

This allows us to obtain the f_i from the M^a through

$$f_i = \sum_a w_i m_i^a M_a. \tag{4.18}$$

When designing a transformation matrix it is customary that the first moments should correspond to the conserved quantities, the following moments to the hydrodynamic quantities, and the remainder will represent so-called ghost modes, i.e. quantities that do not enter the hydrodynamic limit.

For a simple one dimensional model with three velocities $\{v_i\} = \{-1, 0, 1\}$ (D1Q3) this transformation matrix is written as [39]

$$m_i^a = \begin{pmatrix} 1 & 1 & 1 \\ -\sqrt{\frac{1}{\theta}} & 0 & \sqrt{\frac{1}{\theta}} \\ \sqrt{\frac{1-\theta}{\theta}} & -\sqrt{\frac{\theta}{1-\theta}} & \sqrt{\frac{1-\theta}{\theta}} \end{pmatrix}, \tag{4.19}$$

where θ was defined in Eqn. (4.11). It is useful to give the moments M^a separate names related to their physical significance:

$$M^a = \begin{pmatrix} \rho \\ j \\ \Pi \end{pmatrix}, \tag{4.20}$$

where ρ is the particle density, j is the current density, and Π is related to the energy density moments. A particularly nice property of this transformation matrix is that (in general) the value of the non-conserved quantities of the equilibrium distribution in moment space are zero [49]:

$$M^{a,0} = \sum_i m_i^a f_i^0 = \begin{pmatrix} \rho \\ 0 \\ 0 \end{pmatrix}, \tag{4.21}$$

We can now separate out the effect of the flipping operator and the collision process. The flipping operator has a very simple representation in moment space:

$$F(M^a) = \begin{pmatrix} \rho \\ -j \\ \Pi \end{pmatrix}, \tag{4.22}$$

and in general all even velocity moments are unaffected by the flipping operation and all odd velocity moments acquire a negative sign. The effect of the collision is likewise simple: conserved quantities are unaffected and non-conserved quantities are multiplied by the fraction ω. So we can

write the effect of the collision operator in moment space as

$$M^a = M^a + \Omega^a = \begin{pmatrix} 0 \\ -(1-\omega)j \\ (1-\omega)\Pi \end{pmatrix}. \tag{4.23}$$

This is equivalent to defining a new

$$\omega^j = 2 - \omega, \tag{4.24}$$

so that we get the more usual

$$M^a = \begin{pmatrix} 0 \\ (1-\omega^j)j \\ (1-\omega)\Pi \end{pmatrix}. \tag{4.25}$$

With this we can write the lattice Boltzmann equation corresponding to the integer lattice gas with the flipping operation as

$$f_i(x + v_i\Delta t, t + \Delta t) = \sum_a w_i m_i^a (1-\omega^a) m_j^a (f_j^0 - f^j), \tag{4.26}$$

where we introduced the ω^a notation to refer to $(\omega^\rho, \omega^j, \omega)$ and ω^ρ is arbitrary. The key result is that this has the form of a standard MRT lattice Boltzmann equation. In the case where $\omega^j \in [1, 2]$, overrelaxation is observed, which achieves the primary goal of the flipping operation, $F(M^a)$.

According to the derivation of the hydrodynamic limit of the lattice Boltzmann equation in Eqn. (4.26) (see *e.g.* Kaehler *et al.* [60]), we obtain the diffusion equation

$$\partial_t \rho = -D\nabla^2 \rho, \tag{4.27}$$

in which we defined a diffusion constant

$$D = \theta\left(\frac{1}{\omega^j} - \frac{1}{2}\right). \tag{4.28}$$

As shown by Sorenson *et al.* [88], the evolution of the densities follows the diffusion equations for features with wavelength $\lambda \gtrapprox 10\pi/\omega$. In the following we choose $\lambda = 320$ which is well in this regime.

4.5. Results

In order to verify that this diffusive implementation does replicate Eqn. (4.27) in the hydrodynamic limit, we analyze a system with a known analytic solution [17, 23]. We impose the system with a sine wave as the density profile which takes the form

$$\rho(x,0) = N^{ave}\left[1 + \sin\left(\frac{2\pi x}{L}\right)\right]. \tag{4.29}$$

Here, N^{ave} is the average number of particles which exist at each node on the lattice and L is the size of the one-dimensional lattice. The time-evolution of this system has the analytical solution

$$\rho(x,t) = N^{ave}\left[1 + \sin\left(\frac{2\pi x}{L_x}\right)\exp\left(-\frac{4\pi^2 Dt}{L^2}\right)\right] \tag{4.30}$$

$$= N^{ave} + A^{th}(t)\sin\left(\frac{2\pi x}{L}\right), \tag{4.31}$$

where we have a definition for the decay of the amplitude

$$A^{th}(t) = N^{ave}\exp\left(-\frac{4\pi^2 Dt}{L^2}\right). \tag{4.32}$$

An issue arises due to the fact that Eqns. (4.29-4.32) are continuous, the ILG methods are discrete by nature, and $n_i \in \mathbb{Z}$. To remedy this, we can impose our initial density profile which is from a sinusoidal probability distribution

$$P(\rho) = N^{ave}\left[1 + \sin\left(\frac{2\pi x}{L}\right)\right]. \tag{4.33}$$

This can be performed by choosing Poisson distributed random numbers for the occupation numbers with an expectation value $w_i P(\rho)$ based on the weighting of the system [23]. This method allows us to model continuous functions as fully integer valued which properly aligns with the discrete nature of the ILG methods. In the same manner as Blommel, we are able to acquire the amplitude

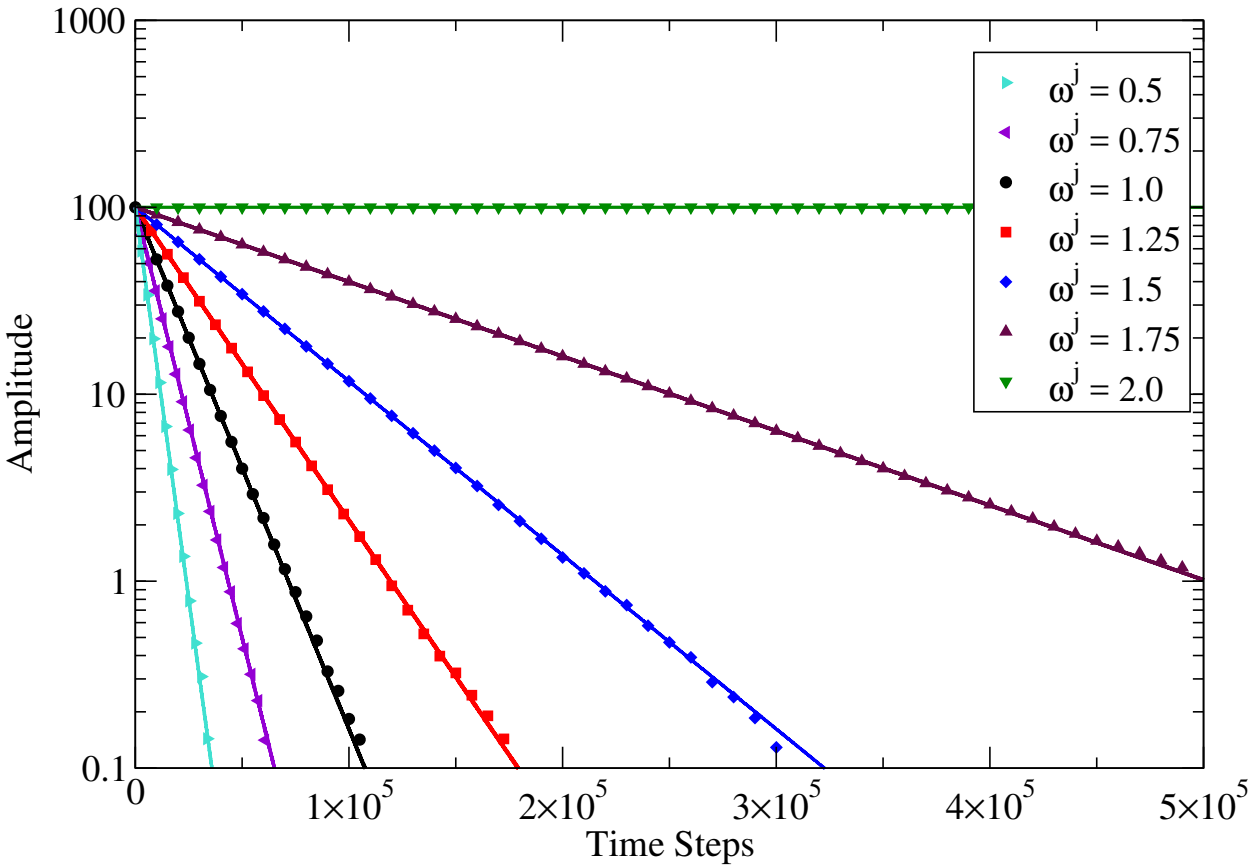


Figure 4.1. Decay of the amplitude of an initial sine wave with varying ω^j values. $\omega^j > 1$ is in the overrelaxed regime. We see good agreement between the measured simulation data (symbols) and the theoretical prediction (solid lines) from Eqn. (4.32) both inside and outside the overrelaxation regime. This data was a result of the average of 500 individual simulations on a D1Q3 lattice with size $L = 320$ and $N^{ave} = 100$.

of the profile at any point by

$$A^{LG}(t) = \frac{\sum_x \sin\left(\frac{2\pi x}{L}\right) N(x,t)}{\sum_x \sin^2\left(\frac{2\pi x}{L}\right)}. \tag{4.34}$$

This measured amplitude can then be compared to the theoretical prediction in Eqn. (4.32). We illustrate this comparison in Figure 4.1 where we see excellent agreement between the measured amplitudes and the theoretical prediction for the decay of the sinusoidal profile. For the values $1 < \omega^j < 2$ in the overrelaxation regime, we find very good agreement between the measured simulation and the theoretical prediction. The values without the flipping operation had previously been verified by Seekins for $\omega \leq 1$ and are shown for completeness. It is interesting to point out the behavior which occurs when $\omega^j = 2$. In this case, our collision probability is $\omega = 0$. Here, the

flipping operation is guaranteed to permute all particles back to their original position through the collision with a probability of 1. This in combination with the streaming step will cause all the particles in the system to continually permute which will not evolve the system at all. This data was acquired using a D1Q3 lattice with a size of $L = 320$ with $N^{ave} = 100$ particles per lattice node averaged over 500 individual simulations.

4.6. Conclusions

We have presented a new method for successfully performing overrelaxation in diffusive integer lattice gas models, which had previously been thought to be impossible. This method introduces a simple permutation of the occupation numbers within the system to over shoot local equilibrium. This is made possible by defining an effective collision probability which nullifies the mathematically impossibility of utilizing probabilities greater than 1. This works in tandem with the sampling collision operator presented by Seekins, but it is also generally possible to implement on any collision operator. The ability to utilize overrelaxation in integer lattices gases will increase the usefulness and practicality of the method. The example of diffusion has provided a pathway in which we intend to develop a fully realized overrelaxed hydrodynamic integer lattice gas.

5. CONCLUSIONS AND OUTLOOK

The motivation for the work contained in this dissertation was initially driven by a desire to more effectively implement fluctuations into lattice Boltzmann methods. Although lattice Boltzmann methods which utilized fluctuations had been studied previously, these early fluctuating lattice Boltzmann models had theoretical shortcomings, such as requiring noise to be constant, or requiring Fourier space, which limited practical use.

We presented a new fluctuating lattice Boltzmann model for diffusive systems which included Poisson distributed noise with a fully decoupled fluctuation-dissipation theorem. This new model was derived in a self-contained manner which gave density dependent noise and did not require a transformation into Fourier space to include fluctuations to the system. Even though this new method required a change to the moment space basis, the overall simplicity of the model provides a strong argument for the practicality of this method.

The simplicity of the diffusive models which were utilized in fluctuating lattice Boltzmann methods motivated interest in the practical applications for which diffusive lattice Boltzmann methods could be used. We investigated the applicability of diffusive lattice Boltzmann for fluid transport through barrier coatings. It was found that a simple diffusive lattice Boltzmann model could reproduce experimental data for such systems. The error between theory and experiment was investigated through higher-order analysis on the equations of motion in the hydrodynamic limit for the system. Although, the corrections which were developed through the higher-order analysis did not significantly correct the error found in the systems, a consistent method for performing higher-order analysis on lattice Boltzmann methods was presented.

After the presentation of Blommel's Monte Carlo integer lattice gas, and the subsequent development of a sampling collision operator for use in integer lattice gas, renewed interest was found in lattice gas methods. The significant drawback of being unable to implement overrelaxation for lattice gas model was brought to the forefront. Using these newly developed integer lattice gas methods gave insight into the possibility of using overrelaxation in lattice gases. It was discovered that, through a simple permutation of occupation numbers over a velocity set, overrelaxation was,

in fact, possible in lattice gas, and we presented the first model which allows for overrelaxation in the probabilistic integer lattice gas.

These novel developments for lattice Boltzmann and lattice gas methods have allowed for the possibility of extending these methods for further practical application and further study. It is of interest to continue research into the ideas presented in this manuscript.

5.1. Molecular Dynamics Lattice Gas for Diffusive Systems

The development of the Molecular Dynamics Lattice Gas (MDLG) analysis tool by Parsa *et al.* allows for deriving lattice Boltzmann methods through a coarse graining process of a molecular dynamics simulation [83, 84]. This analysis tool can gives new insight into the physical properties of lattice Boltzmann methods. Pachalieva *et al.* were able to gain understanding of the physical properties associated with overrelaxation [86] which were not previously understood. By examining diffusive systems from molecular dynamics using MDLG, we can gain further understanding into the behavior and construction of diffusive lattice Boltzmann and integer lattice gas methods.

5.2. Higher Order Analysis and Telegrapher's Equation

Although diffusive lattice Boltzmann and integer lattice gas methods are simple to implement and provide good results, it has been argued by Chopard and Droz that these diffusive models may not be modeling the diffusion equaiton, but rather the telegrapher's equation [15]. The telegrapher's equation is similar to the diffusion equation except that there is an additional second order temporal derivative. The telegrapher's equation is written

$$\partial_t \rho + \frac{D}{\theta} \partial_t^2 \rho = D \nabla^2 \rho. \tag{5.1}$$

This is a reasonable assumption since lattice Boltzmann and lattice gas methods are discrete and must propagate information at a finite speed. This is not the case for the diffusion equation. The second order time derivative in the telegrapher's equation gives a finite speed for information propagation.

As was discussed in section 3.4, we neglected second order time derivatives using the diffusion equation itself by writing the temporal derivatives in terms of the spatial derivatives from Eqns. (3.28-3.29) with

$$\partial_t \rho = \left(\tau - \frac{1}{2} \right) \nabla_\alpha^2 \rho \theta + O(\nabla^3) \tag{5.2}$$

$$\partial_t^2 \rho = \left(\tau - \frac{1}{2}\right)^2 \nabla_\alpha^2 \nabla_\beta^2 \rho\theta^2 + O(\nabla^5). \tag{5.3}$$

Since this is a further approximation, it is of sufficient interest to examine the diffusive lattice Boltzmann and integer lattice gas methods to verify if in fact these models reproduce the telegrapher's equation as opposed to the diffusion equation.

5.3. Overrelaxation for Integer Lattice Gases with Full Hydrodynamics.

The successful implementation of overelaxation for integer lattice gases for diffusion is of particular interest. Utilizing overelaxation in integer lattice gases can extend the range of stable transport coefficients in the method, which is directly related to a hydrodynamic implementation due to the inclusion of viscous terms. The diffusive case does not contain viscous terms, so this simple implementation serves as a proof of concept for further use of overrelaxation in the method. Successfully implementing overrelaxation for a full hydrodynamic system could greatly expand the usefulness and potential of integer lattice gases.

REFERENCES

[1] Sauro Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond (Numerical Mathematics and Scientific Computation)*. Numerical Mathematics and Scientific Computation. Oxford University Press, 1 edition, August 2001.

[2] Timm Krüger, Halim Kusumaatmaja, Alexander Kuzmin, Orest Shardt, Goncalo Silva, and Erlend Magnus Viggen. *The Lattice Boltzmann Method - Principles and Practice*. Springer, 10 2016.

[3] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-Gas Automata for the Navier-Stokes Equation. *Phys. Rev. Lett.*, 56:1505–1508, Apr 1986.

[4] U. Frisch, D. d'Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, and J. P. Rivet. Lattice Gas Hydrodynamics in Two and Three Dimensions. *Complex Syst.*, 1:649–707, 1987.

[5] J.-P. Rivet and J. P. Boon. *Lattice Gas Hydrodynamics*. Cambridge Nonlinear Science Series. Cambridge University Press, 2001.

[6] Jill P. Dahlburg, David Montgomery, and Gary D. Doolen. Noise and Compressibility in Lattice-Gas Fluids. *Phys. Rev. A*, 36:2471–2474, Sep 1987.

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

[8] Anthony J. C. Ladd. Numerical Simulations of Particulate Suspensions via a Discretized Boltzmann Equation. Part 2. Numerical Results. *Journal of Fluid Mechanics*, 271:311–339, 7 1994.

[9] Dieter Wolf-Gladrow. A Lattice Boltzmann Equation for Diffusion. *Journal of statistical physics*, 79(5-6):1023–1032, 1995.

[10] Alexander J. Wagner and Kyle Strand. Fluctuating Lattice Boltzmann Method for the Diffusion Equation. *Phys. Rev. E*, 94:033302, Sep 2016.

[11] Kyle T. Strand, Aaron J. Feickert, and Alexander J. Wagner. Fourth-Order Analysis of a Diffusive Lattice Boltzmann Method for Barrier Coatings. *arxiv:physics.comp-ph/1703.05795 (preprint)*, 2017.

[12] A.J. Wagner and S. May. Electrostatic Interactions across a Charged Lipid Bilayer. *Eur Biophys J*, 36:293–303, April 2007.

[13] Alexander J. Wagner. A practical introduction to the lattice boltzmann method, 2008.

[14] Dieter A. Wolf-Gladow. *Lattice Gas Cellular Automata and Lattice Boltzmann Models*. Lecture Notes in Mathematics. Springer, 2000.

[15] Bastien Chopard and Michel Droz. *Cellular Automata Modeling of Physical Systems*. Collection Alea-Saclay: Monographs and Texts in Statistical Physics. Cambridge University Press, 1998.

[16] 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.*, 94:511–525, May 1954.

[17] Thomas Blommel and Alexander J Wagner. Integer Lattice Gas with Monte Carlo Collision Operator Recovers the Lattice Boltzmann Method with Poisson-Distributed Fluctuations. *Phys. Rev. E*, 97(2):023310, 2018.

[18] Laure Saint-Raymond. Hydrodynamic limits: some improvements of the relative entropy method. *Annales de l'Institut Henri Poincaré C, Analyse non linéaire*, 26(3):705–744, 2009.

[19] Laure Saint-Raymond. Convergence of Solutions to the Boltzmann Equation in the Incompressible Euler Limit. *Archive for Rational Mechanics and Analysis*, 166(1):47–80, January 2003.

[20] P.A. Thompson. *Compressible-fluid Dynamics*. Advanced engineering series. McGraw-Hill, 1972.

[21] A. J. C. Ladd. Short-Time Motion of Colloidal Particles: Numerical Simulation via a Fluctuating Lattice-Boltzmann Equation. *Phys. Rev. Lett.*, 70:1339–1342, Mar 1993.

[22] R. Adhikari, K. Stratford, M. E. Cates, and A. J. Wagner. Fluctuating Lattice Boltzmann. *EPL (Europhysics Letters)*, 71(3):473, 2005.

[23] Noah Seekins and Alexander J Wagner. Integer Lattice Gas with a Sampling Collision Operator for the Fluctuating Diffusion Equation. *Physical Review E*, 105(3):035303, 2022.

[24] J. Hardy, Y. Pomeau, and O. de Pazzis. Time Evolution of a Two-Dimensional Model System. I. Invariant States and Time Correlation Functions. *Journal of Mathematical Physics*, 14(12):1746–1759, 1973.

[25] Mitchel Johnson, Daniel Playne, and Ken Hawick. Data-Parallelism and GPUs for Lattice Gas Fluid Simulations., 01 2010.

[26] S. Harris. Approach to equilibrium in a moderately dense discrete velocity gas. *The Physics of Fluids*, 9(7):1328–1332, 1966.

[27] Stephen Wolfram. Cellular automaton fluids 1: Basic theory. *J. Stat. Phys*, 45, 1986.

[28] Fernand Hayot. The effect of galilean non-invariance in lattice gas automaton one-dimensional flow. *Complex Syst.*, 1(4), 1987.

[29] Hudong Chen, Chris Teixeira, and Kim Molvig. Digital Physics Approach to Computational Fluid Dynamics: Some Basic Theoretical Features. *International Journal of Modern Physics C*, 08(04):675–684, 1997.

[30] Christopher M. Teixeira. Digital Physics Simulation of Lid-Driven Cavity Flow. *International Journal of Modern Physics C*, 08(04):685–696, 1997.

[31] Andrew Anagnost, Ales Alajbegovic, Hudong Chen, David Hill, Chris Teixeira, and Kim Molvig. "digital Physics^{TM}" Analysis of the Morel Body in Ground Proximity. *SAE Transactions*, 106:306–312, 1997.

[32] J. Halliday, C. Teixeira, and C. Alexander. Simulation of Engine Internal Flows Using Digital Physics. *Oil Gas Sci. Tech*, 54:187, 1999.

[33] Sauro Succi, Hudong Chen, Chris Teixeira, Gino Bella, A De Maio, and Kim Molvig. An Integer Lattice Realization of a Lax Scheme for Transport Processes in Multiple Component Fluid Flows. *Journal of Computational Physics*, 152(2):493–516, 1999.

[34] Bruce M. Boghosian, Jeffrey Yepez, Francis J. Alexander, and Norman H. Margolus. Integer Lattice Gases. *Phys. Rev. E*, 55:4137–4147, Apr 1997.

[35] A. Masselot and B. Chopard. A Multiparticle Lattice-Gas Model for Hydrodynamics. *International Journal of Modern Physics C*, 09(08):1221–1230, 1998.

[36] Bastien Chopard, Alexandre Masselot, and Michel Droz. Multiparticle Lattice Gas Model for a Fluid: Application to Ballistic Annihilation. *Phys. Rev. Lett.*, 81:1845–1848, Aug 1998.

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

[38] S Ansumali, I. V Karlin, and H. C Öttinger. Minimal Entropic Kinetic Models for Hydrodynamics. *Europhysics Letters (EPL)*, 63(6):798–804, sep 2003.

[39] Alexander J Wagner and Kyle Strand. Fluctuating Lattice Boltzmann Method for the Diffusion Equation. *Phys. Rev. E*, 94(3):033302, 2016.

[40] Brian Gough. *GNU Scientific Library Reference Manual - Third Edition.* Network Theory Ltd., 3rd edition, 2009.

[41] G. Kaehler and A. J. Wagner. Low Density Limit of Locally Fluctuating Lattice Boltzmann Methods. in preparation.

[42] J. Renn. Einstein's Invention of Brownian Motion. *Annalen der Physik*, 14(S1):23–37, 2005.

[43] Don S. Lemons and Anthony Gythiel. Paul Langevin's 1908 Paper "On the Theory of Brownian Motion" ["Sur la Théorie du Mouvement Brownien," C. R. Acad. Sci. (Paris) 146, 530–533 (1908)]. *American Journal of Physics*, 65(11):1079–1081, 1997.

[44] L. D. Landau and E. M. Lifshitz. *Fluid Mechanics, Second Edition: Volume 6 (Course of Theoretical Physics).* Course of Theoretical Physics / by L. D. Landau and E. M. Lifshitz, Vol. 6. Butterworth-Heinemann, 2 edition, January 1987.

[45] R. F. Fox and G. E. Uhlenbeck. Contributions to Nonequilibrium Thermodynamics. II Fluctuation Theory for the Boltzmann Equation. *Phys. Fluids*, 13:2881–2890, 1970.

[46] Mordechai Bixon and Robert Zwanzig. Boltzmann-Langevin Equation and Hydrodynamic Fluctuations. *Phys. Rev.*, 187:267–272, Nov 1969.

[47] JW Dufty and MH Ernst. Lattice Boltzmann-Langevin Equations. *Pattern Formation and Lattice gas Automata*, 6:99, 1997.

[48] Burkhard Dünweg, Ulf D. Schiller, and Anthony J. C. Ladd. Statistical Mechanics of the Fluctuating Lattice Boltzmann Equation. *Phys. Rev. E*, 76:036704, Sep 2007.

[49] G. Kaehler and A. J. Wagner. Fluctuating Ideal-Gas Lattice Boltzmann Method with Fluctuation Dissipation Theorem for Nonvanishing Velocities. *Phys. Rev. E*, 87:063310, Jun 2013.

[50] FJ Higuera, S Succi, and R Benzi. Lattice Gas Dynamics with Enhanced Collisions. *EPL (Europhysics Letters)*, 9(4):345, 1989.

[51] Xiaowen Shan and Gary Doolen. Diffusion in a Multicomponent Lattice Boltzmann Equation Model. *Phys. Rev. E*, 54:3614–3620, Oct 1996.

[52] Enzo Orlandini, Michael R Swift, and JM Yeomans. A Lattice Boltzmann Model of Binary-Fluid Mixtures. *EPL (Europhysics Letters)*, 32(6):463, 1995.

[53] Like Li, Chen Chen, Renwei Mei, and James F. Klausner. Conjugate Heat and Mass Transfer in the Lattice Boltzmann Equation Method. *Phys. Rev. E*, 89:043308, Apr 2014.

[54] Guigao Le, Othmane Oulaid, and Junfeng Zhang. Publisher's Note: Counter-Extrapolation Method for Conjugate Interfaces in Computational Heat and Mass Transfer [Phys. Rev. E **91** , 033306 (2015)]. *Phys. Rev. E*, 92:049904, Oct 2015.

[55] M. Gross, R. Adhikari, M. E. Cates, and F. Varnik. Thermal Fluctuations in the Lattice Boltzmann Method for Nonideal Fluids. *Phys. Rev. E*, 82:056714, Nov 2010.

[56] Sumesh P. Thampi, Ignacio Pagonabarraga, and R. Adhikari. Lattice-Boltzmann-Langevin Simulations of Binary Mixtures. *Phys. Rev. E*, 84:046709, Oct 2011.

[57] D. Belardinelli, M. Sbragaglia, L. Biferale, M. Gross, and F. Varnik. Fluctuating Multicomponent Lattice Boltzmann Model. *Phys. Rev. E*, 91:023313, Feb 2015.

[58] Y. H. Qian, D. D'Humières, and P. Lallemand. Lattice BGK Models for Navier-Stokes Equation. *EPL (Europhysics Letters)*, 17(6):479, 1992.

[59] A. Einstein. Zur Theorie der Brownschen Bewegung. *Annalen der Physik*, 324(2):371–381, 1906.

[60] G. Kaehler and A. J. Wagner. Derivation of Hydrodynamics for Multi-Relaxation Time Lattice Boltzmann using the Moment Approach. *Comm. Comp. Phys.*, 13:614–628, Mar 2013.

[61] Qun Li and A. J. Wagner. Symmetric Free-Energy-Based Multicomponent Lattice Boltzmann Method. *Phys. Rev. E*, 76:036701, Sep 2007.

[62] Goetz August Kaehler. *Fluctuations in the Lattice Boltzmann Method.* PhD thesis, North Dakota State University, UMI Dissertation Publishing, 11 2012. http://gradworks.umi.com/35/62/3562140.html.

[63] Santtu TT Ollila, Colin Denniston, Mikko Karttunen, and Tapio Ala-Nissila. Fluctuating Lattice-Boltzmann Model for Complex Fluids. *The Journal of chemical physics*, 134(6):064902, 2011.

[64] E. M. Foard and A. J. Wagner. Enslaved Phase-Separation Fronts in One-Dimensional Binary Mixtures. *Phys. Rev. E*, 79:056710, May 2009.

[65] E. M. Foard and A. J. Wagner. Survey of Morphologies Formed in the Wake of an Enslaved Phase-Separation Front in Two Dimensions. *Phys. Rev. E*, 85:011501, Jan 2012.

[66] Dieter Forster. Hydrodynamic Fluctuations, Broken Symmetry, and Correlation Functions. In *Reading, Mass., WA Benjamin, Inc.(Frontiers in Physics. Volume 47), 1975. 343 p.*, volume 47, 1975.

[67] Kyle Strand. FluctuatingD2Q5. `https://github.com/KyleStrandLB/FluctuatingD2Q5/tree/v1.0`, 2016.

[68] S.R. Taylor, F. Contu, R. Santhanam, and P. Suwanna. The Use of Cationic Fluoroprobes to Characterize Ionic Pathways in Organic Coatings. *Progress in Organic Coatings*, 73(2–3):169 – 172, 2012.

[69] D.M. Kroll and S.G. Croll. Influence of Crosslinking Functionality, Femperature and Conversion on Heterogeneities in Polymer Networks. *Polymer*, 79:82 – 90, 2015.

[70] Malia Zee, Aaron J. Feickert, D.M. Kroll, and S.G. Croll. Cavitation in Crosslinked Polymers: Molecular Dynamics Simulations of Network Formation. *Progress in Organic Coatings*, 83:55 – 63, 2015.

[71] Jonathan W. Martin. *Repeatability and Reproducibility of Field Exposure Results*, chapter 1, pages 2–22. American Chemical Society, 2001.

[72] John Crank. *The Mathematics of Diffusion*. Oxford University Press, 1979.

[73] L De Rosa, T Monetta, DB Mitton, and F Bellucci. Monitoring Degradation of Single and Multilayer Organic Coatings I. Absorption and Transport of Water: Theoretical Analysis and Methods. *Journal of the Electrochemical Society*, 145(11):3830–3838, 1998.

[74] Minas M. Mezedur, Massoud Kaviany, and Wayne Moore. Effect of Pore Structure, Randomness and Size on Effective Mass Diffusivity. *AIChE Journal*, 48(1):15–24, 2002.

[75] Muhammad Sahimi and Dietrich Stauffer. Efficient Simulation of Flow and Transport in Porous Media. *Chemical Engineering Science*, 46(9):2225 – 2233, 1991.

[76] Deqiang Mu, Zhong-Sheng Liu, Cheng Huang, and Ned Djilali. Prediction of the Effective Diffusion Coefficient in Random Porous Media Using the Finite Element Method. *Journal of Porous Materials*, 14(1):49–54, 2007.

[77] Aman Pathania, Raj Kumar Arya, and Sanjeev Ahuja. Crosslinked Polymeric Coatings: Preparation, Characterization, and Diffusion Studies. *Progress in Organic Coatings*, 105:149 – 162, 2017.

[78] Viktor Baukh, Hendrik P. Huinink, Olaf C.G. Adan, Sebastiaan J.F. Erich, and Leendert G.J. van der Ven. Predicting Water Transport in Multilayer Coatings. *Polymer*, 53(15):3304 – 3312, 2012.

[79] Michael R. Swift, W. R. Osborn, and J. M. Yeomans. Lattice Boltzmann Simulation of Nonideal Fluids. *Phys. Rev. Lett.*, 75:830–833, Jul 1995.

[80] Irina Ginzburg. Prediction of the Moments in Advection-Diffusion Lattice Boltzmann Method. I. Truncation Dispersion, Skewness, and Kurtosis. *Phys. Rev. E*, 95:013304, Jan 2017.

[81] A. J. Wagner. Thermodynamic Consistency of Liquid-Gas Lattice Boltzmann Simulations. *Phys. Rev. E*, 74:056703, Nov 2006.

[82] Gary D Doolen. *Lattice Gas Methods: Theory, Applications, and Hardware*, volume Vol. 47. MIT Press, 1991.

[83] M Reza Parsa and Alexander J Wagner. Lattice Gas with Molecular Dynamics Collision Operator. *Phys. Rev. E*, 96(1):013314, 2017.

[84] M Reza Parsa, Aleksandra Pachalieva, and Alexander J Wagner. Validity of the Molecular-Dynamics-Lattice-Gas Global Equilibrium Distribution Function. *International Journal of Modern Physics C*, 30, November 2018.

[85] F. Bösch and I. V. Karlin. Exact Lattice Boltzmann Equation. *Phys. Rev. Lett.*, 111:090601, Aug 2013.

[86] Aleksandra Pachalieva and Alexander J Wagner. Connecting Lattice Boltzmann Methods to Physical Reality by Coarse-Graining Molecular Dynamics Simulations. *arXiv preprint arXiv:2109.05009*, 2021.

[87] Y. H Qian and S. A Orszag. Lattice BGK Models for the Navier-Stokes Equation: Nonlinear Deviation in Compressible Regimes. *Europhysics Letters (EPL)*, 21(3):255–259, jan 1993.

[88] C. Sorenson and A.J. Wagner. Entropy Production in Lattice Boltzmann and Integer Lattice Gases. *In progress*, 2022.

[89] Kyle T. Strand, Aaron J. Feickert, and Alexander J. Wagner. Fourth-Order Analysis of a Diffusive Lattice Boltzmann Method for Barrier Coatings. *Phys. Rev. E*, 95:063311, Jun 2017.

[90] Jianying Zhang and Guangwu Yan. A lattice boltzmann model for the korteweg–de vries equation with two conservation laws. *Computer Physics Communications*, 180(7):1054 – 1062, 2009.

[91] Alexander Wagner. Graphical User Interface. `https://www.ndsu.edu/pubweb/~carswagn/GUI/index.html`, Jan 2016.

APPENDIX

A.1. Mass and Momentum Conservation

In the previous section, the diffusion equation was derived from Eqn. (1.59) by imposing that the only conserved quantity in the system was mass. If there is a system which includes mass and momentum conservation, the macroscopic moments of the equilibrium distribution of Eqn. (1.43) with fixed θ can be defined to reproduce momentum conservation. For systems which conserve mass and momentum, the mass and momentum are defined through sums over the distribution functions

$$\rho(\boldsymbol{x},t) = \sum_i f_i(\boldsymbol{x},t) \tag{A.1}$$

$$\rho(\boldsymbol{x},t)\boldsymbol{u}(\boldsymbol{x},t) = \sum_i \boldsymbol{v}_i f_i(\boldsymbol{x},t) \tag{A.2}$$

where $\boldsymbol{u}(\boldsymbol{x},t)$ is the macroscopic flow velocity. By imposing these definitions of the mass and momentum, the definitions can be extended as moments of the equilibrium distributions such that

$$\sum_i f_i^0 = \rho \tag{A.3}$$

$$\sum_i v_{i\alpha} f_i^0 = \rho u_\alpha \tag{A.4}$$

$$\sum_i v_{i\alpha} v_{i\beta} f_i^0 = \rho(u_\alpha u_\beta + \theta\delta_{\alpha\beta}). \tag{A.5}$$

In the same manner as the diffusive case, beginning with Eqn. (1.59) and summing over all i with the new macroscopic moments defined for mass and momentum conservation, the equation becomes

$$\partial_t\rho + \partial_\alpha(\rho u_\alpha) - \left(\tau - \frac{1}{2}\right)\{\partial_t(\partial_t\rho + \partial_\beta\rho u_\beta) + \partial_\alpha\left[\partial_t\rho u_\alpha + \partial_\beta(\rho u_\alpha u_\beta + \rho\theta\delta_{\alpha\beta})\right]\} + O(\partial^3) = 0. \tag{A.6}$$

If $\partial_t \rho + \partial_\alpha(\rho u_\alpha)$ is iteratively substituted into itself, the first term under the $\tau - 1/2$ will go to $O(\partial^3)$. By doing so, the equation becomes

$$\partial_t \rho + \partial_\alpha(\rho u_\alpha) = \left(\tau - \frac{1}{2}\right) \partial_\alpha \left[(\partial_t \rho u_\alpha + \partial_\beta(\rho u_\alpha u_\beta + \rho\theta\delta_{\alpha\beta})\right] + O(\partial^3). \tag{A.7}$$

At this point this mass conservation equation is in its simplest form. Now, if Eqn.(1.59) is multiplied by an additional factor of $\boldsymbol{v}_i$ and summed over i once again, a momentum conservation equation can be found by a similar process

$$(\partial_t + v_{i\beta}\partial_\beta)v_{i\beta} f_i^0 - \left(\tau - \frac{1}{2}\right)(\partial_t + v_{i\gamma}\partial_\gamma)(\partial_t + v_{i\beta}\partial_\beta)v_{i\alpha} f_i^0 + O(\partial^3) = \Omega_i v_{i\alpha}$$

$$\partial_t(\rho u_\alpha) + \partial_\beta(\rho u_\beta u_\alpha + \rho\theta\delta_{\alpha\beta}) = \left(\tau - \frac{1}{2}\right)\left\{\partial_t\left[\partial_t(\rho u_\alpha) + \partial_\beta(\rho u_\beta u_\alpha + \rho\theta\delta_{\alpha\beta})\right] + \right.$$
$$\left. \partial_\gamma\left[\partial_t(\rho u_\gamma u_\alpha + \rho\theta\delta_{\alpha\gamma}) + \partial_\beta \sum_i v_{i\alpha} v_{i\beta} v_{i\gamma} f_i^0\right]\right\} + O(\partial^3). \tag{A.8}$$

Once again, an iterative substitution of the left hand side of Eqn. (A.8) will bring the terms which are second order under ∂_t on the right hand side to $O(\partial^3)$. This leaves

$$\partial_t(\rho u_\alpha) + \partial_\beta(\rho u_\beta u_\alpha + \rho\theta\delta_{\alpha\beta}) = \left(\tau - \frac{1}{2}\right)\partial_\gamma\left[\partial_t(\rho u_\gamma u_\alpha + \rho\theta\delta_{\alpha\gamma}) + \partial_\beta \sum_i v_{i\alpha} v_{i\beta} v_{i\gamma} f_i^0\right] + O(\partial^3). \tag{A.9}$$

If this is substituted into Eqn. (A.7), it is seen that all terms on the right hand side are of $O(\partial^3)$. With this, the equation becomes the mass conservation equation known as the continuity equation which states

$$\partial_t \rho + \partial_\alpha(\rho u_\alpha) + O(\partial^3) = 0. \tag{A.10}$$

Now in returning to Eqn. (A.9), there is still an unknown moment which is yet to be summed. Due to symmetry, it is expected that this third order moment should reproduce the first order moment. With this, this unknown moment is defined as

$$\sum_i v_{i\alpha} v_{i\beta} v_{i\gamma} f_i^0 = \rho\theta(u_\alpha \delta_{\beta\gamma} + u_\beta \delta_{\alpha\gamma} + u_\gamma \delta_{\alpha\beta}). \tag{A.11}$$

Using this moment in Eqn. (A.9) yields

$$\partial_t(\rho u_\alpha)+\partial_\beta(\rho u_\beta u_\alpha + \rho\theta\delta_{\alpha\beta}) = \left(\tau - \frac{1}{2}\right)\partial_\gamma\left\{\partial_t(\rho u_\gamma u_\alpha + \rho\theta\delta_{\alpha\gamma}) + \partial_\beta\left[\rho\theta(u_\alpha\delta_{\beta\gamma} + u_\beta\delta_{\alpha\gamma} + u_\gamma\delta_{\alpha\beta})\right]\right\} + O(\partial^3). \quad \text{(A.12)}$$

Using the product rule to evaluate the derivatives on the left hand side gives the relation

$$\partial_t(\rho u_\alpha) + \partial_\beta(\rho u_\beta u_\alpha + \rho\theta\delta_{\alpha\beta}) = \partial_t(\rho)u_\alpha + \rho\partial_t u_\alpha + \partial_\beta(\rho u_\beta)u_\alpha + \rho u_\beta\partial_\beta u_\alpha + \partial_\beta\rho\theta\delta_{\alpha\beta}. \quad \text{(A.13)}$$

This can be inserted into Eqn. (A.12) such that

$$\partial_t(\rho)u_\alpha + \rho\partial_t u_\alpha + \partial_\beta(\rho u_\beta)u_\alpha + \rho u_\beta\partial_\beta u_\alpha + \partial_\beta\rho\theta\delta_{\alpha\beta} = \left(\tau - \frac{1}{2}\right)\partial_\gamma\left\{\partial_t(\rho u_\gamma u_\alpha + \rho\theta\delta_{\alpha\gamma}) + \partial_\beta\left[\rho\theta(u_\alpha\delta_{\beta\gamma} + u_\beta\delta_{\alpha\gamma} + u_\gamma\delta_{\alpha\beta})\right]\right\} + O(\partial^3). \quad \text{(A.14)}$$

Applying the continuity equation to the left hand side, will allow for further simplification to the form

$$\rho\partial_t u_\alpha + \rho u_\beta\partial_\beta u_\alpha = -\partial_\beta\rho\theta\delta_{\alpha\beta} + \left(\tau - \frac{1}{2}\right)\partial_\gamma\left\{\partial_t(\rho u_\gamma u_\alpha + \rho\theta\delta_{\alpha\gamma}) + \partial_\beta\left[\rho\theta(u_\alpha\delta_{\beta\gamma} + u_\beta\delta_{\alpha\gamma} + u_\gamma\delta_{\alpha\beta})\right]\right\} + O(\partial^3). \quad \text{(A.15)}$$

The product rule can also be applied to the ∂_t term on the right hand side. This becomes

$$\partial_t(\rho u_\gamma u_\alpha + \rho\theta\delta_{\alpha\gamma}) = \partial_t(\rho u_\gamma)u_\alpha + \rho u_\gamma\partial_t u_\beta + \partial_t\rho\theta\delta_{\alpha\beta}. \quad \text{(A.16)}$$

Using the continuity equation, these temporal derivatives can be written in terms of spatial derives. This substitution gives

$$\partial_t(\rho u_\gamma u_\alpha + \rho\theta\delta_{\alpha\gamma}) = \partial_\beta(\rho u_\gamma u_\beta)u_\alpha + \partial_\alpha(\rho\theta\delta_{\alpha\gamma}u_\alpha + \rho u_\gamma u_\beta\partial_\beta u_\alpha + u_\gamma\partial_\gamma(\rho\theta\delta_{\alpha\gamma}) + \partial_\beta\rho u_\beta\theta\delta_{\alpha\gamma}. \quad \text{(A.17)}$$

Now inserting this into Eqn. (A.15), after some simplification, a momentum conservation equation can be reached as

$$\rho\partial_t u_\alpha + \rho u_\beta \partial_\beta u_\alpha = -\partial_\beta \rho\theta\delta_{\alpha\beta} + \left(\tau - \frac{1}{2}\right)\partial_\gamma\left[\rho\theta(\partial_\alpha u_\beta + \partial_\beta u_\alpha)\right] + O(\partial^3). \tag{A.18}$$

This momentum conservation is an equivalent form of the Navier-Stokes equation. If the viscosity is defined

$$\eta = \rho\theta\left(\tau - \frac{1}{2}\right), \tag{A.19}$$

a compact form can be written

$$\rho\partial_t u_\alpha + \rho u_\beta \partial_\beta u_\alpha = -\partial_\beta \rho\theta\delta_{\alpha\beta} + \partial_\gamma\left[\eta(\partial_\alpha u_\beta + \partial_\beta u_\alpha)\right] + O(\partial^3). \tag{A.20}$$

It has been shown that for a system which only conserves mass, the lattice Boltzmann equation reproduces the diffusion equation. For a system in which mass and momentum are conserved, there are two equations of motion which are reproduced: the continuity equation and the Navier-Stokes equation. In principle, if energy was to be conserved by allowing variable θ, a third equation of motion, the heat equation could be recovered. However, lattice Boltzmann methods generally treat systems isothermally, so there is no energy conservation.

A.2. Moments of Poisson Distributed Random Variables

To calculate the expectation value of Eqn. (2.5) we only need to consider two distributions at the same location, so we will drop the spatial dependence here:

$$\langle n_i n_j \rangle. \tag{A.21}$$

We need to distinguish the two cases of equal and different values for i and j. Let us first consider $i \neq j$. We then get

$$\begin{aligned}
&\langle n_i n_j \rangle \\
&= \sum_{n_i n_j} P(n_i) P(n_j) n_i n_j \\
&= \exp(-n_i^{eq} - n_j^{eq}) \sum_{n_i n_j} \frac{(n_i^{eq})^{n_i} (n_j^{eq})^{n_j}}{n_i! n_j!} n_i n_j \\
&= \exp(-n_i^{eq} - n_j^{eq}) (n_i^{eq} \partial_{n_i^{eq}}) (n_j^{eq} \partial_{n_j^{eq}}) \sum_{n_i n_j} \frac{(n_i^{eq})^{n_i} (n_j^{eq})^{n_j}}{n_i! n_j!} \\
&= \exp(-n_i^{eq} - n_j^{eq}) (n_i^{eq} \partial_{n_i^{eq}}) (n_j^{eq} \partial_{n_j^{eq}}) \sum_{n n_j} \frac{(n_i^{eq})^{n-n_j} (n_j^{eq})^{n_j}}{(n-n_j)! n_j!} \\
&= \exp(-n_i^{eq} - n_j^{eq}) (n_i^{eq} \partial_{n_i^{eq}}) (n_j^{eq} \partial_{n_j^{eq}}) \sum_{n} \frac{(n_i^{eq} + n_j^{eq})^{n}}{n!} \\
&= \exp(-n_i^{eq} - n_j^{eq}) (n_i^{eq} \partial_{n_i^{eq}}) (n_j^{eq} \partial_{n_j^{eq}}) \exp(n_i^{eq} + n_j^{eq}) \\
&= n_i^{eq} n_j^{eq}
\end{aligned}$$

where we used the expression for the binomial formula in line

Now let us consider the case $i = j$. We get

$$\begin{aligned}
&\langle n_i n_i \rangle \\
&= \sum_{n_i n_j} P(n_i) n_i n_i \\
&= \exp(-n_i^{eq}) \sum_{n_i} \frac{(n_i^{eq})^{n_i}}{n_i!} n_i n_i \\
&= \exp(-n_i^{eq}) (n_i^{eq} \partial_{n_i^{eq}}) (n_i^{eq} \partial_{n_i^{eq}}) \sum_{n_i n_j} \frac{(n_i^{eq})^{n_i}}{n_i!} \\
&= \exp(-n_i^{eq}) (n_i^{eq} \partial_{n_i^{eq}}) [n_i \exp(n_i^{eq})] \\
&= n_i^{eq} n_i^{eq} + n_i^{eq}
\end{aligned}$$

This can be summarized as

$$\langle n_i n_j \rangle = n_i^{eq} n_j^{eq} + n_i^{eq} \delta_{ij}. \tag{A.22}$$

Similarly we get

$$\begin{aligned}&\langle n_i n_j n_k \rangle \\ =&n_i^{eq} n_j^{eq} n_k^{eq} + n_i^{eq} n_k^{eq} \delta_{ij} + n_i^{eq} n_j^{eq} \delta_{jk} + n_j^{eq} n_k^{eq} \delta_{ik} \\ &+ n_i^{eq} \delta_{ijk}\end{aligned} \tag{A.23}$$

A.3. Continuous and Discrete Solutions

The error function solution in Eqn. (3.14) solves the second-order diffusion equation with the given boundary conditions in continuous real space (see [72] for a basic form of the derivation). However, we later compute a solution by transforming the appropriate initial condition into Fourier space, performing a second-order time evolution, and then transforming back into real space. This process uses a finite number of k modes in each transform, and necessarily implies a discrete lattice sampling of both the initial condition in real space and the time-evolved form in Fourier space. We therefore expect a discrepancy when directly comparing the two solutions: the first is a solution to the continuous diffusion equation that is examined at discrete lattice points for comparison to the simulation, while the second is a sampled solution to the discrete lattice diffusion equation, the continuous form of which would require (in theory) an infinite number of k modes to match the continuous case.

To examine the extent to which these solution forms differ from each other, we compute both at the same scaled four-hour time at each lattice site, and plot the absolute value of the difference, ε, in Fig. A.1. The two solutions agree to within 10^{-5} of each other. Since this error is on the order of the remaining error for the periodically-embedded simulation, we conclude that any further correction of simulation results renders any error obscured by differences between the discrete and continuous solutions to the diffusion equation, and is of no practical consequence.

A.4. Extension of Fourth Order Expansion to Arbitrary Order

Here we introduce the method of generalizing the fourth order expansion to arbitrary order [89]. Beginning with Eqn. (3.21), we first recognize that this can be written as a series. We can

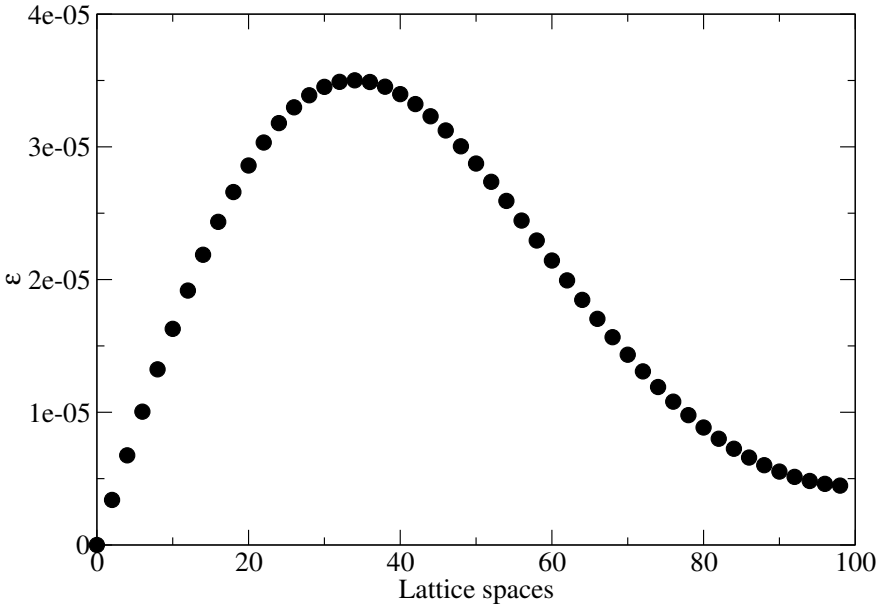


Figure A.1. Absolute error profile ε between second-order error function and second-order Fourier solutions to the continuous and discrete diffusion equation, respectively.

then rewrite this equation as

$$\sum_{m=1}^{4} \lambda_m(\tau) \frac{(\partial_t + v_{i\alpha}\partial_\alpha)^m}{m!}(f_i^0 + \tau F_i) + O(\partial^5) = \Omega_i \tag{A.24}$$

where $\lambda_m(\tau)$ are the Bernoulli polynomials for each specific order [90]. We can generalize this series to arbitrary order by extending the limits on the sum such that

$$\sum_{m=1}^{n} \lambda_m(\tau) \frac{(\partial_t + v_{i\alpha}\partial_\alpha)^m}{m!}(f_i^0 + \tau F_i) + O(\partial^{n+1}) = \Omega_i \tag{A.25}$$

where n is the desired order of the expansion. For the sake of simplicity, we define the sum on the left hand side as

$$\chi_i^n \equiv \sum_{m=1}^{n} \lambda_m(\tau) \frac{(\partial_t + v_{i\alpha}\partial_\alpha)^m}{m!}(f_i^0 + \tau F_i) + O(\partial^{n+1}). \tag{A.26}$$

Now, to attain the equations of motion, we then sum both sides over all i and we are left with

$$\sum_i \chi_i^n = \sum_i \Omega_i. \tag{A.27}$$

This simple and concise form is valid for systems with a single conserved quantity, but this can be generalized further to account for systems which require more than one conserved quantity. In general, to acquire the equations of motion for additional conserved quantities, we multiply Eqn. (3.21) by powers of $v_{i\alpha}$ which correspond to the moments in Eqns. (3.22-3.26). We can define a product over these velocities in the form

$$\eta_c = \prod_{j=0}^{c} v_{j\alpha} \tag{A.28}$$

with c representing the order of the moment which is required for any desired conserved quantity. Combining these products with Eqn. (A.27), we then arrive at

$$\sum_i \eta_c \chi_i^n = \sum_i \eta_c \Omega_i. \tag{A.29}$$

This equation is a simple mathematical statement representing the hydrodynamic limit of any lattice Boltzmann method to arbitrary order and conserved quantity.

A.5. Code Sample for Fluctuating Lattice Boltzmann for Diffusion Equation

The following code sample is a simple fluctuating lattice Boltzmann simulation for modeling the diffusion equation. The code presented is publicly available on GitHub [67]. This code requires Wagner's graphical user interface library for C [91].

```
    /*****************************************************************/
/*        D2Q5 Fluctuating Lattice Boltzmann Simulation          */
/*                      Diffusive System                         */
/*            Kyle Strand:  kyle.t.strand@ndsu.edu               */
/*               North Dakota State University                   */
/*                     14 July 2016                              */
/*****************************************************************/

/* New lattice Boltzmann algorithm for fluctuating diffusion.
   Requires Alexander Wagner's Graph Library.
https://www.ndsu.edu/pubweb/~carswagn/GUI/index.html
   The algorithm will follow the following methodology:
1) Forward matrix transformation
2) Collision step - Noise added here
3) Backward matrix transformation
4) Streaming step
  Future edits will be added as they are completed.
*/

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <mygraph.h>
#include <unistd.h>
```

```
#include <time.h>
#include <complex.h>

#define xdim 100                                    // Number of x lattice points
#define ydim 100                                    // Number of y lattice points

double f[5][xdim][ydim], n[xdim][ydim];
double tau[5]={1,0.6,0.6,0.6,0.6};
double n0[2]={120,120}, theta = 1./3.;
int next = 0, Pause = 1, done = 0, repeat = 1, iterations;

//Haloing routine
void Halo() {

  for (int y=0;y<ydim;y++) {      // periodic boundary conditions
    f[1][0][y]=f[1][xdim-2][y];
    f[2][xdim-1][y]=f[2][1][y];
  }
  for (int x=0;x<xdim;x++) {
    f[3][x][0]=f[3][x][ydim-2];
    f[4][x][ydim-1]=f[4][x][1];
  }
}

//Streaming routine
void Stream() {

  memmove(&f[1][1][0],&f[1][0][0],(xdim-1)*ydim*sizeof(double));
  memmove(&f[2][0][0],&f[2][1][0],(xdim-1)*ydim*sizeof(double));
  memmove(&f[3][0][1],&f[3][0][0],(xdim*ydim-1)*sizeof(double));
```

```
  memmove(&f[4][0][0],&f[4][0][1],(xdim*ydim-1)*sizeof(double));
}

//Noise Routine
void Noise(double *M, int i, int j) {

  double noise[5];

  noise[0]=0;
  for (int a=1; a<5; a++) {
    noise[a]=sqrt(n[i][j]*12)*((double)rand()/RAND_MAX - 0.5);
    M[a]=((1.-1./tau[1])*M[a]+1./tau[1]*(sqrt(2*tau[1]-1.) * noise[a]));
  }
}

void init() {                                        // Initializing Eq. Dists

  iterations = 0;
  for (int i = 0; i < xdim; i++) {
    for (int j = 0; j < ydim; j++) {
      n[i][j]=n0[0];
      f[0][i][j] = n[i][j] * (1 - 2*theta);
      for (int a=1; a<5; a++) f[a][i][j]=n[i][j]/2. * theta;
    }
  }
}

//void iteration(double  m[5][5]) {            // Iteration step
void iteration() {
```

```
double M[5];

double m[5][5] = {
  {1.,1.,1.,1.,1.},
  {0.,sqrt(1./theta),-sqrt(1./theta),0.,0.},
  {0,0,0,sqrt(1./theta),-sqrt(1./theta)},
  {0,sqrt(1./(2.*theta)),sqrt(1./(2.*theta)),
  -sqrt(1./(2.*theta)),-sqrt(1./(2.*theta))},
  {-sqrt(2.*theta/(1.-2.*theta)),sqrt((1.-2.*theta)/(2.*theta)),
  sqrt((1.-2.*theta)/(2.*theta)),sqrt((1.-2.*theta)/(2.*theta)),
  sqrt((1.-2.*theta)/(2.*theta))}
};                                    // Transformation matrix*/

for (int i = 0; i < xdim; i++) {
    for (int j = 0; j < ydim; j++) {
      n[i][j] = f[0][i][j] + f[1][i][j] + f[2][i][j] + f[3][i][j] + f[4][i][j];

      for (int a=0; a<5; a++) M[a]=0;

      // Forward transformation
      for (int a=0; a<5; a++) {
        for (int b=0; b<5; b++) {
          M[a] += m[a][b] * f[b][i][j];
        }
      }

    // call noise routine to add noise to moment space functions
    Noise(M, i, j);

    f[0][i][j] = f[1][i][j] = f[2][i][j] = f[3][i][j] = f[4][i][j] = 0;
```

```
      // Back transform
      for (int a=0; a<5; a++) {
          for (int u=0; u<5; u++) {
            if (a == 0) f[a][i][j] += m[u][a] *(1-2*theta)*M[u];
            else f[a][i][j] += m[u][a] * (theta/2) *M[u];
          }
      }
    }
  }

  //Call halo and streaming routines
  Halo();
  Stream();
  iterations++;
}

void GUI() {
  static int Xdim = xdim;
  static int Ydim = ydim;                                    //

  DefineGraphNxN_R("f0",&f[0][0][0], &Xdim, &Ydim, NULL);
  DefineGraphNxN_R("f1",&f[1][0][0], &Xdim, &Ydim, NULL);
  DefineGraphNxN_R("f2",&f[2][0][0], &Xdim, &Ydim, NULL);
  DefineGraphNxN_R("f3",&f[3][0][0], &Xdim, &Ydim, NULL);
  DefineGraphNxN_R("f4",&f[4][0][0], &Xdim, &Ydim, NULL);
  DefineGraphNxN_R("n",&n[0][0], &Xdim, &Ydim, NULL);
  NewGraph();
  StartMenu("Fluctuating D2Q5",1);
  DefineInt("Iterations",&iterations);
```

```
  StartMenu("Parameters",0);
    DefineDouble("n0_x",&n0[0]);
    DefineDouble("n0_y",&n0[1]);
    DefineDouble("Theta",&theta);
    StartMenu("Relaxation Times",0);
      DefineDouble("Tau_0",&tau[0]);
      DefineDouble("Tau_1",&tau[1]);
      DefineDouble("Tau_2",&tau[2]);
      DefineDouble("Tau_3",&tau[3]);
      DefineDouble("Tau_4",&tau[4]);
    EndMenu();
  EndMenu();
  DefineFunction("init",&init);
  SetActiveGraph(0);
  DefineGraph(contour2d_,"Graphs");
  DefineInt("Repeat",&repeat);
  DefineBool("Next",&next);
  DefineBool("Pause",&Pause);
  DefineBool("Close",&done);
  EndMenu();
}

int main(int argc, char *argv[]) {
  int newdata = 1;
  int i;

  init();
  GUI();

  while (done == 0) {
```

```
    Events(newdata);
    DrawGraphs();
    if (next || !Pause) {
      newdata = 1;
      next = 0;
      for (i = 0; i < repeat; i++) {
        iteration();
      }
    }
    else sleep(1);
  }

  return 0;
}
```

A.6. Code Sample Implementing Flipping Operation for Overrelaxation for an Integer Lattice Gas

The subsequent code sample is used as an example of how to practically implement the flipping operation introduced in Sec. 4.3 which allows for over-relaxation in integer lattice gases. Once again, this code sample uses the Wagner's graphics library [91]. In this code, we have also utilized GNU Scientific Library for sampling random numbers from various distributions [40].

```
/*************************************************************/
/*  D1Q3 Diffusive Integer Lattice Gas with Flipping Operation  */
/*            Kyle Strand: kyle.t.strand@ndsu.edu              */
/*               North Dakota State University                 */
/*                      22 March 2022                          */
/*************************************************************/



#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <mygraph.h>
#include <time.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>

#define xdim 320
#define V 3

int n[3][xdim];
double w[3] = {2./3., 1./6., 1./6.};
```

```
double omega = 1.;
double theta = 1./3.;
double N0 = 100.;
double rho[xdim];

int runcount = 1;
int writeamp = 0;
int iterations = 0;
int done = 0;
int Pause = 1;
int next = 0;
int repeat = 1000;

const gsl_rng_type * TYPE;
gsl_rng * RANDOM;

void CalculateDensity() {
  for (int x = 0; x < xdim; x++) {
    rho[x] = n[0][x] + n[1][x] + n[2][x];
  }
}

void Initialize() {
  iterations = 0;
  for (int x = 0; x < xdim; x++) {
    for (int i = 0; i < 3; i++) {
      n[i][x] = gsl_ran_poisson(RANDOM, N0*w[i]*(1+sin(2.*M_PI*x/xdim)));
    }
  }
```

```
  CalculateDensity();
}

  void WriteAmplitude() {
  double Nsum = 0;
  double Nave;
  for (int x = 0; x < xdim; x++) {
    Nsum += rho[x];
  }
  Nave = Nsum/xdim;

  double Aexp;
  double AsumNum = 0;
  double AsumDen = 0;
  double D = (1./omega - 0.5)*1./3.;
  for (int x = 0; x < xdim; x++) {
    AsumNum += sin(2.*M_PI*x/xdim)*rho[x];
    AsumDen += pow(sin(2*M_PI*x/xdim),2);
  }
  Aexp = AsumNum/AsumDen;

  //Output for specific simulation
  char filename[100];
  snprintf(filename,sizeof(filename),"Data/MultiRun/AmpDecayOmega%f.dat",omega);
  FILE *data = fopen(filename, "a");
  fprintf(data, "%i %e\n", iterations, Aexp);
  fclose(data);
}

void Stream() {
```

```
  int tmp1 = n[1][xdim-1];
  int tmp2 = n[2][0];
  memmove(&n[1][1], &n[1][0], (xdim-1)*sizeof(int));
  memmove(&n[2][0], &n[2][1], (xdim-1)*sizeof(int));
  n[1][0] = tmp1;
  n[2][xdim-1] = tmp2;
}

void Collision(double omegac) {
  for (int x = 0; x < xdim; x++) {
    int n0 = 0;
    for (int i = 0; i < 3; i++) {
      //pick a fraction of particles
      int nl = gsl_ran_binomial(RANDOM, omegac, n[i][x]);
      n[i][x] -= nl;
      n0 += nl;
    }

    int nn[3];
    gsl_ran_multinomial(RANDOM, V, n0, w, nn);
    for (int i = 0; i < 3; i++) {
      //add the Multinominally distributed particles back to the n[i][x]
      n[i][x] += nn[i];
    }
  }
}

void FlipN() {
  int flip[V]={0,2,1};
  for (int x = 0; x < xdim; x++) {
```

```
    int nflip[V];
    for (int i = 0; i < V; i++) {
      nflip[i] = n[flip[i]][x];
    }
    for (int i = 0; i<V; i++){
      n[i][x]=nflip[i];
    }
  }
}

void Iteration() {
  double or;
  int reverse=0;
  if (omega>1){
    reverse=1;
    or=2-omega;
  }
  else or=omega;

  if (reverse) FlipN();
  Collision(or);
  Stream();

  if (writeamp == 1) {
    WriteAmplitude();
    if (iterations == 500000) {
      if (runcount < 500) {
        runcount++;
        Initialize();
      } else {
```

```
        omega += 0.25;
        runcount = 1;
        Initialize();
      }
      if (omega > 1) {
        printf("All simulations complete.\n");
        Pause = 1;
      }
    }
  }

  iterations++;
  CalculateDensity();
}

void GUI() {
  static int XDIM = xdim;

  DefineGraphN_R("rho", &rho[0], &XDIM, NULL);
  StartMenu("D1Q3 Diffusive LG", 1);
    DefineInt("Iterations", &iterations);
    DefineDouble("Omega", &omega);
    DefineFunction("Initialize", &Initialize);
    DefineGraph(curve2d_, "Graphs");
    DefineBool("Write Amplitude", &writeamp);
    DefineInt("Run Count", &runcount);
    DefineInt("Repeat", &repeat);
    DefineBool("Next", &next);
    DefineBool("Pause", &Pause);
    DefineBool("Quit", &done);
```

```
  EndMenu();
}

int main(int argc, char *argv[]) {
  int newdata = 1;

  gsl_rng_env_setup();
  TYPE = gsl_rng_default;
  RANDOM = gsl_rng_alloc(TYPE);

  Initialize();
  GUI();

  while (done == 1) {
    Events(newdata);
    DrawGraphs();
    if (next || !Pause) {
      newdata = 1;
      next = 0;
      for (int i = 0; i < repeat; i++) {
        Iteration();
      }
    }
    else sleep(1);
  }
}
```