

LATTICE GASES WITH MOLECULAR DYNAMICS COLLISION OPERATOR

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

By

Mohammad Reza Parsa

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

Major Program:  
Materials and Nanotechnology

June 2018

Fargo, North Dakota

# NORTH DAKOTA STATE UNIVERSITY

Graduate School

---

**Title**

LATTICE GASES WITH MOLECULAR DYNAMICS COLLISION  
OPERATOR

---

**By**

Mohammad Reza Parsa

---

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:

Professor Alexander Wagner

Chair

---

Professor Sylvio May

---

Professor Stuart Croll

---

Professor Rhonda Magel

---

Approved:

July 5, 2018

Date

Professor Erik Hobbie

Department Chair

# ABSTRACT

The purpose of this dissertation is to provide a direct microscopic underpinning for lattice Boltzmann (and lattice gas) methods. Lattice gases are idealized discrete models that conserve mass and momentum. These conservation laws imply, through the formalism of kinetic theory, that on a macroscopic scale these methods recover the continuity and Navier-Stokes equations. As part of the kinetic theory approach, an ensemble average of the lattice gas is taken leading to a lattice Boltzmann equation. These lattice Boltzmann equations can be implemented directly leading to the new how ubiquitous lattice Boltzmann methods. In this dissertation we step away from justifying lattice Boltzmann methods and the ability of recovering suitable macroscopic equations. Rather, their correspondence to coarse-grained Molecular Dynamics simulations is examined and can be cast in the form of a lattice gas evolution equation. We call this lattice gas the Molecular Dynamic Lattice Gas (MDLG). We use this MDLG to derive the exact formulation for lattice Boltzmann equilibrium distributions, relaxation parameters, and fluctuating properties.

## ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Alexander Wagner for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research.

Besides my advisor, I would like to thank the rest of my committee, Prof. May, Prof. Croll, and Prof. Magel, for their insightful comments and encouragement, but also for the hard question which incited me to widen my research from various perspectives.

I thank my fellow labmates Kyle Strand and Kent Ridl for the stimulating discussions and for all the fun we have had in the last four years.

Last but not the least, I would like to thank my parents and my sister for supporting me spiritually throughout writing this dissertation and my life in general.

# TABLE OF CONTENTS

ABSTRACT . . . . .	iii
ACKNOWLEDGEMENTS . . . . .	iv
LIST OF TABLES . . . . .	vii
LIST OF FIGURES . . . . .	viii
1. GENERAL INTRODUCTION . . . . .	1
1.1. Historical Overview . . . . .	1
1.2. The Origins of LGA and LBE Methods . . . . .	4
1.2.1. The Lattice Gas Automata . . . . .	5
1.2.2. The Hydrodynamics of Lattice Gas Automata . . . . .	9
1.2.3. The Lattice Boltzmann Equation . . . . .	11
1.3. Molecular Dynamics . . . . .	16
2. LATTICE GAS WITH MOLECULAR DYNAMICS COLLISION OPERATOR . . . . .	21
2.1. Abstract . . . . .	21
2.2. Introduction . . . . .	21
2.3. Lattice Gas . . . . .	23
2.4. Lattice Gas with Molecular Dynamics Collision Operator . . . . .	25
2.5. MDLG for an Two-Dimensional Lennard-Jones Gas . . . . .	30
2.5.1. Global Equilibrium Distribution . . . . .	32
2.6. Lattice Boltzmann . . . . .	38
2.7. Relation of MDLG Equilibrium Functions to Lattice Boltzmann Equilibria . . . . .	40
2.7.1. Moments of the Equilibrium Distribution . . . . .	44
2.8. Consistent Discretizations . . . . .	50

2.9. Outlook . . . . .	52
2.10. Acknowledgment . . . . .	54
3. ESTABLISHING CORRECT FLUCTUATIONS FOR LATTICE BOLTZMANN METHODS USING THE MOLECULAR DYNAMICS LATTICE GAS METHODS	55
3.1. Abstract . . . . .	55
3.2. Introduction . . . . .	55
3.2.1. The Fluctuating Lattice Boltzmann Equation . . . . .	56
3.2.2. The Multi Relaxation Time Collision Operator and Moment Space Representation . . . . .	57
3.3. Lattice Gas, Lattice Boltzmann and Molecular Dynamics Lattice Gas . . . . .	60
3.4. Molecular Dynamics Lattice Gas . . . . .	64
3.5. Two Particle Correlations . . . . .	65
4. FURTHER RESEARCH . . . . .	72
4.1. Examination of the Fluctuating Behavior of the MDLG Model . . . . .	72
4.2. Formulation of the Collision Operator of the MDLG Model . . . . .	73
4.3. Multi-Phase Behavior . . . . .	73
REFERENCES . . . . .	74
APPENDIX. SUPPLEMENTARY CODE . . . . .	85
A.1. Code for Molecular Dynamic Lattice Gas . . . . .	85
A.2. Code for Mean Square Displacement . . . . .	92
A.3. Code for Velocity Autocorrelation Function . . . . .	97
A.4. Code for Measured Correlators $\langle \delta x_1 \delta x_2 \rangle$ . . . . .	102

# LIST OF TABLES

<u>Table</u>	<u>Page</u>
1.1. Collision rules for FHP model. . . . .	9

# LIST OF FIGURES

Figure	Page
1.1. Lattice gas automata simulation of fluid in two dimensions [1]. . . . .	3
1.2. Lattice Boltzmann simulation of fluid flow [2]. . . . .	3
1.3. Collision process for FHP model. . . . .	6
1.4. Schematic of 9-velocity model for lattice Boltzmann equation. . . . .	16
2.1. Sketch of the MDLG algorithm: a lattice (blue line) is superimposed on the domain of the MD simulation. Particles in the reciprocal lattice cells (indicated by the red boundaries), are associated with the corresponding lattice point. Particles then get associated with the $n_i$ for the $v_i$ which corresponds to their lattice displacement in the time-interval $\Delta t$ . . . . .	26
2.2. Simple thought experiment for lattice gas representation of a particle moving with a constant velocity. Clearly the lattice gas definition of momentum will fluctuate as a function of time even though the underlying MD momentum is conserved. Averaging over all lattice placements, however, will recover the correct momentum.	29
2.3. The numbering convention for the velocities $v_i$ in two dimensions. The central point is 0 and corresponds to velocity $v_0 = (0, 0)$ , and the other velocities are given by the connecting vector between the central point and the lattice point in question. . . . .	31
2.4. (a) Measured velocity correlation function from MD simulation data compared to the exponential fit. (b) Measured mean square displacement from MD simulation data compared to the predicted value according to the Eq. (2.39). . . . .	36
2.5. Measured equilibrium distributions $f_i^{eq}$ as a function of the mean-squared displacement measure $a$ from Eq. (2.36). They are compared to the analytic solution given by Eq. (2.34). We find excellent agreement between the predicted and measured equilibrium distributions. The horizontal lines indicate the value of D2Q9 lattice Boltzmann weights, and the green vertical line indicates the value of $a^2 = 1/6$ for which these weights agree with the MDLG results. . . . .	41
2.6. Pairs of spatial and temporal discretizations $(\Delta x, \Delta t)$ that lead to equilibrium distributions of MDLG that are consistent with the lattice Boltzmann equilibrium distribution $f_i^0(\rho, 0) = w_i$ . Notice the two scaling regimes of $\Delta t \propto \Delta x$ for the ballistic regime for small times and the diffusive regime $(\Delta t)^2 \propto \Delta x$ for large times.	42

2.7.	Dependence of the equilibrium distribution function on an imposed velocity $U$ for $\Delta x = 100 \sigma$ and $\Delta t = 34.16 \tau$ , corresponding to the green line in Fig. (2.5). The prediction of Eq. (2.34) (solid lines) agrees perfectly with the measured averages of the $n_i$ (symbols). These results are compared to the lattice Boltzmann equilibrium distribution for D2Q9 (dashed lines). We find good agreement between the MDLG equilibrium distribution and the LB equilibrium distribution for velocities below about 0.2. . . . .	43
2.8.	The normalized second moment of Eq. (2.59) as a function of the second moment $a^2$ from Eq. (2.36) of the mean squared displacement probability density. The value of $\Psi$ is shown for different 11 different values of $u_x$ between 0 and 0.5. The values for $u_x = 0$ correspond to the bottom points of the graph. For a Galilean-Invariant discrete equilibrium distribution moment the value of $\Psi$ should be independent of $u$ . This is compared to a standard lattice Boltzmann second moment with $\theta = 1/3$ . . . . .	46
2.9.	Third velocity moment given by Eq. (2.60) for an unrestricted velocity set and a velocity set restricted to only a single shell of velocities for 10 equally spaced velocities $u_x$ between 0 and 0.001. . . . .	47
2.10.	Two manifestations of the temperature $\theta$ from the second and third moments, given by Eqs. (2.62) and (2.63). We show the moments for the full velocity set and moments for a velocity set restricted to the first shell (thinner lines). For the restricted velocity set we only see agreement between the two moments for $\theta = 1/3$ corresponding to $a^2 = 1/6$ . . . . .	49
2.11.	The difference of the discrete second moment $m_2$ of Eq. (2.70) minus $a^2$ converges quickly to a constant $1/6$ . . . . .	52
3.1.	Measured correlations $E_{ij}$ for different densities (a) Low density ( $\phi = 0.01$ ) (b) Medium density ( $\phi = 0.08$ ). These data obtained for 99 856 particles with box size of 1 000 and 8 000 and $25 \times 25$ lattice. . . . .	61
3.2.	(a) Measured $\langle v_i v_j \rangle$ for different mean densities: low density (green), medium density (black), medium density for longer simulation (blue), high density (red). (b) Focused on medium density. . . . .	62
3.3.	Measured correlators $\langle \delta x_1 \delta x_2 \rangle$ for different timesteps $\Delta t$ and different mean densities (a) high density (b) medium density (c) low density. . . . .	69

# 1. GENERAL INTRODUCTION

Lattice Boltzmann equation (LBE) and Lattice-gas automata (LGA) are relatively novel methods having originated about three decades ago. However, they have gained the attention of a lot of physicists, mathematicians and engineers related to different disciplines. The reason for the increased interest is the methods have demonstrated high potentials in different complex systems including chemical reactive flows, magneto hydrodynamics, and multi-component and multi-phase fluid hydrodynamics. The methods have proven effective in areas where other methods were simply too difficult to implement or were impractical [3, 4, 5, 6].

Although LBE and LGA methods have been subjected, by different groups of researchers, a true definition of LB density is still controversial. This issue has been solved to a large extent by our new Molecular Dynamic Lattice Gas (MDLG) approach. This study will highlight the achievements of the new model over the traditional understanding of equilibrium density function.

This dissertation has been divided into three chapters. Chapter 1 contains the introduction of the LBE and LGA models, including the details of the hydrodynamics feature. Chapter 2 introduces the MDLG model along with the mathematical derivation [7]. The moments of the new model as well as the equilibrium equation function are discussed in detail. Chapter 3 contains the correct fluctuations for lattice Boltzmann methods using the MDLG approach.

## 1.1. Historical Overview

Prior to the introduction of the LBE and LGA models, similar models used. Broadwell proposed the Boltzmann equation in 1964 to study aerodynamics using only a few discrete velocities [8]. Later in 1973, Hardy, dePazzis, and Pomeau (HPP) introduced a single speed lattice gas cellular automation model to study the statistical mechanical properties of 2-

dimensional fluids including the divergence of 2D transport coefficients. The model was proposed on a 2D square lattice to achieve the study objective. At this point, it must be stressed that both the HPP and Broadwell models were more theoretical than computational [9].

Frisch, Hasslacher, and Pomeau and Wolfram suggested the first 2D LGA model in 1986 [10, 11]. The model was proposed for computational fluid dynamics. This led to the development of a 3D LGA model a few years later. The first proposal for the use of the LBE model was made in 1988 [1]. The proof that simple models, such as LGA and the floating number counterpart the LBE models, can stimulate hydrodynamics to a high degree of accuracy opened a new chapter in computational physics. Some of the main ideas of the models were indeed revolutionary in nature [12, 13].

While the models are relatively recent, LBE and LGA models have already shown their capabilities in different areas of computational physics, including multi-component and multi-phase fluids through porous media, chemical reactive flows, turbulent external flow over structures with complicated geometries, and other complex systems. There are many advantages of LBE and LGA models over traditional models such as solving the Navier-Stokes equation. Having said that, it is important to remain careful when identifying the areas where these methods are more suitable due to the nature of the issues [3, 4, 5, 6, 14].

Before we go on to discuss the technical details relating to the LBE and LGA methods, it is important to first overview the history of the methods. In order to truly appreciate the benefits of the methods, it is important that we find out how the methods evolved over time.

A classic example of fluid dynamics using the LBE and LGA methods is the simulation of the 2D flow past a cylinder. In Fig. (1.1), the LGA is indeed seen to mimic hydrodynamics. It shows von Karman vortex street behind a cylinder. However, the simulation was not quantitative, and rather qualitative in nature. It was He and Doolen who demonstrated the LBE method can accurately simulate hydrodynamics [1, 2].

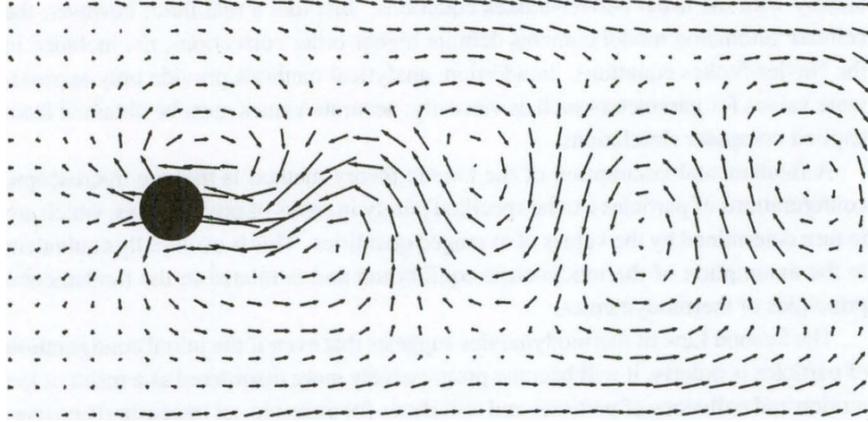


Figure 1.1. Lattice gas automata simulation of fluid in two dimensions [1].

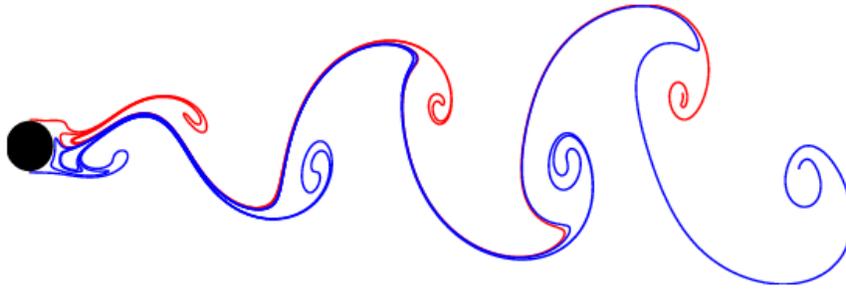


Figure 1.2. Lattice Boltzmann simulation of fluid flow [2].

Fig. (1.2) represents the von Karman vortex street behind a cylinder with a Reynolds number of 100. Different quantities such as lift and drag coefficients had accurately compared and measured existing experimental and numerical results. In addition, the computational speed of the LBE was found to be comparable to traditional methods in solving Navier-Stokes equations. Moreover, accurate results have been obtained by using the LBE method in simulating the 2D flow past impulsively started cylinder having a much higher Reynolds number ( $Re = 10\,000$ ) [15, 16].

Attempts have been made to recover external and internal flows through porous media and low Mach regimes using the LBE method by certain companies such as Exa Corporation, which develops and distributes computer-aided software programs for engineers. There are many other examples of using the LBE method for direct numerical simulation, including turbulent flows [17, 18, 19, 20]. It can be conciliated that LBE and LGA have matured over the years and can be effectively used in many different cases.

## 1.2. The Origins of LGA and LBE Methods

A fluid is really a discrete system having a large number of molecules ( $\sim 10^{23}$ ). A system that contains many molecules can be described either as a hierarchy of kinetic equations such as, the Bogoliubov-Born-Green-Kirkwood-Yvon hierarchy (BBGKY) or the Newtonian equation, which leads to molecular dynamics (MD). These two accurately describe the behavior of particles in fluid.

The molecular chaos assumption allows closing of the BBGKY hierarchy choosing a single equation called the Boltzmann equation for the single particle distribution function. Also, a fluid can also be referred to as a continuum that is described by partial differential equations for fluid velocity, density, and temperature or the Navier-Stokes equations.

Navier-Stokes equations can be conveniently used to solve a number of fluid problems. However, the equations cannot be easily solved under certain conditions including the granular flow and multi-component or multi-phase flows. The interfaces between different phases (water and vapor) and components (water and oil) cause difficulties in solving the equations for multi-phase or multi-component flow.

While computationally one might be able to monitor a few interfaces, tracking a lot of interfaces is not easy. It is impractical to create realistic simulations of fluid systems with composition in homogeneity or densities. This issue can be looked at from different perspectives that is phases of a fluid system or interfaces between different phases or components of a fluid system are thermodynamic effects that result from molecular interaction. In order to solve the Navier-Stokes equations, it is important to know about the equation of state that is mostly unknown at an interface. That is why it's difficult to include thermodynamics in the Navier-Stokes equations in a priori or consistent fashion. Thus, there are fundamental difficulties involved with the system.

While Navier-Stokes equations are difficult to solve, neither the Boltzmann equation nor molecular dynamics are practical alternatives since solving the equations is a formidable task that requires a lot of computation efforts. These three methods are particularly diffi-

cult to solve, however if the hydrodynamic moments are known they may be used in place of LBE and LGA methods. The fact is that hydrodynamics are insensitive to the underlying mesoscopic or microscopic dynamics. The Navier-Stokes equations are in reality statements of conservation laws representing the same conservation laws in microscopic dynamics. Moreover, they represent constitutive relations that represent the irreversible nature of the macroscopic dynamics.

Since the microscopic dynamics details are not a fundamental issue, if only hydrodynamic system behavior is being studied. This leads to the question about what constitutes a minimal mesoscopic or microscopic dynamic system that can provide the required physics at the macroscopic level (thermodynamics, hydrodynamics, etc.). The vital element in such a mesoscopic or microscopic dynamic system is the associated symmetries and conservation laws. In the next sections, we will look at how the LBE and LGA equations were realized [21, 22, 23, 24].

### **1.2.1. The Lattice Gas Automata**

Wolfram published a number of articles in the 1980s showing cellular automata, while having a simple construction, is sufficiently complex to achieve universal computing. At the initial state, the evolution of certain automaton can be implemented at any chosen finite algorithm. However, based on the previous experience of the HPP model [9] whereby it was realized that the 2D square lattice does not have symmetry for hydrodynamics as well as the kinetic theory. Wolfram and Frisch et al. [10, 11, 25, 26] independently discovered a two-dimensional triangular lattice based on a simple cellular automaton can be able to stimulate the Navier-Stokes equations. The scholars proposed the LGA model was developed on a 2D triangular lattice space. The particles possess momenta that allow them to move from one site to another on the lattice in discrete time steps. On a particular site at the lattice, there is either no particle present or one particle is present having a momentum that points to a neighboring site. In this way there are, at the maximum, six particles at any one site simultaneously. As a result, this model is also known as the Frisch, Hasslacher and Pomeau (FHP) or the 6-bit model.

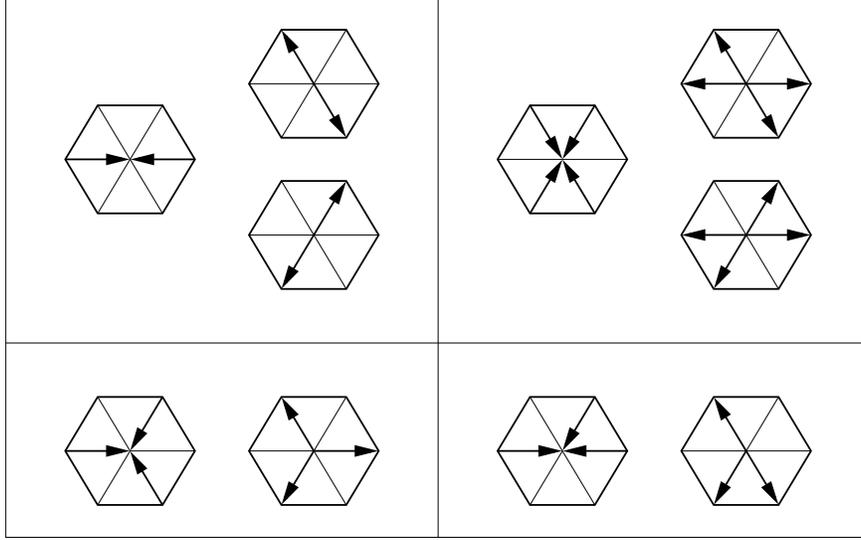


Figure 1.3. Collision process for FHP model.

The development of the LGA model consist of two main steps i.e. collision and streaming. The first step is depicted in Fig. (1.3). For instance, two particles that strike with opposite momenta will tend to rotate their momenta either counter-clockwise or clockwise direction. An important point to note is that the particle momentum, particular number, and the energy are conserved in the collision process both locally and accurately. Since the FHP model has only a single speed, the energy is not an independent variable. It is equivalent to the number of particles. That being said, the energy is an independent variable for multi-speed models.

The LGA evolution equation can be written as

$$n_i(\mathbf{x} + \mathbf{v}_i\delta_t, t + \delta_t) = n_i(\mathbf{x}, t) + \Xi_i \quad (1.1)$$

where  $x$  represents a vector in the lattice space having a lattice constant of  $\delta_x$ ,  $v_i\delta_t$  is a lattice vector,  $n_i$  represents the Boolean particle number having a velocity  $v_i$ ,  $t$  represents discrete time with step size  $\delta_t$ , and  $\Xi_i$  represents that collision operator. Generally, we set both  $\delta_t$  and  $\delta_x$  to unity. The subscript  $i$  represents an index for velocity, as depicted in Fig. (1.3). The  $i$  range is from 1 to 6 for the FHP model. The particles after colliding stream to the next site according to the velocities.

Based on the collision rules depicted in Fig. (1.3), the collision operator  $\Xi_i$  can be written as follows

$$\Xi_i(\{n_i(\mathbf{x}, t)\}) = \sum_{\mathbf{m}, \mathbf{m}'} (m'_i - m_i) \eta_{\mathbf{m}\mathbf{m}'} \prod_{\alpha} n_{\alpha}^{m_{\alpha}} (1 - n_{\alpha})^{(1-m_{\alpha})} \quad (1.2)$$

where  $\mathbf{m}'$  and  $\mathbf{m}$  are possible outgoing and incoming configurations at a given time  $t$  and site  $\mathbf{x}$ , respectively.  $\eta_{\mathbf{m}\mathbf{m}'}$  that is a Boolean random number in time and space determines the transition between  $\mathbf{m}'$  and  $\mathbf{m}$  as per the following normalization condition

$$\sum_{\mathbf{m}'} \eta_{\mathbf{m}\mathbf{m}'} = 1 \quad \forall \mathbf{m}. \quad (1.3)$$

For any states between  $\mathbf{m}'$  and  $\mathbf{m}$ , the Boolean random number  $\eta_{\mathbf{m}\mathbf{m}'}$  must have rotational symmetry. In other words, the random number  $\eta_{\mathbf{m}\mathbf{m}'}$  is invariant if states  $\mathbf{m}'$  and  $\mathbf{m}$  are both subjected to improper or proper rotations simultaneously. It is therefore obvious that the following equation holds for the Boolean numbers  $n_{\alpha}$  and  $m_{\alpha}$

$$n_{\alpha}^{m_{\alpha}} (1 - n_{\alpha})^{(1-m_{\alpha})} = \delta_{n_{\alpha} m_{\alpha}} \quad (1.4)$$

where  $\delta_{n_{\alpha} m_{\alpha}}$  is the Kronecker delta symbol having two indices. Thus, we can write Eq. (1.2) as follows

$$\Xi_i(\{n_i(\mathbf{x}, t)\}) = \sum_{\mathbf{m}, \mathbf{m}'} (m'_i - m_i) \eta_{\mathbf{m}\mathbf{m}'} \delta_{\mathbf{n}\mathbf{m}} \quad (1.5)$$

where  $\delta_{\mathbf{n}\mathbf{m}} = \delta_{n_1 m_1} \delta_{n_2 m_2} \cdots \delta_{n_b m_b}$ .

Eqs. (1.2) or (1.5) are static to a certain extent. For a two-body collision, the following is an example of the collision operator

$$\begin{aligned} \Xi_i^{(2)} = & \eta_r^{(2)} n_{i+1} n_{i+4} \bar{n}_i \bar{n}_{i+2} \bar{n}_{i+3} \bar{n}_{i+5} + \\ & \eta_l^{(2)} n_{i+2} n_{i+5} \bar{n}_i \bar{n}_{i+1} \bar{n}_{i+3} \bar{n}_{i+4} - \\ & (\eta_r^{(2)} + \eta_l^{(2)}) n_i n_{i+3} \bar{n}_{i+1} \bar{n}_{i+2} \bar{n}_{i+4} \bar{n}_{i+5} \end{aligned} \quad (1.6)$$

where  $\bar{n}_i = 1 - n_i$  and represents the complement of  $n_i$ , while  $\eta_r^{(2)}$  and  $\eta_l^{(2)}$  reflect Boolean random numbers. The numbers represent the head-on two-body collision outcomes. They represent the random nature of the two-body collision outcomes. It is obvious that in order for the collision operator to satisfy the complete lattice symmetry group statistically, the following must hold true

$$\langle \eta_r^{(2)} \rangle = \langle \eta_l^{(2)} \rangle \quad (1.7)$$

where  $\langle \cdot \rangle$  represents the ensemble average. The energy, momentum, and the conservation laws of the particle number of the LGA micro-dynamics can be written as follows ( $u$  represents the macroscopic velocity)

$$\sum_i \eta_{mm'} (m'_i - m_i) = 0 \quad (1.8)$$

$$\sum_i \eta_{mm'} (m'_i - m_i) \mathbf{v}_i = 0 \quad (1.9)$$

$$\sum_i \eta_{mm'} (m'_i - m_i) (\mathbf{v}_i - \mathbf{u})^2 = 0. \quad (1.10)$$

The collision in practice can be simulated using different algorithms. One can either make use of a table to look-up or use the logical operation that is indicated by Eq. (1.6). In Fig. (1.3), the collision rule can also be depicted by the collision table that is shown by Table (1.1). In this table, each bit in a binary represents a particle number  $n_i$  from right to left i.e.  $i = 1, 2, \dots, 6$ .

The shortcoming of the table look-up method is the table size, which is  $2^b$ , where  $b$  represents bits of the model. Both the table look-up and the logic operation can be particularly fast on digital computers particularly the dedicated computers [27, 28, 29, 30].

Table 1.1. Collision rules for FHP model.

Input ( $m$ )	Output ( $m'$ )
011011	110110
	101101
001011	100110
010101	101010
001001	010010
	100100

### 1.2.2. The Hydrodynamics of Lattice Gas Automata

The ensemble average that is depicted in Eq. (1.1) resulted in the development of the lattice Boltzmann equation

$$f_i(\mathbf{x} + \mathbf{v}_i \delta_t, t + \delta_t) = f_i(\mathbf{x}, t) + \Omega_i \quad (1.11)$$

where  $\Omega_i = \langle \Xi_i \rangle$  and  $f_i = m \langle n_i \rangle$ , and  $m$  is the particle mass. Moreover, the assumptions are that there are negligible correlations among the colliding particles exist where

$$\langle n_1 n_2 \cdots n_n \rangle = \langle n_1 \rangle \langle n_2 \rangle \cdots \langle n_n \rangle. \quad (1.12)$$

The above approximation is similar to the molecular chaos assumption put forward by Boltzmann (Strosszahlansatz). The lattice Boltzmann collision operator with the molecular chaos approximation can be written in equation form as follows

$$\Omega_i(\{f_i(\mathbf{x}, t)\}) = \sum_{\mathbf{m}, \mathbf{m}'} (m'_i - m_i) \Theta_{\mathbf{m}\mathbf{m}'} \prod_{\alpha} f_{\alpha}^{m_{\alpha}} (1 - f_{\alpha})^{(1 - m_{\alpha})} \quad (1.13)$$

where  $\Theta_{\mathbf{m}\mathbf{m}'} = \langle \eta_{\mathbf{m}\mathbf{m}'} \rangle$  represents the probability of transition from the state  $\mathbf{m}'$  and  $\mathbf{m}$ .

The following equations depict the hydrodynamic moments

$$\sum_i f_i = \rho \quad (1.14)$$

$$\sum_i f_i \mathbf{v}_i = \rho \mathbf{u} \quad (1.15)$$

$$\sum_i f_i (\mathbf{v}_i - \mathbf{u})^2 = 2\rho\epsilon \quad (1.16)$$

where  $\rho$ ,  $\epsilon$  and  $\mathbf{u}$  are the mass density, the internal energy density, and velocity, respectively.

The Eq. (1.11) can be extended in a Taylor series of  $\delta_t$  as per the following equation

$$(\partial_t + \mathbf{v}_i \cdot \nabla) f_i + (\partial_t + \mathbf{v}_i \cdot \nabla)^2 f_i = \Omega_i. \quad (1.17)$$

The  $f_i^{(eq)}$  that represents the equilibrium distribution and is a solution of  $\Omega_i(\{f_i\}) = 0$  must be a Fermi-Dirac distribution since the system is a binary one. This is represented in the following equation

$$f_i^{(eq)} = [1 + \exp(a + b\mathbf{u} \cdot \mathbf{v}_i)]^{-1} \quad (1.18)$$

where coefficients  $a$  and  $b$  represent the function of  $\mathbf{u}^2$  and  $\rho$  in general. Since the coefficients  $a$  and  $b$  in  $f_i^{(eq)}$  cannot be exactly determined, the  $f_i^{(eq)}$  should be expanded in a Taylor series of  $\mathbf{u}$  small velocity expansion or low Mach number. The following hydrodynamic equations are derived from the FHP-LGA model using the small velocity expansion of  $f_i^{(eq)}$  equilibrium position through Chapman-Enskog analysis [27, 11, 31, 32, 33, 34]

$$\partial_t \rho + \nabla \cdot \rho \mathbf{u} = 0 \quad (1.19)$$

$$\partial_t \rho \mathbf{u} + \nabla(g\rho \mathbf{u} \mathbf{u}) = -\nabla P + \nu \nabla^2 \rho \mathbf{u} \quad (1.20)$$

where  $g$  represents a function of  $\rho$ ,

$$P = c_s^2 \rho \left[ 1 - g\left(\frac{\mathbf{u}}{c}\right)^2 \right] \quad (1.21)$$

$$c_s = \frac{c}{\sqrt{2}} \quad (1.22)$$

$$\nu = -\frac{1}{8}(2\lambda^{-1} + 1)c\delta_x \quad (1.23)$$

$$c = \frac{\delta_x}{\delta_t}. \quad (1.24)$$

$\lambda$  represents the eigen value of the liberalized collision operator, and  $c_s$  represents the sound speed [27].

The shortcomings of the LGA hydrodynamics model are clearly evident from the above equations. These include:

1. Factor  $g(\rho)$  does not have unity because the Galilean invariance is not valid.
2. Large fluctuations in  $n_i$  means that the simulations are intrinsically noisy.
3. Increasing Reynolds number  $Re$  is not easy.
4. Equation of state is dependent upon  $\mathbf{u}^2$ , which is not physical
5. A spurious conserved quantity is present due to the simple symmetry of LGA

The above issues can be solved by using more complex LGA models, or other alternative models such as the LBE method [35, 36, 37, 38].

### 1.2.3. The Lattice Boltzmann Equation

The LBE model equations have evolved from the Boolean counterparts i.e. LGA model. Eq. (1.11) represents the original LBE model that replaces the corresponding hydrodynamics LGA model. It was later realized that a simple relaxation model can replace the collision operator and that it can be linearized. More recently, it has been found out that the LBE is a special discretized form of the continuous Boltzmann equation. For simplifying the equation, the following analysis shows the application of the LBE with the Bhatnagar-Gross-Krook (BGK) approximation [13, 39, 40].

We can write the Boltzmann BGK equation as follows

$$\frac{Df}{Dt} + \frac{1}{\lambda}f = \frac{1}{\lambda}f^{(eq)} \quad (1.25)$$

in which,  $D_t = \partial_t + \mathbf{v} \cdot \nabla$  represents the Lagrangian derivation along the microscopic  $\mathbf{v}$  velocity,  $\lambda$  represents the relaxation time during collision,  $f^{(eq)}$  represents the Maxwell-Boltzmann distribution function, and  $f = f(\mathbf{x}, \mathbf{v}, t)$  represents the single particle distribution function and

$$f^{(eq)} = \frac{\rho}{(2\pi k_B T)^{d/2}} e^{-\frac{(\mathbf{v}-\mathbf{u})^2}{2\theta}} \quad (1.26)$$

where,  $d$  represents the dimension of the space;  $k_B$ ,  $T$ , and  $m$  represent the Boltzmann constant, temperature, and particle mass, respectively, and  $\rho$ ,  $\mathbf{u}$ ,  $\theta = k_B T/m$  represent the macroscopic density of the mass, velocity, and normalized temperature. The following equation represents the macroscopic variables that are the moments of the distribution function  $f$  based on the  $\mathbf{v}$  velocity

$$\int f d\mathbf{v} = \rho \quad (1.27)$$

$$\int f \mathbf{v} d\mathbf{v} = \rho \mathbf{u} \quad (1.28)$$

$$\int f (\mathbf{v} - \mathbf{u})^2 d\mathbf{v} = 2\rho\theta. \quad (1.29)$$

We can integrate Eq. (1.25) over a time interval  $\delta_t$  as follows

$$f(\mathbf{x} + \mathbf{v}\delta_t, \mathbf{v}, t + \delta_t) = e^{(-\delta_t/\lambda)} \left[ f(\mathbf{x}, \mathbf{v}, t) + \frac{1}{\lambda} \int_0^{\delta_t} e^{(t'/\lambda)} f^{(eq)}(\mathbf{x} + \mathbf{v}t', \mathbf{v}, t + t') dt' \right]. \quad (1.30)$$

The following equation can be derived if we assume that  $f^{(eq)}$  is smooth locally, and  $\delta_t$  is small and we neglect the terms of the order  $O(\delta_t^2)$  or smaller that is used in the Taylor

expansion of the right hand side of the above equation

$$f(\mathbf{x} + \mathbf{v}\delta_t, \mathbf{v}, t + \delta_t) - f(\mathbf{x}, \mathbf{v}, t) = -\frac{1}{\tau} \left[ f(\mathbf{x}, \mathbf{v}, t) - f^{(eq)}(\mathbf{x}, \mathbf{v}, t) \right] \quad (1.31)$$

in which,  $\tau = \frac{\lambda}{\delta_t}$  represents the dimensionless relaxation of time. We can expand the equilibrium distribution function  $f^{(eq)}$  to achieve a Taylor series in  $\mathbf{u}$ . Thus, we can obtain the following equation by retaining the Taylor expansion up to  $\mathbf{u}^2$

$$f^{(eq)} = \frac{\rho}{(2\pi\theta)^{d/2}} e^{-\frac{\mathbf{v}^2}{2\theta}} \left[ 1 + \frac{(\mathbf{v} \cdot \mathbf{u})}{\theta} + \frac{(\mathbf{v} \cdot \mathbf{u})^2}{2\theta^2} - \frac{\mathbf{u}^2}{2\theta} \right]. \quad (1.32)$$

The above equation represents a sufficient expansion for deriving the Naiver Stoke equations. In addition, the following moment integral must be analyzed exactly to derive the Navier-Stokes equations

$$\int f^{(eq)} \mathbf{v}^k d\mathbf{v} \quad (1.33)$$

where  $0 \leq k \leq 3$  for the isothermal models. In the above equation, the following integral is present that can be evaluated by using the Gaussian-type quadrature

$$I = \int e^{(-\frac{\mathbf{v}^2}{2\theta})} \psi(\mathbf{v}) d\mathbf{v} = \sum_i W_i e^{(-\frac{\mathbf{v}_i^2}{2\theta})} \psi(\mathbf{v}_i) \quad (1.34)$$

where  $\mathbf{v}_i$  and  $W_i$  represent the discrete velocities (or abscissas) of the quadrature and weights, respectively, and  $\psi(\mathbf{v})$  is a polynomial in  $\mathbf{v}$ . The hydrodynamic moments depicted in Eqs. (1.27–1.29) can be calculated by quadrature as well

$$\sum_i f_i = \rho \quad (1.35)$$

$$\sum_i f_i \mathbf{v}_i = \rho \mathbf{u} \quad (1.36)$$

$$\sum_i f_i (\mathbf{v}_i - \mathbf{u})^2 = 2\rho\theta \quad (1.37)$$

where  $f_i = f_i(\mathbf{x}, t) = W_i(\mathbf{x}, \mathbf{v}_i, t)$ . In order to derive the LBE models, we shall use the 9-velocity isothermal LBE model on square lattice space as an example. The derivation of Eq. (1.31) on a discretized phase time and space along with proper equilibrium distribution function results in the Navier-Stokes equations. In order to derive the 9-velocity LBE model, we have used a Cartesian coordinate system, and set  $\psi(\mathbf{v}) = v_x^m v_y^n$ . The integral Eq. (1.34) thus becomes

$$I = (2\theta)^{\frac{(m+n+2)}{2}} I_m I_n \quad (1.38)$$

where

$$I_m = \int_{-\infty}^{\infty} e^{-\sigma^2} \sigma^m d\sigma \quad (1.39)$$

$$\sigma = \frac{v_x}{\sqrt{2\theta}} \quad \text{or} \quad \frac{v_y}{\sqrt{2\theta}}. \quad (1.40)$$

The third-order Hermite formal is therefore the best choice to assess  $I_m$  in order to derive the 9-velocity LBE model, or  $I_m = \sum_{i=1}^3 \omega_i \sigma_i^m$ . The corresponding weights  $\omega_i$  and the three abscissas  $\sigma_i$  of the quadrature are represented in equation form as follows

$$\sigma_1 = \sqrt{\frac{3}{2}} \quad (1.41)$$

$$\sigma_2 = 0 \quad (1.42)$$

$$\sigma_3 = -\sqrt{\frac{3}{2}} \quad (1.43)$$

$$\omega_1 = \frac{\sqrt{\pi}}{6} \quad (1.44)$$

$$\omega_2 = 2\frac{\sqrt{\pi}}{6} \quad (1.45)$$

$$\omega_3 = \frac{\sqrt{\pi}}{6}. \quad (1.46)$$

Moreover, the integral of Eq. (1.38) becomes

$$I = 2\theta \left[ \omega_2^2 \psi(\mathbf{0}) + \sum_{i=1}^4 \omega_1 \omega_2 \psi(\mathbf{v}_i) + \sum_{i=5}^8 \omega_1^2 \psi(\mathbf{v}_i) \right] \quad (1.47)$$

where  $\mathbf{v}_i$  represents the zero-vector velocity for  $i = 0$ ,  $\alpha = 1 - 4$  represents the vectors of  $\sqrt{3\theta}(\pm 1, 0)$  and  $\sqrt{3\theta}(0, \pm 1)$ , while  $\alpha = 5 - 8$  represents vectors of  $\sqrt{3\theta}(\pm 1, \pm 1)$ . An important point to note is that the above quadrature is true for  $(m + n) \leq 5$ . Nine discrete velocities i.e.  $\{\mathbf{v}_i \mid i = 0, 1, \dots, 8\}$  represent momentum space. In order to obtain the 9-velocity model, the configuration space is accordingly discretized into a square lattice having a lattice constant of  $\delta_x = \sqrt{3\theta}\delta_t$ . It is important to note that the temperature  $\theta$  is not physically significant here since only the isothermal model is relevant. Thus, we can choose  $\delta_x$  to be a significant quantity, and so,  $\sqrt{3\theta} = c = \delta_x/\delta_t$ , or  $\theta = c_s^2 = c^2/3$ , where  $c$  represents the models' sound speed. After comparing Eqs. (1.34) and (1.47), we can recognize the weights that are defined in the Eq. (1.34)

$$W_i = 2\pi\theta \exp\left(\frac{\mathbf{v}_i^2}{2\theta}\right)\omega_i \quad (1.48)$$

$$\omega_\alpha = \begin{cases} 4/9 & \alpha = 0 \\ 1/9 & \alpha = 1, 2, 3, 4 \\ 1/36 & \alpha = 5, 6, 7, 8. \end{cases} \quad (1.49)$$

The equilibrium distribution function of the 9-velocity model is depicted by the equations

$$f_i^{(eq)} = W_i f^{(eq)}(\mathbf{x}, \mathbf{v}_i, t) = \omega_i \rho \left[ 1 + \frac{3(\mathbf{v}_i \cdot \mathbf{u})}{c^2} + \frac{9(\mathbf{v}_i \cdot \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} \right] \quad (1.50)$$

$$v_\alpha = \begin{cases} (0, 0) & \alpha = 0 \\ (\cos \theta_\alpha, \sin \theta_\alpha)c & \alpha = 1, 2, 3, 4 \\ (\cos \theta_\alpha, \sin \theta_\alpha)\sqrt{2}c & \alpha = 5, 6, 7, 8 \end{cases} \quad (1.51)$$

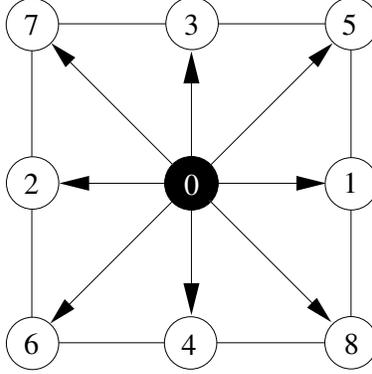


Figure 1.4. Schematic of 9-velocity model for lattice Boltzmann equation.

and  $\theta_i = (i - 1)\pi/2$  for  $i = 1 - 4$  and  $(i - 5)\pi/2 + \pi/4$  for  $i = 5 - 8$ , as is depicted in Fig. (1.4). The following equation represents the Navier-Stokes equation that has been derived from the LBE model

$$\rho \partial_t \mathbf{u} + \rho \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla P + \rho \nu \nabla^2 \mathbf{u} \quad (1.52)$$

where  $P = c_s^2 \rho$  represents the equation of ideal gas,  $c_s = c/\sqrt{3}$  represents the sound speed, and  $\nu = (2\tau - 1)c\delta_x/6$  represents the viscosity for the 9-velocity model. Similarly, the 2D 6- and 7-velocity and 3D 27-velocity LBE models can be derived [41, 42, 43, 44, 45].

The above derivation equations show that the discretion of space is achieved by discretizing momentum space in a way that allows lattice structure in configuration space to be simultaneously accomplished. The discretization of configuration space is possible by the momentum space. This discretization of configuration space and momentum space can be done independently that has two immediate consequences: significant improvement of the Reynolds number in the LBE hydrodynamic simulations and the arbitrary mesh grids.

### 1.3. Molecular Dynamics

Molecular Dynamics is considered as an important theoretical tool for physicists and chemists which allows them to simulate the microscopic dynamic of different complex systems. In MD solving the equations of motion for all individual atoms describes the detailed microscopic behavior of the given system. Moreover, MD was used for measuring equilibrium properties by sampling from a statistical ensemble of the N-body system.

For starting the simulation, MD requires initial structure of the system and inter-particle interaction rules or force field which allows MD to predict accurately the behavior of the system. Ab initio molecular dynamics is the alternative approach for force field which is not the subject of this dissertation [46, 47, 48, 49].

The initial structure includes size of the simulation box, number, type, and the initial distance of particles. In this research, a two dimensional square box with one type of particle was used for simulation. Particles were equilibrated in the box before the actual simulation and eventually results were collected in the form of position and velocity in two dimensions for each individual particle.

Moreover, particles interacted through the standard Lennard-Jones potential

$$V(x) = 4\epsilon \left[ \left(\frac{\sigma}{x}\right)^{12} - \left(\frac{\sigma}{x}\right)^6 \right]. \quad (1.53)$$

Eq. (1.53) represents the Lennard-Jones potential energy where  $x$  is the distance between two particles,  $\epsilon$  is the interaction strength, and  $\sigma$  represents the spatial scaling. In general the first and second terms in Eq. (1.53) describe the repulsive and the attractive dipole-dipole interaction respectively. It should be noticed that in the Lennard-Jones potential  $\sigma$  and  $\epsilon$  are units for length and energy. By assuming  $m$  as a unit for mass,  $\tau$  or the time scale can be shown by Eq. (1.54)

$$\tau = \sqrt{\frac{m\sigma^2}{\epsilon}}. \quad (1.54)$$

In this dissertation, Molecular Dynamics simulation was performed using the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) [50] package for 100 000 particles in a two dimensional box with the length of  $1000 \sigma$ . A homogeneous distribution of particles was used for the initial point with the kinetic energy corresponding to a temperature of 50 in the unit of Lennard-Jones. The simulation was run for 21 000 000 and 3 000 000 and then the early time data of 1 000 000 was discarded, to ensure that we were only probing the

equilibrium dynamics. The following code shows a simple LAMMPS implementation of the current simulation [51, 52, 53, 54].

```
#2d Lennard-Jones
dimension          2
units              lj
atom_style        atomic
timestep          0.0001
read_data         initialStructure.input
mass              1 1.0
change_box        all boundary pp pp pp units box
velocity          all create 50 87287 loop geom
pair_style        lj/cut 2.5
pair_coeff         1 1 1.0 1.0 2.5
neigh_modify      delay 0 every 20 check no
fix               1 all nvt temp 50.0 50.0 50.0
thermo            10000
thermo_style      custom step temp pe ke
run               1000000
reset_timestep    0
dump              1 all custom 1000 dump.relax id type xs ys zs vx vy vz
run               200000000
```

*“dimension”* command specifies the dimensionality of the simulation. This command should be used before defining the simulation box. For simplicity, two dimensional simulation was used in this dissertation.

*“unit”* command defines the unit of simulation for the output files. Since we have hard spherical particles in our system, lj (Lennard-Jones) style was used for the output file. It should be noted that lj style uses unit-less quantities which means  $\epsilon$ ,  $\sigma$ ,  $m$ , and  $k_B$  are equal to 1 and all masses, distances, and energies are a magnitude of these fundamental values.

*“atom\_style”* command describes the style of atoms and types of interactions during the simulation. atomic style represents our system since there is no interaction or bond between particles.

*“timestep”* command shows the time-step size in the unit of  $\tau$ .

*“read\_data”* command helps LAMMPS to read the initial information for starting the simulation. This command could be used several times during the simulation to make changes in the structure of system. We used this command for defining the initial simulation box with the length of  $1000\sigma$  and the initial position of particles. A complicated system with multiple types of molecules, bonds, and angles requires a more sophisticated input file.

*“mass”* command specifies the mass of each type of particles. For example in this case, the mass of particle type 1 is equal to 1 unit of Lennard-Jones.

*“velocity”* command sets the velocity of all particles by using a random number generator at a specific temperature (in our case 50).

*“dump”* command stores requested data for further analysis. In this simulation, the coordination and velocity of all particles in two dimensions were saved at each 1000 time step.

*“pair\_style”* command computes the standard Lennard-Jones potential (Eq. (1.53)) for less than specific distance. In the current model,  $2.5\sigma$  was used for the cutoff.

*“neigh\_modify”* command allows us to build a new neighbor list after specific time steps. This command is vital for the previous cutoff command.

*“fix”* command generates velocities of all particles from a requested ensemble. Canonical ensemble with the temperature of 50 unit of Lennard-Jones was set for the current model.

*“thermo”* and *“thermo\_style”* commands print requested information on the screen.

*“run”* command continues the simulation for the certain time steps.

## 2. LATTICE GAS WITH MOLECULAR DYNAMICS COLLISION OPERATOR

### 2.1. Abstract

We introduce a lattice gas implementation that is based on coarse-graining a Molecular Dynamics (MD) simulation. Such a lattice gas is similar to standard lattice gases, but its collision operator is informed by an underlying MD simulation. This can be considered an optimal lattice gas implementation because it allows for the representation of any system that can be simulated with MD. We show here that equilibrium behavior of the popular lattice Boltzmann algorithm is consistent with this optimal lattice gas. This comparison allows us to make a more accurate identification of the expressions for temperature and pressure in lattice Boltzmann simulations which turn out to be related not only to the physical temperature and pressure but also to the lattice discretization. We show that for any spatial discretization we need to choose a particular temporal discretization to recover the lattice Boltzmann equilibrium.

### 2.2. Introduction

Lattice Boltzmann methods are an important computational tool that is most commonly employed to simulate hydrodynamic systems [55], and it has been adapted to address many complex phenomena from turbulence [56, 57] over multi-phase and multi-component flow [6, 40, 58, 59, 60] to pore-scale simulations of porous media [61, 62] and simulations of immersed boundaries [63, 64]. It derives its power from an underlying mesoscopic description that ensures exact mass and momentum conservation. The exact physical meaning of the lattice Boltzmann densities, however, remains poorly understood.

---

The content of this chapter has been published in “M. Reza Parsa and Alexander J. Wagner. Lattice gas with molecular dynamics collision operator. *Physical Review E*, 96(1): 013314, 2017.”

The lattice Boltzmann method was derived as a theoretical tool for the analysis of lattice gas methods [13]. Lattice gas methods consist of particles moving on a lattice with velocities that connect neighboring sites. After the particles have moved a stochastic collision step rearranges the particles. If these collisions conserve both the number of particles and the total momentum there will be a hydrodynamics limit for mass and momentum conservation equation. The introduction of a hexagonal instead of a square lattice by Frisch, Hasslacher and Pomeau [10] recovered the necessary isotropy to allow the momentum equation to be related to the Navier-Stokes equation.

These lattice gas models had some deficiencies, one unfavorable feature was a large and essentially uncontrolled amount of noise that required a significant amount of averaging. To derive the Navier-Stokes equations from the lattice gas dynamics a theoretical ensemble average was performed, leading to a lattice Boltzmann representation. Higuera then proposed to simulate the ensemble averaged lattice Boltzmann evolution equation directly, and thereby avoid the need to average results of the lattice gas equation [65, 66]. The collision operation of this first lattice Boltzmann method could be mapped one-to-one to the lattice gas and shared some of the positive features of the lattice gas, like the existence of an H-theorem with unconditional stability, and also some of its deficiencies like velocity dependent viscosities.

It was then realized that there existed much more freedom in the choice of the collision operator, and in particular the relaxation towards a local equilibrium function, often called the Bhatnagar-Gross-Krook (BGK) approach, allowed the full recovery of the Navier-Stokes equation to second order [67].

At this time a second approach to derive the lattice Boltzmann equation directly from the continuous Boltzmann equation with a BGK collision operator gained popularity [40]. Over the years several different local equilibrium distributions have been proposed, and currently the most popular method is a standard form of a second order expansion in terms of velocities.

Typically these lattice Boltzmann methods are validated by their ability to recover the Navier-Stokes equation. Here, however, we want to establish a relation to an underlying Molecular Dynamics simulation. For any Molecular Dynamics simulation, we can bin the particles into lattice cells corresponding to the lattice Boltzmann lattice. We can then observe where the particles in cell  $x$  migrate to after a time  $\Delta t$ , and associate these particles with a lattice velocity vector  $v_i = x(t) - x(t + \Delta t)$ . These particles will collect at their new lattice cells. After another timestep  $\Delta t$  these particles are re-distributed to new lattice sites, and can be associated with new lattice velocities. We call this representation of the MD simulation Molecular-Dynamics-lattice-gas (MDLG). This redistribution can be understood to be an effective MDLG-collision operator. In some very fundamental sense this is the collision operator that the lattice Boltzmann approach is trying to mimic. The purpose of this paper is to understand the physical meaning of the lattice Boltzmann densities in terms of this fundamental MDLG representation.

The paper is organized as follows: we first introduce a general idea of a lattice gas and then derive a new lattice gas which consists of a coarse-graining of an underlying MD simulation. We then apply this general idea to a specific MD simulation of a Lennard Jones gas in two dimensions. We analyse the equilibrium properties of the associated MDLG method and show that we are able to predict its mathematical form analytically. We then introduce the lattice Boltzmann method and compare the equilibrium properties of the MDLG method to the lattice Boltzmann equilibrium. We show that there are particular choices for the coarse-graining time and space discretization that lead to equilibria that are compatible with the lattice Boltzmann results.

### **2.3. Lattice Gas**

At its very basis a lattice gas consists of particles, all located on lattice points, that move with a lattice velocity  $v_i$ . What we mean by lattice velocity is that if  $x$  is a lattice point, so is  $x + v_i$ . There are  $n_i(x, t)$  particles at time  $t$  at position  $x$  moving with velocity  $v_i$ . The evolution consists of two steps. A collision step that redistributes the particles at

the lattice point  $x$  to different velocities is shown below

$$n_i^*(x, t) = n_i(x, t) + \Xi_i(\{n_j(x, t)\}) \quad (2.1)$$

where the collision operator  $\Xi_i$  is a function of all the particles and their velocities that are located at lattice point  $x$  at time  $t$ . This collision operator will ensure that none of the locally conserved quantities are changed in the collision process. These locally conserved quantities will vary, depending on the desired physical system that one wants to model. In the majority of cases one will ensure mass and momentum conservation. Early lattice gases restricted the number of particles to at most one per velocity  $v_i$  at a lattice site, and the velocity vectors all had the same length, ensuring that mass and energy conservation were synonymous. Each conserved quantity will lead to a corresponding hydrodynamic equation. Most applications focused on the fluid flow, and the key hydrodynamic equations to recover were the continuity and Navier-Stokes equations. Energy conservation is often abandoned in favor of an isothermal condition for many practical applications. Local mass and momentum densities are defined as

$$\rho = \sum_i n_i \quad (2.2)$$

$$\rho u_\alpha = \sum_i v_{i\alpha} n_i \quad (2.3)$$

and the conservation of these quantities then implies

$$\sum_i \Xi_i = 0 \quad (2.4)$$

$$\sum_i v_{i\alpha} \Xi_i = 0. \quad (2.5)$$

For the new kind of lattice gas collision operator proposed in section 2.4 we will see that mass conservation of Eq. (2.4) is indeed fulfilled, but the momentum conservation of Eq.

(2.5) is not strictly obeyed. Because the new algorithm is based on MD, however, the algorithm conserves momentum rigorously. Its representation of momentum through Eq. (2.3), however, is inexact. This collision is then followed by a streaming step

$$n_i(x + v_i, t + 1) = n_i^*(x, t) \quad (2.6)$$

where the particles move to the lattice site indicated by the velocity index  $i$ , *i.e.* they move from  $x$  to  $x + v_i$ . The full evolution equation for these densities can then be written as

$$n_i(x + v_i, t + 1) = n_i(x, t) + \Xi_i. \quad (2.7)$$

Of course to make this description complete we need to define the collision operator. Originally lattice gases were defined such that there could be at most one particle for each  $n_i$  [10]. For the purpose of this paper, however, we will make no such restriction. Instead we investigate a collision operator that is defined by an underlying molecular dynamic simulation.

#### 2.4. Lattice Gas with Molecular Dynamics Collision Operator

In principle most systems of interest for a lattice gas (LG) simulation could be simulated using a Molecular Dynamics (MD) approach as well. MD is a standard tool which follows classical particle trajectories for particles interacting with a pair-potential by numerically integrating Newton's equation of motion.

To construct a lattice gas method from a molecular dynamics simulation we overlay a lattice onto our MD simulation. The number of particles in each reciprocal lattice cell around the lattice position  $x$  then corresponds to the lattice gas density  $\rho(x)$ , as shown in Fig. (2.1). If we then choose a time-step  $\Delta t$  we can observe where particles ending up in cell  $x$  came from. The number of particles moving from cell  $x - v_i$  to cell  $x$  then corresponds to the lattice gas occupation number  $n_i(x, t)$ . What is important to note is that this resulting lattice gas model is fundamentally correct in the sense that will obey the continuity and Navier-Stokes equations simply because the molecular dynamics simulation does so.

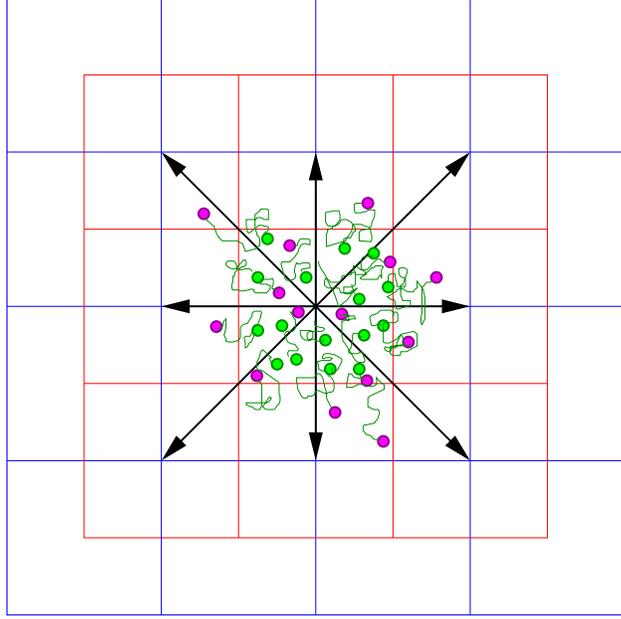


Figure 2.1. Sketch of the MDLG algorithm: a lattice (blue line) is superimposed on the domain of the MD simulation. Particles in the reciprocal lattice cells (indicated by the red boundaries), are associated with the corresponding lattice point. Particles then get associated with the  $n_i$  for the  $v_i$  which corresponds to their lattice displacement in the time-interval  $\Delta t$ .

The initial condition for a set of  $N$  particles in a finite container with periodic boundary conditions is given by their initial positions  $x_i(0)$  and velocities  $v_i(0)$ . These particles then interact through an interaction pair-potential that we take here to only depend on the distance between the two particles:  $V_{ij} = V(|x_i(t) - x_j(t)|)$ . The MD simulation then provides (to good accuracy) the trajectories  $x_i(t)$  which solve Newton's second law

$$\frac{dx_i(t)}{dt} = v_i(t) \quad (2.8)$$

$$\frac{dv_i(t)}{dt} = -\frac{\partial}{\partial x_i} \left( \frac{1}{2} \sum_{j \neq i} V_{ij} \right). \quad (2.9)$$

We now superimpose a lattice onto the computational domain of the MD simulation. For simplicity we can imagine a cubic lattice of lattice spacing  $\Delta x$ , although any lattice will do here. Let us define a function that determines if a particle resides in a specific cell of the

reciprocal lattice associated with a lattice point  $x$

$$\Delta_x(x') = \begin{cases} 1 & x_\alpha < x'_{i\alpha}(t) \leq x_\alpha + \Delta x \ \forall \alpha \in \{x, y, z\} \\ 0 & \text{otherwise.} \end{cases} \quad (2.10)$$

Next we pick a time step  $\Delta t$ . We can now determine the lattice displacement each particle originally residing in a lattice point  $x$  experiences. The set of all such displacements makes up the minimal set of lattice velocities  $v_i$  for our lattice gas method, and the number of particles associated with this displacement makes up the lattice gas densities  $n_i(x, t)$ . We define

$$n_i(x, t) = \sum_j \Delta_x(x_j(t)) \Delta_{x-v_i}(x_j(t - \Delta t)). \quad (2.11)$$

This definition ensures that the particle numbers  $n_i(x, t)$  will undergo a streaming step given by Eq. (2.6). For any given MD simulation we then know all  $n_i(x, t)$ . From Eq. (2.7) we see that the collision operator is then given by

$$\Xi_i = n_i(x + v_i, t + 1) - n_i(x, t). \quad (2.12)$$

This fully defines the MDLG algorithm, a lattice gas with a collision operator that is defined through an underlying MD simulation. In some sense this is an ideal lattice gas model that can handle even the most complex situations, *i.e.* anything that can be addressed by MD, correctly. The key question is whether this collision operator can be reduced to some stochastic collision operator that only depends on the local  $n_i(x, t)$ . Clearly this will only be the case for very simple systems since the MDLB collision operator contains information about temporal and spatial correlations of the underlying MD algorithm and can, in principle, deal with many complex phenomena like liquid-gas-solid coexistence, large varieties of transport parameters, including phenomena at high Knudsen, high Mach, and/or high Reynolds numbers, which we don't expect to be accessible to a simple lattice gas algorithm of Eq. (2.7) with a local collision operator. Such extensions will be subject of future research, but are outside the scope of the current paper.

The local number of particles in lattice cell  $x$  at time  $t$  is give by

$$N(x, t) = \sum_j \Delta_x(x_j(t)). \quad (2.13)$$

This is consistent with the lattice gas definition of the local density because

$$\begin{aligned} N(x, t) &= \sum_i n_i(x, t) \\ &= \sum_i \sum_j \Delta_x(x_j(t)) \Delta_{x-v_i}(x_j(t - \Delta t)) \\ &= \sum_j \Delta_x(x_j(t)). \end{aligned} \quad (2.14)$$

The last equality follows because

$$\sum_i \Delta_{x-v_i}(x_j(t - \Delta t)) = 1 \quad (2.15)$$

*i.e.* every particle will be found somewhere on the lattice. Note that we have not yet restricted the velocity set. We will use as many velocities as needed. Mass conservation of Eq. (2.4) is clearly fulfilled since

$$\begin{aligned} &\sum_i \Xi_i \\ &= \sum_i [n_i(x + v_i, t + 1) - n_i(x, t)] \\ &= \sum_i \left\{ \sum_j \Delta_{x+v_i}[x_j(t + \Delta t)] \Delta_x(x_j(t)) \right. \\ &\quad \left. - \sum_k \Delta_x(x_k(t)) \Delta_{x-v_i}(x_k(t - \Delta t)) \right\} \\ &= \rho(x, t) - \rho(x, t) \\ &= 0. \end{aligned} \quad (2.16)$$

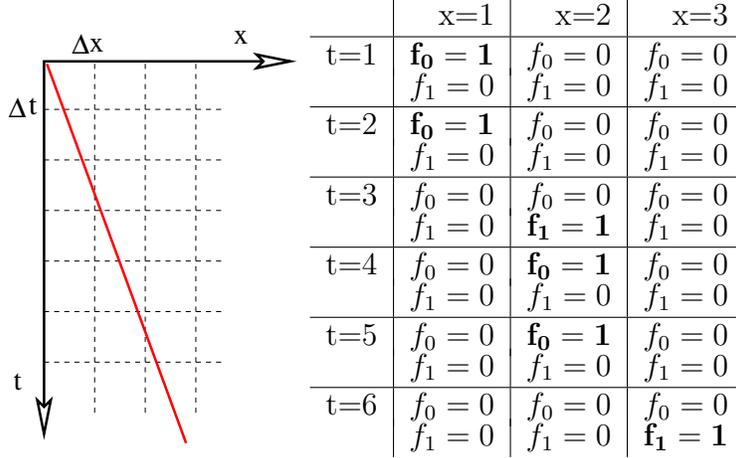


Figure 2.2. Simple thought experiment for lattice gas representation of a particle moving with a constant velocity. Clearly the lattice gas definition of momentum will fluctuate as a function of time even though the underlying MD momentum is conserved. Averaging over all lattice placements, however, will recover the correct momentum.

The definition of momentum in the lattice gas sense is typically defined as

$$N(x, t)u(x, t) = \sum_i n_i(x, t)v_i. \quad (2.17)$$

However, relating this to the underlying momentum of the MD simulation is not exact, as can be seen in the example of a single MD particle moving with a velocity less than a lattice velocity shown in Fig. (2.2). The correspondence could be made exact if we were to introduce an average over all possible placements of the lattice. Such an average would make no difference to the global equilibrium distribution, which is the main focus of the remaining paper. We therefore avoid this additional complication for the current paper.

Similarly, momentum conservation of Eq. (2.5) is only exact if we introduce an average over lattice placements

$$\sum_i v_{i\alpha}\Xi_i \neq 0 \text{ in general.} \quad (2.18)$$

Of course this does not mean that there is a problem with momentum conservation. Instead the problem arises due to the definition of momentum through measured mass transfer between sites for a fixed lattice.

Despite the apparent lack of momentum conservation the MDLG collision rules are still correct, even without the averaging, since the underlying MD simulation respects momentum conservation. As such the apparent violation of momentum conservation of the MDLG model is benign. We reserve a closer examination of this averaged lattice gas implementation for a followup paper.

The key question is then whether the collision operator (Eq. (2.12)) can take the form of Eq. (2.1), *i.e.* a stochastic collision operator that only depends on the current local occupation numbers  $n_j(x, t)$ . Since there is a whole ensemble of MD simulations that is consistent with a set of  $n_j(x, t)$ , and these different MD simulations will lead to different collision terms, it is clear that there can be no exact mapping. However, it is reasonable to hope that we will be able to construct a stochastic lattice gas collision operator that is statistically equivalent to the collision operators for the ensemble of corresponding molecular dynamics simulations. Establishing this is not a trivial task, and we will focus on the easier problem of showing that these collision operators are consistent with the equilibrium behavior of the lattice gas. In the next section we will present the lattice Boltzmann method which conceptually represents the ensemble average of a lattice gas method. Given the complexity of the task we focus in this paper on examining for which, if any, discretizations the MDLG and the standard lattice Boltzmann method give an equivalent equilibrium behavior.

## 2.5. MDLG for an Two-Dimensional Lennard-Jones Gas

As a test case we use for our underlying MD simulations particles interacting with the standard Lennard-Jones interaction potential, which is given by

$$V(x) = 4\epsilon \left[ \left( \frac{\sigma}{x} \right)^{12} - \left( \frac{\sigma}{x} \right)^6 \right]. \quad (2.19)$$

	47	41	33	27	31	39	45
	43	23	17	11	15	21	37
	35	19	7	3	5	13	29
	26	10	2	0	1	9	25
	30	14	6	4	8	20	36
	38	22	16	12	18	24	44
	46	40	32	28	34	42	48

Figure 2.3. The numbering convention for the velocities  $v_i$  in two dimensions. The central point is 0 and corresponds to velocity  $v_0 = (0, 0)$ , and the other velocities are given by the connecting vector between the central point and the lattice point in question.

The interaction strength is controlled by  $\epsilon$  and the spatial scaling by  $\sigma$ . If  $m$  is the mass of a particle we can construct the time-scale

$$\tau = \sqrt{\frac{m\sigma^2}{\epsilon}}. \quad (2.20)$$

We performed a molecular dynamics simulation using LAMMPS for 100 000 particles in a two-dimensional box with the length of  $1000\sigma$ , corresponding to a nominal volume fraction of 0.0785 if we approximate the particles as circles with diameter  $\sigma$ . The system was initialized with a homogeneous distribution of particles with a kinetic energy corresponding to a temperature of 50 in the LJ units defined above. This corresponds to a gas at a high temperature, dense enough so that there are a significant number of collisions. The temperature is well above the critical temperature for a liquid-gas coexistence of  $T_c = 1.3120(7)$  [68]. We ran an simulation of an equilibrium system with a time-step of  $0.0001\tau$  for 3 000 000 time-steps, *i.e.* up to  $\tau = 300$ . Early time data of 1 000 000 timesteps was discarded, to ensure that we were only probing the equilibrium dynamics.

We then analyse the resulting MD trajectories to obtain the resulting averaged MDLG occupation numbers  $n_i(x, t)$  from Eq. (2.11). We should note here that results for different mean velocities  $U$  can be obtained simply by using a lattice displaced by  $-U \Delta t$  for the MDLG analysis. It is therefore not necessary to re-run the MD simulations to examine different mean velocities.

The first information to be gleaned from this is the resulting velocity set for the  $v_i$ . For small times  $\Delta t$ , only the nearest neighbors have non-negligible contributions, but as  $\Delta t$  is increased more densities get populated. We identify the velocities  $v_i$  using the numbering scheme shown in Fig. (2.3). So for  $\Delta t \rightarrow 0$  only  $v_0 - v_8$  will have contributions. These velocities form a complete shell around the central point  $v_0$ . Most standard lattice Boltzmann methods work hard to make due with this minimal velocity set. This comes at some cost, the most important one is that only one temperature,  $\theta = 1/3$  in lattice units, is allowable in lattice units to recover the correct viscous stress tensor (see Eq. (2.49)). For larger  $\Delta t$  particle will travel further and a larger velocity set is required. The average occupation numbers are given by the global equilibrium distribution. The next subsection discusses how these equilibrium distribution can be obtained analytically.

### 2.5.1. Global Equilibrium Distribution

For a system in thermal equilibrium, sufficient averaging will give an equilibrium distribution

$$\begin{aligned} f_i^{eq} &= \langle n_i \rangle \\ &= \sum_j \langle \Delta_x(x_j(t)) \Delta_{x-v_i}(x_j(t - \Delta t)) \rangle. \end{aligned} \quad (2.21)$$

We can numerically approximate this equilibrium density by averaging the values of the  $n_i$  from Eq. (2.11) over the whole lattice and for the duration of the simulation. For a given MD simulation the results will depend both on the lattice spacing  $\Delta x$  and on the time step  $\Delta t$ . As mentioned above the MD simulations considered in this paper deal with fairly hot gases, that should be reasonably well approximated by an ideal gas.

To theoretically calculate this expectation value we assume that the particles are uniformly distributed, so only the displacement during the time-interval  $\Delta t$  will enter the averaging

$$\delta x = x(t) - x(t - \Delta t). \quad (2.22)$$

The key for our averaging will then be the probability density of finding such a displacement  $P(\delta x)$ , and this allows us to write the average as

$$\begin{aligned} f_i^{eq} &= \frac{\rho^{eq}}{(\Delta x)^d} \int dx \int d(\delta x) \Delta_x(x) \Delta_{x-v_i}(x - \delta x) P(\delta x) \\ &= \frac{\rho^{eq}}{(\Delta x)^d} \int dx \int d(\delta x) \Delta_x(x) \Delta_x(x + v_i - \delta x) P(\delta x) \\ &= \rho^{eq} \int d(\delta x) P(\delta x) W(v_i - \delta x) \\ &= \rho^{eq} \int d(\delta x) P(\delta x + v_i) W(\delta x) \end{aligned} \quad (2.23)$$

where  $W(x)$  is the d-dimensional wedge function defined as

$$\begin{aligned} W(x) &= \prod_{\alpha=1}^d W_\alpha(x) \\ &= \prod_{\alpha=1}^d \left( 1 - \frac{|x_\alpha|}{\Delta x} \right) \Theta \left( 1 - \frac{|x_\alpha|}{\Delta x} \right) \end{aligned} \quad (2.24)$$

where  $\Theta$  is the Heaviside function and  $\alpha$  denotes cartesian coordinate index.

For very short times  $\Delta t$ , *i.e.* times shorter than the mean free time between two collisions, particles simply undergo ballistic motion. The velocity distribution of the particles is given by the Maxwell-Boltzmann distribution

$$P(v) = \frac{1}{(2\pi k_B T)^{d/2}} \exp \left( -\frac{(v - u)^2}{2k_B T} \right). \quad (2.25)$$

With this, and neglecting any collisions between the particles, we get for the mean squared

displacement in one dimension

$$\langle(\delta x_\alpha)^2\rangle^{bal} = k_B T (\Delta t)^2. \quad (2.26)$$

The probability density distribution for the displacement is then given by

$$P^{bal}(\delta x) = \frac{1}{(2\pi k_B T)^{d/2} (\Delta t)^d} \exp\left(-\frac{(\delta x - u\Delta t)^2}{2k_B T (\Delta t)^2}\right). \quad (2.27)$$

For times much longer than the mean free time particles undergo multiple collisions and instead of following a ballistic motion they will diffuse. If we call the self-diffusion constant  $D$  we then have

$$\langle(\delta x_\alpha)^2\rangle^{dif} = 2D(\Delta t). \quad (2.28)$$

This implies that the probability density of the displacement is given by

$$P^{dif}(\delta x) = \frac{1}{(4\pi(\Delta t)D)^{d/2}} \exp\left(-\frac{(\delta x - u\Delta t)^2}{4D(\Delta t)}\right). \quad (2.29)$$

Now since both limiting displacements are given by Gaussian distributions it is reasonable to expect that the intermediate probabilities are also well approximated by a Gaussian and, if we know the mean squared displacement in one dimension  $\langle(\delta x_\alpha)^2\rangle$  (and assume isotropy), we get for the probability density

$$P(\delta x) = \frac{1}{(2\pi\langle(\delta x_\alpha)^2\rangle)^{d/2}} \exp\left(-\frac{(\delta x - u\Delta t)^2}{2\langle(\delta x_\alpha)^2\rangle}\right). \quad (2.30)$$

In all of these cases the probability density distribution factorizes

$$P(\delta x) = \prod_{\alpha=1}^d P_\alpha(\delta x) \quad (2.31)$$

where

$$P_\alpha(\delta x) = \frac{1}{\sqrt{2\pi\langle(\delta x_\alpha)^2\rangle}} \exp\left(-\frac{(\delta x_\alpha - u_\alpha\Delta t)^2}{2\langle(\delta x_\alpha)^2\rangle}\right) \quad (2.32)$$

and we can write Eq. (2.23) as a product of Gaussian integrals

$$f_i^{eq} = \rho^{eq} \prod_{\alpha=1}^d \int d(\delta x_\alpha) P(\delta x_\alpha + v_{i\alpha}) W_\alpha(\delta x). \quad (2.33)$$

The solution is given by

$$\frac{f_i^{eq}}{\rho^{eq}} = \prod_{\alpha=1}^d f_{i,\alpha}^{eq} \quad (2.34)$$

where

$$\begin{aligned} f_{i,\alpha}^{eq} = & N \left( e^{-\frac{(u_{i,\alpha}-1)^2}{2a^2}} - 2e^{-\frac{u_{i,\alpha}^2}{2a^2}} + e^{-\frac{(u_{i,\alpha}+1)^2}{2a^2}} \right) \\ & + \frac{u_{i,\alpha} - 1}{2} \left[ \operatorname{erf}\left(\frac{u_{i,\alpha} - 1}{\sqrt{2a}}\right) - \operatorname{erf}\left(\frac{u_{i,\alpha}}{\sqrt{2a}}\right) \right] \\ & + \frac{u_{i,\alpha} + 1}{2} \left[ \operatorname{erf}\left(\frac{u_{i,\alpha} + 1}{\sqrt{2a}}\right) - \operatorname{erf}\left(\frac{u_{i,\alpha}}{\sqrt{2a}}\right) \right] \end{aligned} \quad (2.35)$$

where

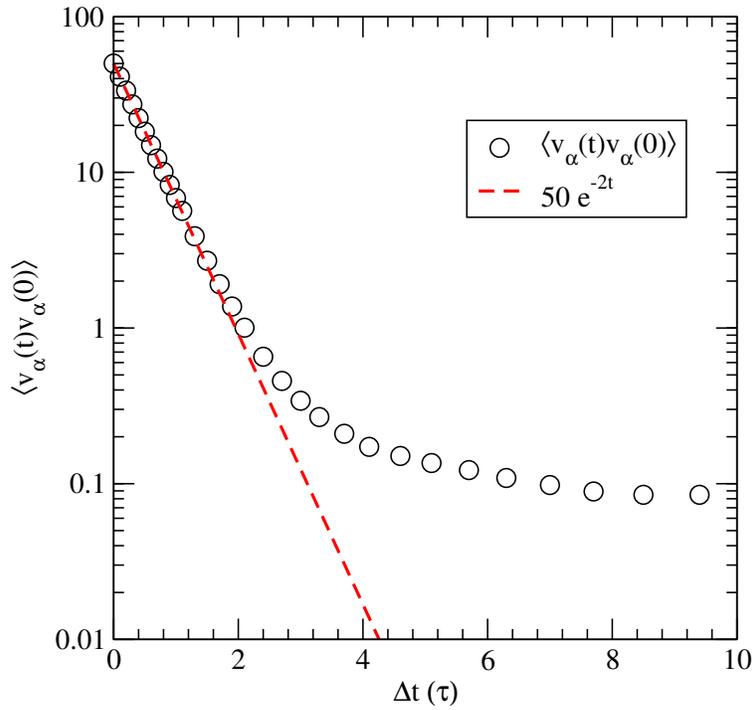
$$a^2 = \frac{\langle(\delta x_\alpha)^2\rangle}{(\Delta x)^2} \quad (2.36)$$

$$N = \frac{a}{\sqrt{2\pi}} \quad (2.37)$$

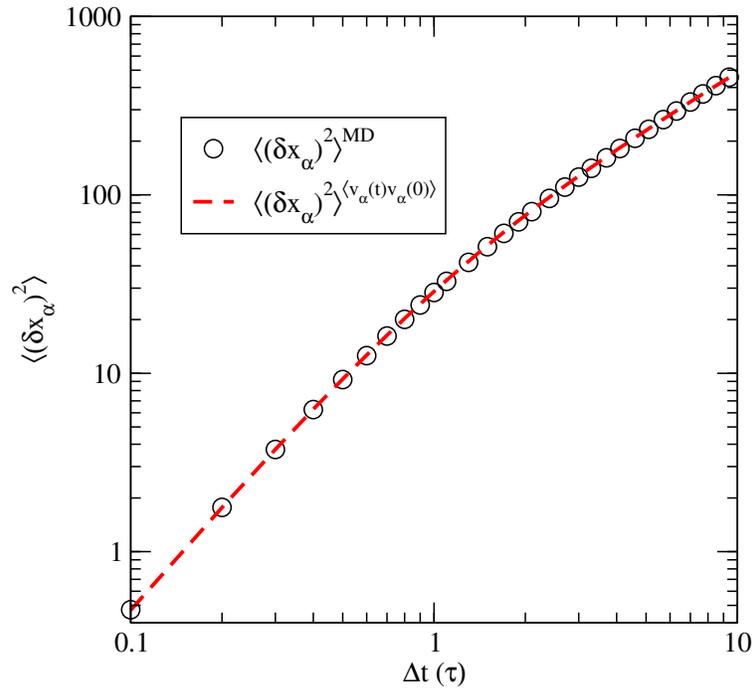
$$u_{i,\alpha} = v_{i\alpha} - u_\alpha. \quad (2.38)$$

This is a lattice equilibrium distribution function derived from first principles. At first glance it looks different than other lattice equilibrium distributions, and we will examine its relation to know equilibrium distribution functions below.

First we need to fully define the lattice equilibrium distribution. To do so we need to obtain the mean square displacement  $\langle(\delta x_\alpha)^2\rangle$ . In general the mean square displacement can be measured in our MD simulations, but this would require us to consider a whole function of



(a)



(b)

Figure 2.4. (a) Measured velocity correlation function from MD simulation data compared to the exponential fit. (b) Measured mean square displacement from MD simulation data compared to the predicted value according to the Eq. (2.39).

$\Delta t$  as in input parameter. For our simple ideal gas system we can obtain a simpler dependence on a single parameter by expressing it in terms of the velocity correlation function as

$$\langle(\delta x_\alpha)^2\rangle = 2 \int_0^t dt' (t - t') \langle v_\alpha(t') v_\alpha(0) \rangle. \quad (2.39)$$

For gases this velocity correlation function is typically well approximated by a simple exponential decay. There is also a long range  $1/t$  contribution to the velocity correlation function for our two dimensional system, but for the times  $\Delta t$  that are of interest here this divergent contribution does not yet contribute noticeably

$$\langle v_\alpha(t) v_\alpha(0) \rangle = k_B T \exp\left(-\frac{t}{\tau}\right). \quad (2.40)$$

For our system we compare this prediction of an exponentially decaying velocity correlation function to the measured correlation function in Fig. (2.4)(a). We see that for early times we see good agreement with this prediction for  $\tau = 0.5(0)$ . We also see the long time-tale typical for a two dimensional system, which we ignore here. This is justified below [69, 70, 71, 72, 73, 74]. Then the mean squared displacement can be predicted according to Eq. (2.39) as

$$\langle(\delta x_\alpha)^2\rangle = 2k_B T \tau^2 \left( e^{-\frac{t}{\tau}} + \frac{t}{\tau} - 1 \right). \quad (2.41)$$

We show that this prediction recovers the measured mean squared displacement well in Fig. (2.4)(b). Deviations resulting from the long time tails of the velocity correlation function only show up for later times and larger displacements considered in this paper, which justifies our ignoring these long time tails here.

This fully completes the definition of the MDLG equilibrium function in the case of gases. To verify our results we compare a numerically measured equilibrium distribution

with the theoretically predicted one for different discretizations. The results are shown in Fig. (2.5). The agreement between our theoretical results and the experimental ones is excellent.

In the next section we introduce the lattice Boltzmann method and then examine the relation of this MDLG equilibrium function with existing lattice equilibrium distribution functions derived for lattice Boltzmann methods.

## 2.6. Lattice Boltzmann

Lattice Boltzmann methods were derived as ensemble averages of lattice Boltzmann methods. The variables in a lattice Boltzmann method are distribution functions

$$f_i = \langle n_i \rangle_{neq} \quad (2.42)$$

where the  $\langle \dots \rangle_{neq}$  represents a non-equilibrium ensemble average over microscopic lattice gas states. Taking the same ensemble average, the evolution equation for these lattice Boltzmann densities derives from the underlying lattice gas evolution Eq. (2.7)

$$f_i(x + v_i, t + 1) = f_i(x, t) + \Omega_i \quad (2.43)$$

where the collision operator  $\Omega_i = \langle \Xi_i \rangle$  is a deterministic function of all the densities at lattice point  $x$ . We will investigate later if this collision operator can be, at least approximately and for some suitable discretization, be cast in the standard BGK form typically employed for lattice Boltzmann simulations. This question is a crucial first step if one wants to relate lattice Boltzmann to an explicit molecular system which will be represented by our MDLB algorithm.

This standard LB collision operator is a first order BGK approximation and can be written as

$$\Omega_i = \sum_j \Lambda_{ij} (f_j^0 - f_j) \quad (2.44)$$

where  $f_j^0$  is a local equilibrium distribution which depends only on the locally conserved quantities

$$\rho = \sum_i f_i \quad (2.45)$$

$$\rho u_\alpha = \sum_i v_{i\alpha} f_i. \quad (2.46)$$

Although other collision operators are also being used [75, 76, 77, 78] and it is a longer term goal of the MDLG method to help identify which of these collision operators are most realistic.

To ensure that the lattice Boltzmann equation reproduces the continuity and Navier-Stokes equations in the hydrodynamic limit it is necessary that the equilibrium distribution matches the first four (apart sometimes from a  $u^3$  term) velocity moments of the Maxwell-Boltzmann distribution

$$\sum_i f_i^0 = \rho \quad (2.47)$$

$$\sum_i (v_{i\alpha} - u_\alpha) f_i^0 = 0 \quad (2.48)$$

$$\sum_i (v_{i\alpha} - u_\alpha)(v_{i\beta} - u_\beta) f_i^0 = \rho \theta \delta_{\alpha\beta} \quad (2.49)$$

$$\sum_i (v_{i\alpha} - u_\alpha)(v_{i\beta} - u_\beta)(v_{i\gamma} - u_\gamma) f_i^0 = Q_{\alpha\beta\gamma} \quad (2.50)$$

where  $Q_{\alpha\beta\gamma}$  should be zero. For velocity sets including only one shell we have  $v_{i\alpha} \in \{-1, 0, 1\}$ . For these velocity sets these moments overconstrain the equilibrium distributions. In particular we have  $v_{i\alpha}^3 = v_{i\alpha}$ , which couples the first and the third moment. This is a key source of Galilean invariance violations in lattice Boltzmann [79]. These moments can only be reconciled for the special choice of

$$\theta = 1/3 \quad (2.51)$$

and the third order tensor  $Q_{\alpha\beta\gamma} = \rho u_\alpha u_\beta u_\gamma \ll 1$  which is assumed to be small because  $u < 0.1$  in typical situations. The equilibrium distribution is typically given in terms of an expansion in terms of the local velocity  $u$  up to second order as

$$f_i^0 = \rho w_i \left( 1 + \frac{v_{i\alpha} u_\alpha}{\theta} + \frac{1}{2} \frac{v_{i\alpha} u_\alpha v_{i\beta} u_\beta}{\theta^2} - \frac{1}{2} \frac{u_\alpha u_\alpha}{\theta} \right) \quad (2.52)$$

where the weights  $w_i$  depend on the velocity set and summation over repeated Greek indices is implied. In this article we focus on the question whether this form of an equilibrium distribution is compatible with a concrete MDLB implementation.

This collision operator together with the local equilibrium distribution implies mass and momentum conservation

$$\sum_i \Omega_i = 0 \quad (2.53)$$

$$\sum_i v_{i\alpha} \Omega_i = 0 \quad (2.54)$$

which is consistent with the typical conditions for lattice gases of Eqs. (2.4) and (2.5).

In the following we will examine the MDLG method for the example of a hot, dilute gas. For this lattice gas we examine the resulting distribution functions and see under which circumstances this lattice gas can reproduce (to some approximation) the lattice Boltzmann method equilibrium distribution Eq. (2.52).

## 2.7. Relation of MDLG Equilibrium Functions to Lattice Boltzmann Equilibria

We are now in a position to predict for which set of parameters  $\Delta x, \Delta t$ , if any, we can recover the traditional form of the lattice Boltzmann equilibrium from our MDLB algorithm. Most lattice Boltzmann methods use a limited velocity set that corresponds to a single shell in Fig. (2.3). For our two dimensions this corresponds to 9 velocities. The corresponding equilibrium distribution is typically given as a second order polynomial in the velocities, as we have presented earlier in Eq. (2.52). For the two dimensional D2Q9 lattice Boltzmann

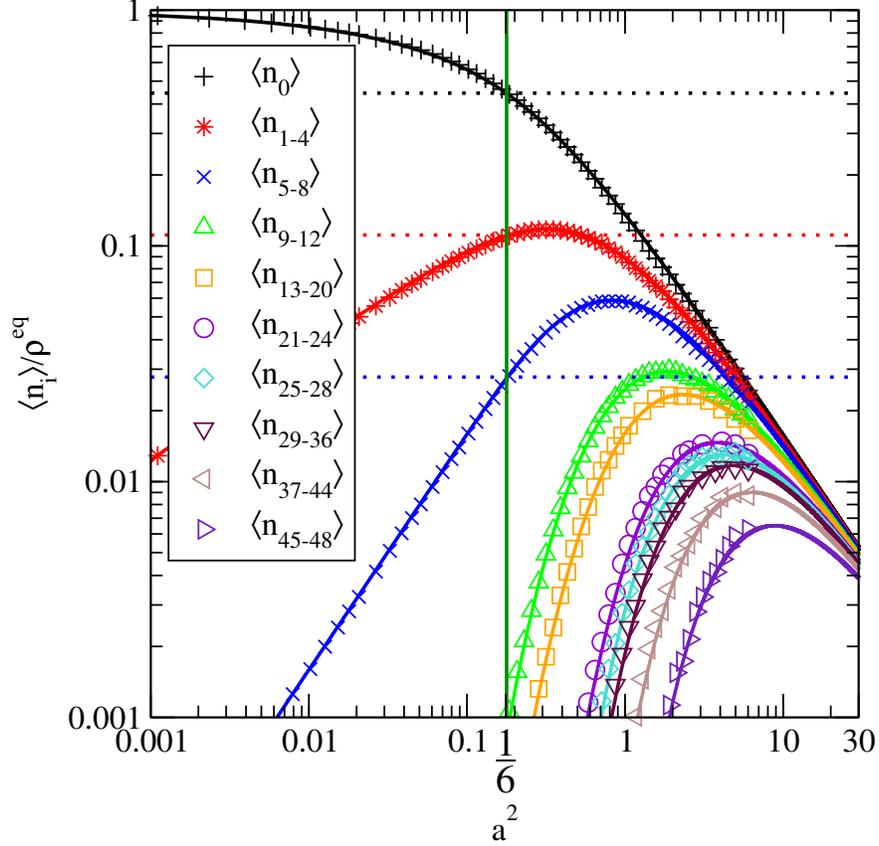


Figure 2.5. Measured equilibrium distributions  $f_i^{eq}$  as a function of the mean-squared displacement measure  $a$  from Eq. (2.36). They are compared to the analytic solution given by Eq. (2.34). We find excellent agreement between the predicted and measured equilibrium distributions. The horizontal lines indicate the value of D2Q9 lattice Boltzmann weights, and the green vertical line indicates the value of  $a^2 = 1/6$  for which these weights agree with the MDLG results.

method we consider here the weights  $w_i$  in Eq. (2.52) are given by

$$w_0 = 4/9 \tag{2.55}$$

$$w_{1-4} = 1/9 \tag{2.56}$$

$$w_{5-8} = 1/36 \tag{2.57}$$

where the velocity indices correspond to the numbering of Fig. (2.3).

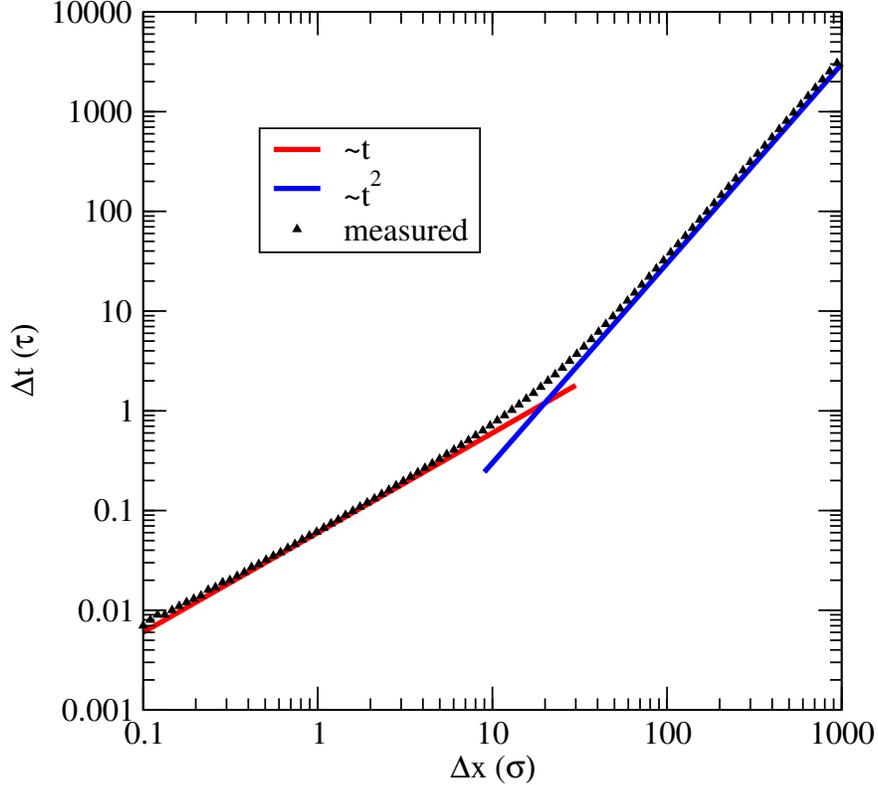


Figure 2.6. Pairs of spatial and temporal discretizations  $(\Delta x, \Delta t)$  that lead to equilibrium distributions of MDLG that are consistent with the lattice Boltzmann equilibrium distribution  $f_i^0(\rho, 0) = w_i$ . Notice the two scaling regimes of  $\Delta t \propto \Delta x$  for the ballistic regime for small times and the diffusive regime  $(\Delta t)^2 \propto \Delta x$  for large times.

In Fig. (2.5) we show the lattice Boltzmann weights  $w_i$  as horizontal lines. To match the MDLG and LB equilibria we require

$$f_i^{eq}(a^2)/\rho^{eq} = w_i. \quad (2.58)$$

This is a over-determined system of equations.

Fortuitously the solutions for the three distinct  $w_i$  for a D2Q9 lattice Boltzmann give the same value for  $a^2 \approx 1/6$  to very good approximation. This is shown as the green vertical line in Fig. (2.5).

This suggests that matching lattice Boltzmann and MD simulations would likely benefit from using the conditions where the  $f_i^{eq}$  match up, and the methodology explained

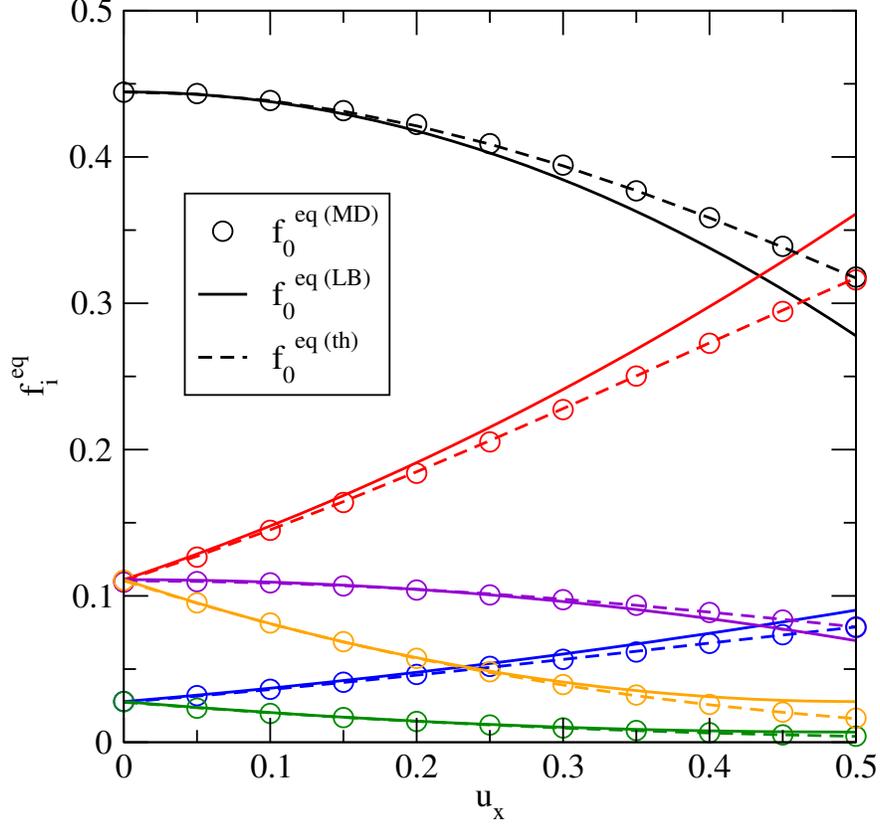


Figure 2.7. Dependence of the equilibrium distribution function on an imposed velocity  $U$  for  $\Delta x = 100 \sigma$  and  $\Delta t = 34.16 \tau$ , corresponding to the green line in Fig. (2.5). The prediction of Eq. (2.34) (solid lines) agrees perfectly with the measured averages of the  $n_i$  (symbols). These results are compared to the lattice Boltzmann equilibrium distribution for D2Q9 (dashed lines). We find good agreement between the MDLG equilibrium distribution and the LB equilibrium distribution for velocities below about 0.2.

above would give guidance on the appropriate time step  $\Delta t$  for a given spatial discretization  $\Delta x$ . Given a  $\Delta x$  can numerically solve Eq. (2.58) for  $\Delta t$  for a system with zero mean momentum. This is shown in Fig. (2.6). We find that there is close agreement between the solutions for different velocities  $v_i$ . Corresponding to the transition from ballistic to diffusive regime around  $\Delta t = \tau$  that we saw in Fig. (2.4)(b) we also see a transition here from a  $\Delta t \propto t$  regime for  $\Delta t \ll 1$  to a  $\Delta t \propto t^2$  regime for  $\Delta t \gg 1$ . We expect this relation that gives  $\Delta t$  in terms of  $\Delta x$  to be valuable when one tries to generate a coarse-graining transition between an MD and LG region in a multi-scale numerical method.

So far we have only matched the equilibrium distribution at zero velocity. The theory contains the mean velocity as  $u$  in Eq. (2.35). For the measurements we could set up different simulations for mean velocities using the algorithm described above in Eq. (2.11). It is more practical, however, to move the Grid instead or equivalently use Galilean transformed particle positions of  $\hat{x}_i(t) = x_i(t) + ut$  instead. Using this approach to find the equilibrium distribution for different mean velocities  $u$  we show our comparison between the measured discrete equilibria (Eq. (2.11)), their theoretical prediction (Eq. (2.34)), and the lattice Boltzmann equilibrium distribution (Eq. (2.52)) in Fig. (2.7). For small velocities  $|u| < 0.1$  we find good agreement between all three densities. This is the relevant range, as lattice Boltzmann is only considered reliable for small enough velocities. The agreement between the measured and predicted MDLG equilibrium distributions continues to be excellent throughout the whole regime. Note, that the agreement would continue to be excellent for larger velocities. We would only need to adapt the velocity set we consider as velocities with magnitude larger than 0.5 are the same as velocities with magnitude larger than 0.5 plus an additional integer lattice displacement.

### 2.7.1. Moments of the Equilibrium Distribution

The key property of the equilibrium distribution in kinetic theory are the velocity moments. For the derivation of the Navier-Stokes equation moments up to third order are required. It is therefore helpful to examine the moments of the discrete MDLG equilibrium distribution, compare them to the expected moment for a lattice Boltzmann equilibrium distribution and examine how these moments relate to the continuous velocity distribution function for the MD simulation.

Let us spend a moment considering these different concepts, since they are not usually clearly separated in a LB derivation. Firstly we have the velocity distribution of the MD simulation, given by Eq. (2.25). The moments of this velocity distribution are usually used as a rational for constructing LB equilibrium distributions such that the relevant moments

of the discrete LB equilibrium distribution function match those of the continuous Maxwell Boltzmann distribution. This is the rational behind the moment Eqs. (2.47–2.50) for the lattice Boltzmann equilibrium function.

The moments of the discrete MDLG method are not a priori constrained to obey such a constraint, and indeed we don't expect such an agreement for two reasons. First the underlying displacement probability densities (Eq. (2.30)) are in general different from the displacement probability densities (Eq. (2.27)) directly related to the Maxwell-Boltzmann distribution (Eq. (2.25)). Second there is no reason to believe that the averaging procedure of Eq. (2.33) will preserve the moments in general.

Now let us consider the first three velocity moments specifically. The zeroth moment relates to the total mass. Since the algorithm conserves mass exactly we expect all three approaches to agree on this moment. Indeed, as we saw in Eq. (2.16) mass is clearly conserved, and consequently this moment will agree for all of the above approach.

The first moment relates to the local momentum. Even if the MDLG approach does not locally conserve the momentum, the averaged momentum of the equilibrium distribution remains exact. This simply follows from the fact that this discrete moment corresponds to the net mass flow through the lattice. Even though this flow can be inexact at any instance in time (since particles may not cross a boundary despite the fact that they are moving) on average the count of particles crossing boundaries has to give the exact mass current.

Let us next consider the second moment. This second moment in the lattice Boltzmann approach (Eq. (2.49)) is related ideal gas equation of state  $p = \rho\theta$ . We calculate the second moment of our discrete MDLG equilibrium distribution

$$\Psi_{\alpha\beta}(a, u) = \frac{\sum_i f_i^{eq} (v_{i\alpha} - u_\alpha)(v_{i\beta} - u_\beta)}{\rho^{eq} a^2}. \quad (2.59)$$

For an equilibrium distribution that obeys the lattice Boltzmann moment (Eq. (2.49)) with a temperature  $\theta = a^2$  this expression would give exactly one. For the MDLG equilibrium

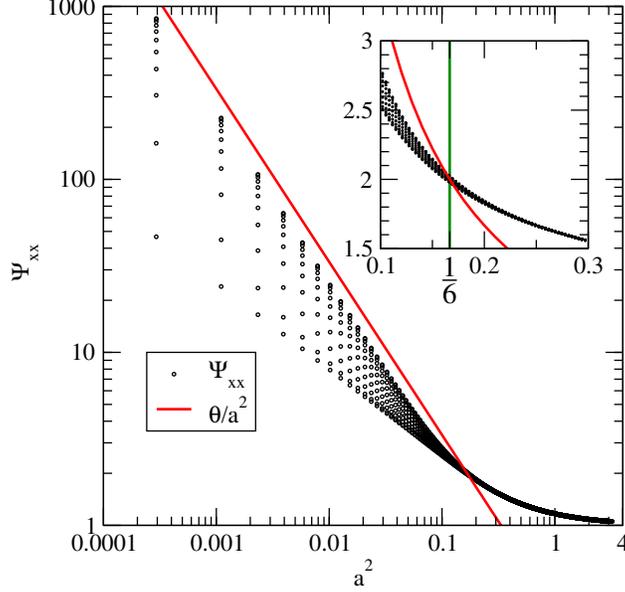


Figure 2.8. The normalized second moment of Eq. (2.59) as a function of the second moment  $a^2$  from Eq. (2.36) of the mean squared displacement probability density. The value of  $\Psi$  is shown for different 11 different values of  $u_x$  between 0 and 0.5. The values for  $u_x = 0$  correspond to the bottom points of the graph. For a Galilean-Invariant discrete equilibrium distribution moment the value of  $\Psi$  should be independent of  $u$ . This is compared to a standard lattice Boltzmann second moment with  $\theta = 1/3$ .

distribution this second moment is shown in Fig. (2.8). This shows that the discrete second moment does only agree with the MD temperature for large  $a^2 \gg 1$ . As we see in Fig. (2.5) this corresponds to a situation where the populated set of velocities encompasses several shells in Fig. (2.3). For lower values of  $a^2$  we find that the second moment diverges. The reason lies in the way we define the discrete equilibrium distribution. Even for very small  $\langle(\delta x)^2\rangle$ , corresponding to very small  $\Delta t$ , we will identify a fraction of particles that happen to cross from one lattice point to the next and are therefore assigned a lattice velocity  $v_i$  of order one. This appearance of apparent large displacements causes the divergence of the discrete second moment. This effect is significantly enhanced by imposed velocities  $u$ .

For  $a^2 < 0.1$  we see that we get significantly diverging values of  $\Psi$  for different velocities  $u$ . This implies that this discrete second moment is not Galilean invariant. It is important to note that despite such violations of Galilean invariance of the discrete moments the full MDLG algorithm does not suffer from a Galilean invariance problem. Instead this is

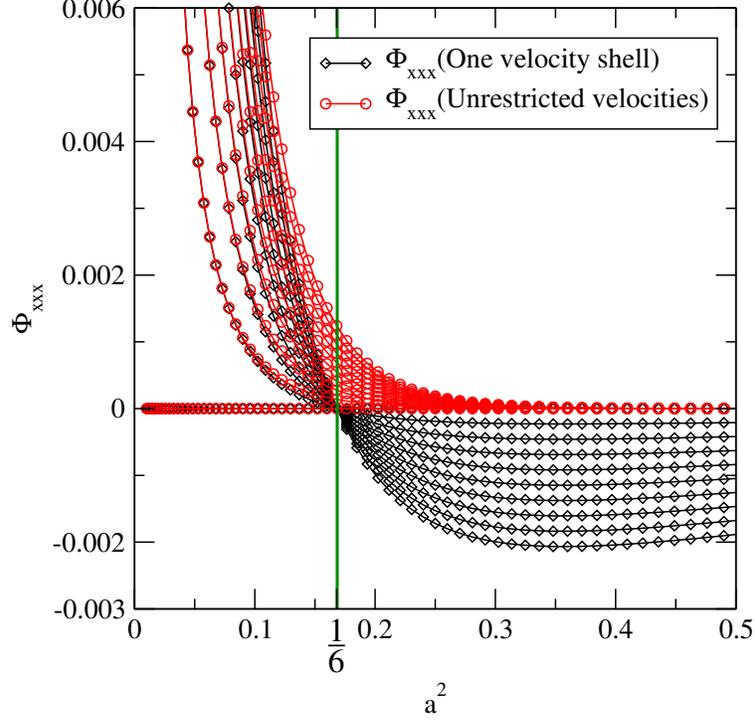


Figure 2.9. Third velocity moment given by Eq. (2.60) for an unrestricted velocity set and a velocity set restricted to only a single shell of velocities for 10 equally spaced velocities  $u_x$  between 0 and 0.001.

an indication that the collision operator Eq. (2.12) must exactly compensate this apparent Galilean invariance violation. For lattice Boltzmann methods one typically tries to avoid Galilean invariance violations by ensuring that both the local equilibrium distributions and the collision operator independently obey Galilean invariance.

For the lattice Boltzmann method we expect this second moment to be  $\Psi_{\alpha\beta} = \theta/a^2\delta_{\alpha\beta}$ . As we saw above [see Eq. (2.51)] lattice Boltzmann methods which have a velocity set consisting of a single shell in velocity space require  $\theta = 1/3$ . This value of  $\theta$  is consistent with the moment of the MDLG equilibrium for a value of  $a^2 \approx 1/6$ . We find  $\theta \approx 2a^2 = \langle(\delta x)^2\rangle/(\Delta x)^2$ . This corresponds to about the lowest value for  $a^2$  where  $\Psi$  does not strongly depend on  $u$  and would therefore violate Galilean invariance.

The condition that  $\theta = 1/3$  came out of a consideration for the third moment for a minimal velocity set with  $v_{ix} = v_{ix}^3$ . We can define a third moment as

$$\Phi_{\alpha\beta\gamma} = \frac{\sum_i f_i^{eq} (v_{i\alpha} - u_\alpha)(v_{i\beta} - u_\beta)(v_{i\gamma} - u_\gamma)}{\rho^{eq} a^3}. \quad (2.60)$$

We examine the behavior of  $\Phi_{xxx}$  in Fig. (2.9). In fact this third moment should be zero, and for sufficiently large  $a$  it converges to zero exponentially. However, if we artificially restrict our velocity set to a single shell, neglecting the small densities for discrete velocities outside the first shell, we find that there is a collapse of  $\Phi_{xxx}$  to zero for the same  $a^2 \approx 1/6$  that we found in Fig. (2.8). For the full velocity set, however, nothing special occurs at  $a^2 = 1/6$ . This initially surprised us because we see in Fig. (2.5) that the densities associated with the second shell are still a factor of 30 smaller than the densities of the first shell and might therefore be taken to be negligible. The second shell velocities that are about a factor 2 larger than the first shell velocities, so these densities get multiplied by a factor of  $2^3$  which allows these densities to contribute enough that we now have

$$\sum_i f_i^{eq} v_{ix} \neq \sum_i f_i^{eq} v_{ix}^3 \quad (2.61)$$

and they are different enough that the cancellation of terms that is supposed to lead to for  $\Phi_{xxx} = 0$  at  $a^2 = 1/6$  no longer exists. Instead  $\Phi_{xxx}$  monotonously approaches zero.

The second and third moments of the distribution functions both relate to the lattice Boltzmann temperature of Eq. (2.49). We can derive two expressions for the temperature

$$\theta_2 = a^2 \Psi_{xx} \quad (2.62)$$

$$\theta_3 = \frac{1}{3\rho^{eq}} \partial_{u_x} \sum_i f_i^{eq}(\rho^{eq}, u)(v_{ix})^3. \quad (2.63)$$

The dependence of these two quantities on the mean-square displacement is shown in Fig. (2.10). We see that these two definitions only agree with each other for large  $a^2$ , where

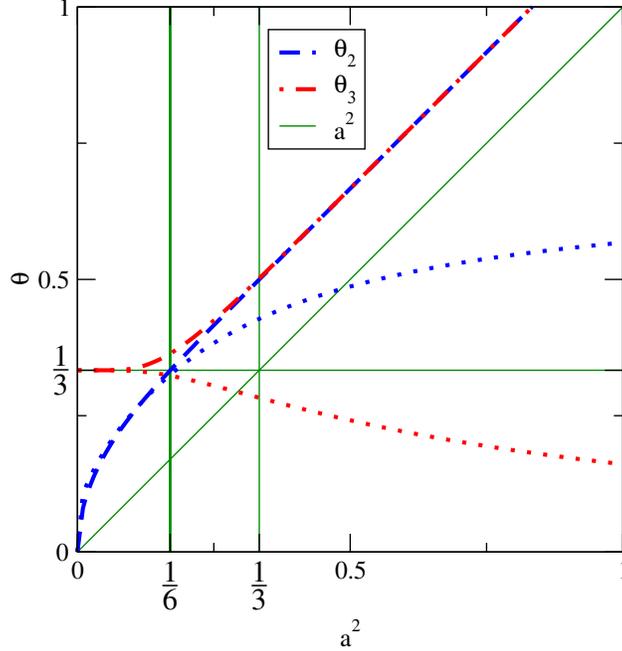


Figure 2.10. Two manifestations of the temperature  $\theta$  from the second and third moments, given by Eqs. (2.62) and (2.63). We show the moments for the full velocity set and moments for a velocity set restricted to the first shell (thinner lines). For the restricted velocity set we only see agreement between the two moments for  $\theta = 1/3$  corresponding to  $a^2 = 1/6$ .

agreement is facilitated by utilizing a large set velocity for the discretization of  $f_i^{eq}$ . For small  $a^2$ , and therefore a minimal velocity set, we get  $\theta_3 = 1/3$ , for the same reason that was discussed before, *i.e.*  $v_{ix} = v_{ix}^3$ .

If we artificially restrict the velocity set to a single shell (*e.g.* 9 velocities in our two-dimensional example) we affect both definitions of the LB temperature. In this case the agreement for large  $a^2$  disappears, and we have exactly one point for  $\theta = 1/3$  corresponding to  $a^2 = 1/6$  for which both expressions for the temperature agree. This corresponds to the special point in Fig. (2.9) where the results for the one-shell velocity set become independent of  $u$  (for small enough  $u$ ).

This suggests that there is a serendipitous agreement between the MDLG equilibrium distribution and the standard LB equilibrium distribution for one-shell velocity sets. It allows us to recover velocity moments up to order 3, which is exactly the order required by kinetic theory to recover the continuity and Navier-Stokes equations. As seen in Figs. (2.8) and (2.9), this value is just large enough to avoid apparent Galilean-invariance violations in the

moments of the MDLG equilibrium distribution, and just small enough so that the values of the  $f_i$  for velocities with  $v_{i\alpha} > 1$  are small enough not to contribute considerably to the moments of the equilibrium distributions. These effects can only be noticed for the third moment, where they conspire to induce agreement between the measures of temperature implied by the second and third moments, as seen in Fig. (2.10).

## 2.8. Consistent Discretizations

Up to now we have seen how the moments of the equilibrium distribution depend on  $a^2$  which is a measure of the mean squared displacement. We saw that the discrete moments of  $f_i^{eq}$  differ from the continuous moments of the Maxwell-Boltzmann distribution even in the ballistic regime. The underlying reason for this disagreements result from two conspiring effects. Firstly we only know the position of our particles to lie somewhere within their assigned lattice cell. This uncertainty enters the definition of the  $f_i^{eq}$  in Eq. (2.23). Secondly we use a discrete second moment. Here we show how both of these give an offset of  $1/12$  in  $a^2$  giving rise to a total shift of  $1/6$  observed in the previous numerical results.

For simplicity let us consider large times away from the ballistic regime. This occurs without loss of generality if our assumption of a Gaussian distribution in Eq. (2.30) remains correct. We can then assume the motion of the particles is entirely diffusive with some diffusion constant  $D$

$$\partial_t \rho(x, t) = D \partial_x^2 \rho(x, t). \quad (2.64)$$

If we had known the position of the particle initially, *i.e.*  $\rho_c(x, 0) = \delta(x)$ , then the particle probability density would evolve as

$$\rho_c(x, t) = \frac{1}{\sqrt{4\pi Dt}} e^{-\frac{x^2}{4Dt}} \quad (2.65)$$

which has a second moment of

$$\frac{1}{\sqrt{4\pi Dt}} \int_{-\infty}^{\infty} x^2 e^{-\frac{x^2}{4Dt}} dx = 2Dt. \quad (2.66)$$

If we only know that at time  $t = 0$  the particle is inside a lattice cell centered around the origin with width  $\Delta x$ , then the density  $\rho$  as a function of time is given by

$$\rho_d(x, t) = \frac{1}{2\Delta x} \left[ \operatorname{erf} \left( \frac{x + \Delta x/2}{2\sqrt{Dt}} \right) - \operatorname{erf} \left( \frac{x - \Delta x/2}{2\sqrt{Dt}} \right) \right] \quad (2.67)$$

which for  $t \rightarrow 0$  gives a density that is  $1/\Delta x$  inside the interval  $[-\Delta x, \Delta x]$  and zero outside it. This probability density distribution then spreads out and approaches a Gaussian at late times. The second moment of the position is then given by

$$\int_{-\infty}^{\infty} x^2 \rho_d(x, t) dx = 2Dt + \frac{(\Delta x)^2}{12} \quad (2.68)$$

*i.e.* a simple offset of  $(\Delta x)^2/12$  that does not depend on time. We can identify

$$a^2 = \frac{2Dt}{(\Delta x)^2} \quad (2.69)$$

from the definition of Eq. (2.36). We then we see that the results are expected to be shifted by  $1/12$ . However, in Fig. (2.10) we clearly see that this only represents half of the observed shift  $1/6$ .

The second effect relates to discretizing the position into displacement bins. We now calculate the discrete second moment (normalized by  $(\Delta x)^2$ ) as

$$m_2 = \sum_{i=-\infty}^{\infty} i^2 \int_{i\Delta x - \frac{\Delta x}{2}}^{i\Delta x + \frac{\Delta x}{2}} \rho_d(x, t) dx. \quad (2.70)$$

In Fig. (2.11) we show that this discrete moment quickly converges to  $a^2 + 1/6$ . We clearly see that the missing additional offset of  $1/12$  that we observed in Fig. (2.10) is the result of taking the discrete moment.

This shows that there are two effects of discretization. Firstly the initially broader distribution of particles confined to a lattice site rather than a point shifts the second moment

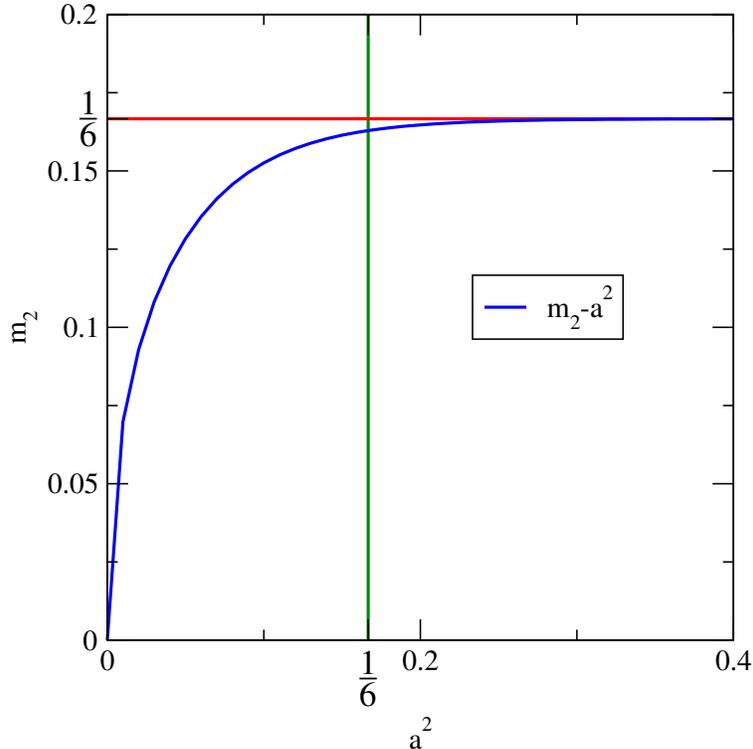


Figure 2.11. The difference of the discrete second moment  $m_2$  of Eq. (2.70) minus  $a^2$  converges quickly to a constant  $1/6$ .

by exactly  $1/12$ . The second effect of discretization is more complicated, particularly for small  $a^2$ . But for  $a^2 > 0.2$  this effect quickly converges to another offset in  $a^2$  of  $1/12$ . Together they make up the offset of  $1/6$ , seen repeatedly in our numerical results of Figs. (2.5), (2.8), (2.9), and (2.10).

## 2.9. Outlook

In this paper we have introduced a new tool for comparing the results of Molecular Dynamics simulations with those of coarse-grained lattice gas or lattice Boltzmann methods. It consists of re-interpreting the MD results as a lattice gas which we call the MDLG. The dynamics of this special kind of lattice gas is entirely given by the MD simulation, and therefore will be able to give a coarse-grained picture for any results that are obtainable with MD simulations.

The approach bears a superficial similarity with the direct simulation Monte Carlo (DSMC) method, a particle-based method where a grid is placed on the lattice and two-

particle collisions are performed between random particles within one cell [80, 81]. This similarity has been remarked upon by both reviewers of this article. It has been shown that DSMC will simulate the Boltzmann equation, and by extension also reproduce the Navier-Stokes equation using kinetic theory arguments similar to those typically used when analyzing the lattice Boltzmann method [82, 83]. This coarse-graining approach to the MD method leads to an approximate approach that is much faster than any hypothetical MD implementation. The MDLG approach, however, is fundamentally different. Since it relies on an underlying MD simulation there are no savings in computational time, and since it continues to track the exact MD result it does not lose accuracy apart from the information lost by projecting onto lattice densities. The key aspect of the MDLG method is that it reproduces an exactly correct lattice gas method which can then be used to compare it to other, efficient, LG or LB methods and validate (or invalidate) their behavior by direct comparison. We believe that this is an important new tool that allows us to analyze an LG or LB method directly rather than go through the usual indirect method of recovering the hydrodynamic equations through some kinetic theory approach.

In this paper, we focus entirely on the averaged equilibrium behavior and show that there exists a close connection between the equilibria of lattice Boltzmann methods and the equilibrium for the MDLG method, when applied to a hot dilute gas. We were able to determine this equilibrium distribution analytically and were able to verify this analytical solution with the results of the MDLG method. Importantly there is a surprisingly good agreement between our equilibrium distribution and the standard lattice Boltzmann result for carefully chosen (and analytically known) pairs of time and space discretizations  $\Delta t$  and  $\Delta x$ . We were able to understand the observed offset of  $1/6$  in the dimensionless measure of the mean squared displacement  $a^2$  in terms of our discretization procedure.

This opens the way for a more careful analysis of the fundamental underpinnings of lattice gas and lattice Boltzmann methods. We intend to utilize our MDLG method to investigate the fundamental properties of the collision operator, including its fluctuating

properties. Further down the line we hope to investigate how the behavior of liquids alters the behavior of the MDLG method and examine if MDLG can also be matched with lattice Boltzmann methods. We anticipate that this method will also be instrumental in putting lattice Boltzmann methods for non-ideal and multi-component systems on a firmer footing.

### **2.10. Acknowledgment**

M.R.P. would like to acknowledge support from NDSU's Doctoral Dissertation Fellowship program.

# 3. ESTABLISHING CORRECT FLUCTUATIONS FOR LATTICE BOLTZMANN METHODS USING THE MOLECULAR DYNAMICS LATTICE GAS METHODS

## 3.1. Abstract

Using the recently introduced Molecular Dynamics Lattice Gas (MDLG) method we investigate the fluctuating properties of lattice Boltzmann methods. In standard fluctuating lattice Boltzmann methods noise terms are derived by assuming that the densities are Poisson distributed. We show that this assumption can be validated for dilute gases and we show how this assumption starts to break down for denser systems.

## 3.2. Introduction

In this chapter we investigate the fundamental fluctuating properties of an equilibrium lattice gas or lattice Boltzmann simulation by relating these method to the recently introduced Molecular Dynamics Lattice Gas (MDLG) approach. The fluctuations in this method are directly related to an underlying Molecular Dynamics simulation and therefore represent the real behavior of particles.

A property of the general LBM is that, unlike the discrete particle behavior of LGA, fluctuations are not inherently included. In LGA methods, the discrete particle behavior gives rise to particle noise which can be thought of as fluctuations. LBM resolves this issue by implementing the distribution functions. The  $f_i$  in LBM can be thought of as non equilibrium 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) [84, 85, 86].

### 3.2.1. The Fluctuating Lattice Boltzmann Equation

The general LBM collision operator can be written as follow (this equation can be extracted from Eq. (1.31))

$$\Omega_i(\{f_j\}) = \sum_j \Lambda_{ij}(f_j^0 - f_j)\Delta t. \quad (3.1)$$

Eq. (3.1) 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  $\xi_i$  which has moments

$$\sum_i \xi_i = 0 \quad (3.2)$$

$$\sum_i v_{i\alpha} \xi_i = 0. \quad (3.3)$$

These noise terms are correlated. The multi-relaxation time (MRT) collision operator is desired since it allows for each degree of freedom to be relaxed by its own independent relaxation time  $\tau_i$ . The fluctuating MRT collision operator takes the form

$$\Omega_i(\{f_j\}) = \sum_j \Lambda_{ij}(f_j^0 - f_j) + \xi_i \quad (3.4)$$

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

$$f_i(\mathbf{x} + \mathbf{v}_i, t + 1) = f_i(\mathbf{x}, t) + \sum_j \Lambda_{ij}(f_j^0 - f_j) + \xi_i. \quad (3.5)$$

With proper forms of  $\Lambda_{ij}$  and  $\xi_i$ , fluctuations will be reintroduced into the LBM.

By transforming into a moment space representation, the noise terms can be decoupled and dealt with individually thus leading to uncorrelated noise terms. This will be discussed in the following section.

### 3.2.2. The Multi Relaxation Time Collision Operator and Moment Space Representation

The multi-relaxation time (MRT) collision operator in Eq. (3.1) 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  $\tau_i$ , 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, they distribution functions are transformed back into velocity space where streaming can take place [87, 88, 89, 90].

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

$$M^a = \sum_i m_i^a f_i \quad (3.6)$$

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

$$f_i = \sum_a n_i^a M^a. \quad (3.7)$$

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

$$\sum_i n_i^a m_i^b = \delta^{ab} \quad (3.8)$$

$$\sum_a m_i^a n_j^a = \delta_{ij}. \quad (3.9)$$

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

$$M^0 = \rho = \sum_i \mathbf{1} f_i \quad (3.10)$$

which is equivalent to saying that the first, second and third row of the transformation matrix is

$$m_i^0 = (1, \dots, 1) \quad (3.11)$$

$$m_i^1 = V_{ix} \quad (3.12)$$

$$m_i^2 = V_{iy} \quad (3.13)$$

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

$$\Lambda_{ij} = \frac{1}{\tau} \delta_{ij}. \quad (3.14)$$

In moment space, the single relation time collision matrix is written

$$\Lambda^{ab} = \sum_{ij} m_i^a \Lambda_{ij} n_j^b = \frac{1}{\tau} m_i^a \delta_{ij} n_j^b = \frac{1}{\tau} \delta^{ab}. \quad (3.15)$$

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

$$\Lambda^{ab} = \frac{1}{\tau^a} \delta^{ab} \quad (3.16)$$

where  $\tau^a$  is the relaxation time which corresponds to moment  $a$ .

In the moment space representation, the collision operator in Eq. (3.4) will become

$$\Omega^a(\{M^a\}) = \frac{1}{\tau^a} (M^{a,0} - M^a) + \xi^a. \quad (3.17)$$

This moment space collision operator will now decouple the noise and still satisfies the conservation of mass and momentum [91, 92, 93, 94, 95].

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

$$\langle \xi^a \xi^b \rangle = \rho \frac{2\tau^a - 1}{(\tau^a)^2} (1 - \delta^{a0} \delta^{b0} - \delta^{a1} \delta^{b1} - \delta^{a2} \delta^{b2}) \delta^{ab} \quad (3.18)$$

which is a fully diagonal matrix representing uncorrelated noise terms [96, 97, 98, 99, 100].

The basic conjecture of the fluctuating lattice Boltzmann method is to assume that the densities have the same second moment as uncorrelated Poisson distributed noise. In particular this implies

$$\langle f_i f_j \rangle = f_i^{eq} f_j^{eq} + f_i^{eq} \delta_{ij} \quad (3.19)$$

which is, at least to good approximation, fulfilled by modern fluctuating lattice Boltzmann methods. To validate this fundamental assumption we now utilize our MDLG approach by

measuring

$$E_{ij} = \frac{\langle n_i n_j \rangle - f_i^{eq} f_j^{eq}}{\sqrt{f_i^{eq} f_j^{eq}}} \quad (3.20)$$

and we observe the deviation (if any) from a  $\delta_{ij}$  function. Deviations will indicate situations where the fluctuations of a real system deviate from those of the FLBM, and also of those expected for an ideal gas.

In Fig. (3.1)(a) we see that for small enough densities we recover the expected ideal gas behavior, but that the fluctuations develop a dependence on the timestep  $\Delta t$  for higher densities Fig. (3.1)(b). Interestingly, for both small and large timesteps the ideal gas behavior is recovered, but for intermediate timesteps we find noticeable correlations.

The first idea one might have as to what causes these correlations might be to assume that the velocities of the molecules at short distances become correlated. We tested this assumption by measuring the velocity correlation as a function of distance. In particular we measured

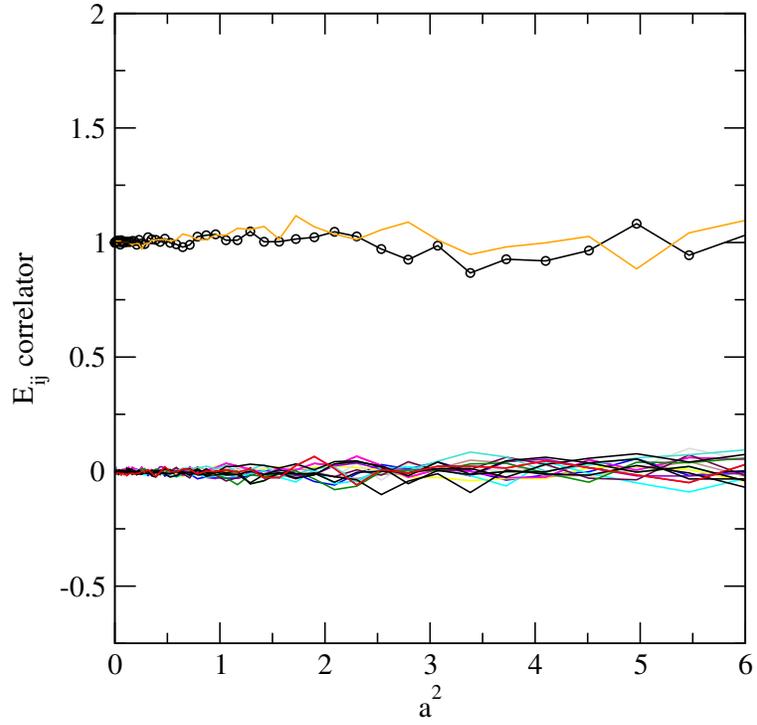
$$\langle v_i(t) v_j(t) \rangle (\Delta X) \quad (3.21)$$

by binning the velocity products for different particle displacements  $\Delta X = |x_i(t) - x_j(t)|$ . However, as shown in Fig. (3.2), no such correlation exists for the velocities.

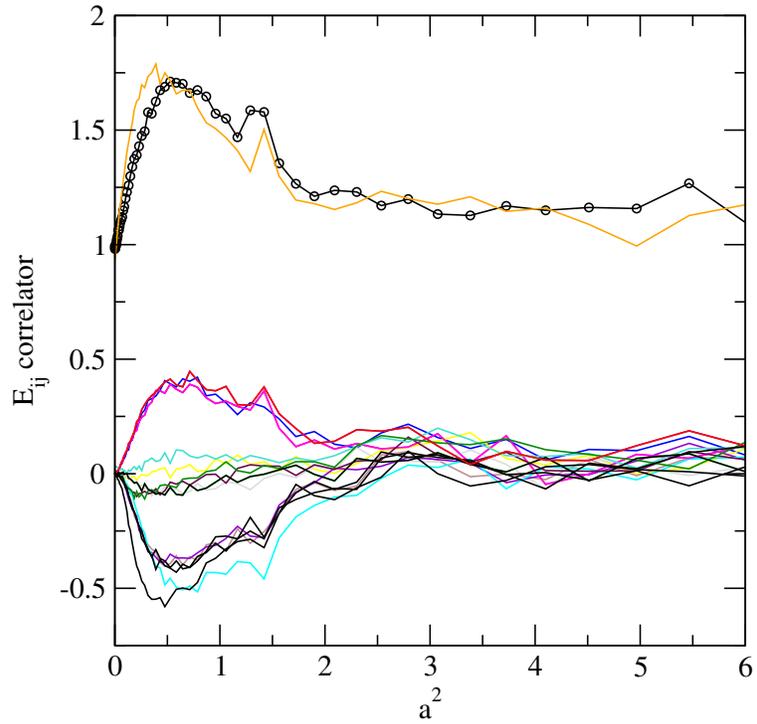
This leads us to examine the origin of these correlations from a fundamental perspective. In the next section we briefly recap the relevant properties of the MDLG method, and then we go on to examine the fundamental properties of the  $\langle n_i(x, t) n_j(x, t) \rangle$ .

### 3.3. Lattice Gas, Lattice Boltzmann and Molecular Dynamics Lattice Gas

Lattice gas methods consist of discrete particles associated with discrete lattice sites. In one time-step these particles move from one lattice site  $x$  to another lattice site  $x + v_i$  where the lattice displacement  $v_i$  is typically referred to as a lattice velocity. These lattice gas methods typically only allow a restricted velocity set consisting of displacements that lead to bordering lattice sites. The particles moving from lattice point  $x$  to lattice point

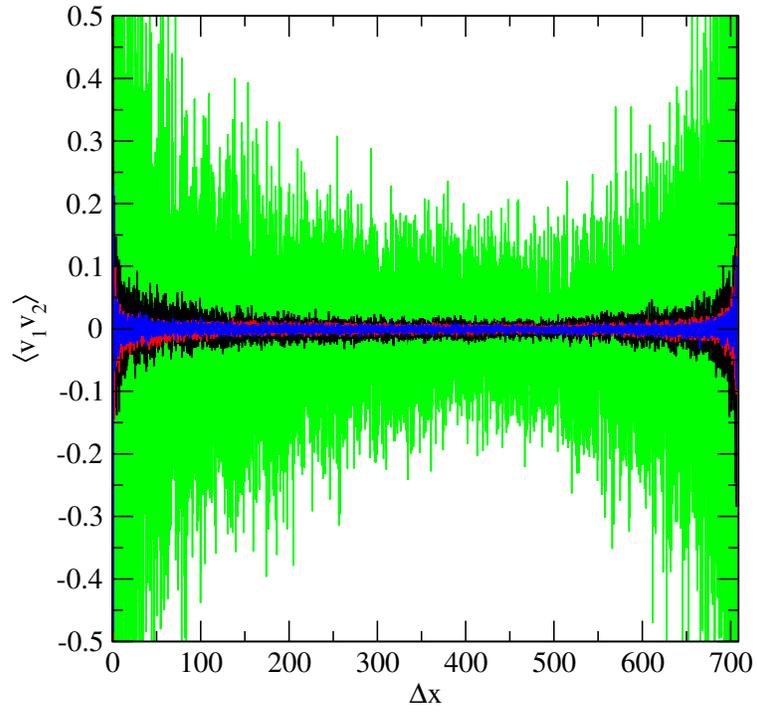


(a)

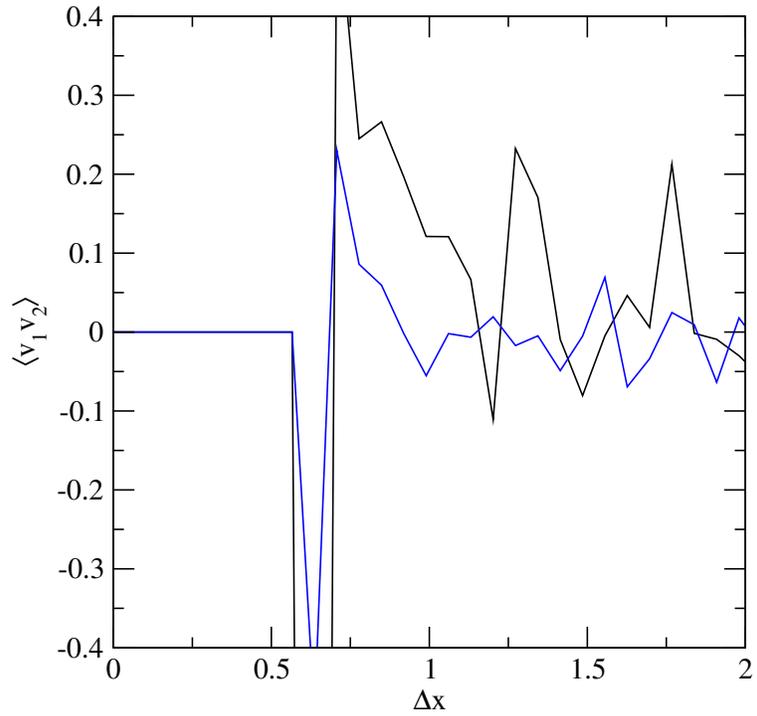


(b)

Figure 3.1. Measured correlations  $E_{ij}$  for different densities (a) Low density ( $\phi = 0.01$ ) (b) Medium density ( $\phi = 0.08$ ). These data obtained for 99 856 particles with box size of 1 000 and 8 000 and  $25 \times 25$  lattice.



(a)



(b)

Figure 3.2. (a) Measured  $\langle v_i v_j \rangle$  for different mean densities: low density (green), medium density (black), medium density for longer simulation (blue), high density (red). (b) Focused on medium density.

$x + v_i$  between time  $t$  and  $t + 1$  are denoted by  $n_i(x, t)$ . After moving to a new lattice site the particles at this lattice site are re-distributed in a collision step that respects mass and momentum conservation. This means that the local mass and momentum are defined as

$$N(x, t) = \sum_i n_i(x, t) \quad (3.22)$$

$$N(x, t)U_\alpha(x, t) = \sum_i n_i(x, t)v_{i\alpha} \quad (3.23)$$

where Greek indices are used to represent the spatial dimensions in standard tensor notation. Traditional lattice gas methods have only one speed, which means that mass conservation and energy conservation are synonymous. A recently introduced lattice gas method with a Monte Carlo collision operator recovers an isothermal condition for velocities  $|U_\alpha| < 0.2$  [101], which is the approach typically employed for lattice Boltzmann methods. In any case the lattice gas methods can then be written in the form of the evolution equation

$$n_i(x + v_i, t + 1) = n_i(x, t) + \Xi_i \quad (3.24)$$

where the collision operator consists of a set of collision rules with (typically) probabilistic outcomes. These collisions both relax the states towards an equilibrium state and also add fluctuations. Therefore LG methods are fluctuating methods which recover many of the fluctuating properties of an ideal gas.

We earlier introduced a coarse-graining of Molecular Dynamics (MD) simulations that allows us to relate the MD simulations to lattice gas (LG) and lattice Boltzmann (LB) methods [7]. The basic idea is that in a LG (or LB) simulation particles move from one lattice site to another lattice site and the number (or density) is associated by their lattice displacement  $v_i$  and denoted as  $n_i$  (or  $f_i$ ).

### 3.4. Molecular Dynamics Lattice Gas

Correlation between two particles in MDLG system follows the same principles of one particle distribution. Finding the proper function and probability density for correlation between two particles are the main change in this part. For better understanding of this correlation first we review one particle distribution function. Given a set of trajectories for  $N$  particles  $x_l(t)$ , the lattice gas occupation numbers was defined in Eq. (2.11) as

$$n_i(x, t) = \sum_{l=1}^N \int d\mathbf{x} \Delta_{\mathbf{x}}[\mathbf{x}_l(t)] \Delta_{\mathbf{x}-\mathbf{v}_i}[\mathbf{x}_l(t - \Delta t)] \quad (3.25)$$

where we defined

$$\Delta_{\mathbf{x}}(\mathbf{y}) = \prod_{\alpha} \Delta_{x_{\alpha}}(y_{\alpha}) \quad (3.26)$$

and the one-dimensional  $\Delta$  function was previously defined in Eq. (2.10) as

$$\Delta_x(y) = \begin{cases} 1 & \text{if } x < y \leq x + \Delta x \\ 0 & \text{otherwise.} \end{cases} \quad (3.27)$$

For a system at equilibrium we will have a translationally invariant state. If  $P(x, \delta x)$  is the probability density that a particle moves by a displacement  $\delta \mathbf{x}$  in a time  $\Delta t$  (this probability does not depend on initial position  $x$  since we assumed particles are homogeneously distributed) then we can write the expectation value as

$$\begin{aligned} f_i^{eq} &= \langle n_i(x, t) \rangle \\ &= \frac{\rho^{eq}}{(\Delta x)^D} \int dx' \int d(\delta x') \Delta_x[x'] \Delta_{x-v_i}[x' + \delta x'] P(\delta x') \\ &= \rho^{eq} \int d(\delta x') W(v_i - \delta x') P(\delta x') \end{aligned} \quad (3.28)$$

where the wedge function  $W$  was defined in Eq. (2.24) as

$$\begin{aligned}
W(x) &= \int_{-\infty}^{\infty} dy \Delta_0(y) \Delta_0(y+x) \\
&= \int_{\max(0,x)}^{\min(\Delta x, \Delta x+x)} 1 dy \\
&= \begin{cases} \min(\Delta x, \Delta x+x) - \max(0, x) & \text{if positive} \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} \Delta x - |x| & \text{if } x < \Delta x \\ 0 & \text{otherwise.} \end{cases} \tag{3.29}
\end{aligned}$$

### 3.5. Two Particle Correlations

Correlation between two particles can be divided in two different parts: first, correlation between two different particles and second, correlation of particle and itself. The following equation shows these parts in two different integrals

$$\begin{aligned}
\langle n_i n_j \rangle &= \sum_k \sum_l \langle \Delta_x[x_k(t)] \Delta_{x-v_i}[x_k(t-\Delta t)] \\
&\quad \Delta_x[x_l(t)] \Delta_{x-v_j}[x_l(t-\Delta t)] \rangle \\
&= \frac{\rho^{eq}}{(\Delta x)^d} \int dx_1 \int d(\delta x_1) \Delta_x[x_1] \Delta_{x-v_i}[x_1 + \delta x_1] P(x_1, \delta x_1) \delta_{ij} \\
&\quad + \frac{\rho^{eq} \rho^{eq}}{(\Delta x)^{2d}} \int dx_1 \int d(\delta x_1) \int dx_2 \int d(\delta x_2) \\
&\quad \Delta_x[x_1] \Delta_{x-v_i}[x_1 + \delta x_1] \Delta_x[x_2] \Delta_{x-v_j}[x_2 + \delta x_2] \\
&\quad P(x_1, \delta x_1, x_2, \delta x_2). \tag{3.30}
\end{aligned}$$

Since in ideal gas we can assume particles are independent then we can write

$$P^{id}(x_1, \delta x_1, x_2, \delta x_2) = P(\delta x_1) P(\delta x_2) \tag{3.31}$$

and also in ideal gas it would be expected that these fluctuations are independent so by replacing Eq. (3.31) in Eq. (3.30), two integrals are independent and are equal to  $f_i^{eq} f_j^{eq}$ , so that we would find

$$\langle n_i n_j \rangle = f_i^{eq} f_j^{eq} + f_i^{eq} \delta_{ij} \quad (3.32)$$

where we have analytical expressions for the global equilibrium distribution  $f_i^{eq}$ .

To understand this behavior, the key is now to determine the two-particle probability. Here we hope that we can find a set of variables that both explains the observed correlations and at the same time factorizes. Our ansatz is

$$P(x_1, \delta x_1, x_2, \delta x_2) = P^+(\delta x^+; \Delta X) P^-(\delta x^-; \Delta X) P^\Delta(\Delta X) \quad (3.33)$$

where

$$2\delta x^+ = \delta x_1 + \delta x_2 \quad (3.34)$$

$$2\delta x^- = \delta x_1 - \delta x_2. \quad (3.35)$$

In theory the probability for correlation of two particles after timestep  $\Delta t$  depends on initial position of particles ( $x_1$  and  $x_2$ ) and the distance for each particle during timestep  $\Delta t$ . For simplicity we assume the initial position of particles can be replaced by initial distance of two particles ( $\Delta X$ ) since particles distributed homogeneously. Also because of correlation between  $\delta x_1$  and  $\delta x_2$  we cannot factorize the probability based on these two variables. Two new variables  $\delta x^+$  and  $\delta x^-$  can represent the correlation and factorization property at the same time.

We make the ansatz that

$$P^+(\delta x^+) = \frac{1}{(2\pi\sigma^+)^{D/2}} \exp\left(-\frac{(\delta x^+)^2}{2(\sigma^+)^2}\right) \quad (3.36)$$

$$P^-(\delta x^-) = \frac{1}{(2\pi\sigma^-)^{D/2}} \exp\left(-\frac{(\delta x^-)^2}{2(\sigma^-)^2}\right) \quad (3.37)$$

$$P^\Delta(\Delta x) = \frac{N-1}{V-\pi r^2} \Theta(\Delta x - r) \quad (3.38)$$

where  $r$  is the effective radius of our Lennard-Jones particles and there is a  $\Delta X$  dependence in the variances  $\sigma^+(|\Delta X|)$  and  $\sigma^- (|\Delta X|)$ . We know that for large distances the particles have to be uncorrelated, so

$$\lim_{\Delta X \rightarrow \infty} \sigma^+(|\Delta X|) = \langle \delta x^2 \rangle \quad (3.39)$$

$$\lim_{\Delta X \rightarrow \infty} \sigma^-(|\Delta X|) = \langle \delta x^2 \rangle. \quad (3.40)$$

To obtain  $\sigma^+(|\Delta X|)$  and  $\sigma^-(|\Delta X|)$  we examine the correlations of two displacements

as

$$\begin{aligned} & \langle \delta x_1 \cdot \delta x_2 \rangle (\Delta x) \\ &= \int P(\delta x_1, \delta x_2; \Delta x) \delta x_1 \cdot \delta x_2 d\delta x_1 d\delta x_2 \\ &= \int P(\delta x^+; \Delta X) P(\delta x^-; \Delta X) \cdot (\delta x^+ \cdot \delta x^+ - \delta x^- \cdot \delta x^-) d\delta x^+ d\delta x^- \\ &= \sigma^+ - \sigma^-. \end{aligned} \quad (3.41)$$

We also have the condition that integrating over the second particle coordinate we recover

the one-particle distribution function

$$\begin{aligned}
P(\delta x_1) &= \int P(\delta x_1, \delta x_2; \Delta X) d\delta x_2 \\
&= \int \exp\left(-\frac{(\delta x_1 + \delta x_2)^2}{2(\sigma^+)^2}\right) \exp\left(-\frac{(\delta x_1 - \delta x_2)^2}{2(\sigma^-)^2}\right) d\delta x_1 \\
&= \exp\left(-\frac{2(\delta x_1)^2}{\sigma^+ + \sigma^-}\right)
\end{aligned} \tag{3.42}$$

from which we can see that

$$\sigma^+ + \sigma^- = 2\langle \delta x^2 \rangle. \tag{3.43}$$

Now we can measure this correlator directly and we obtain for  $\sigma^+$  and  $\sigma^-$

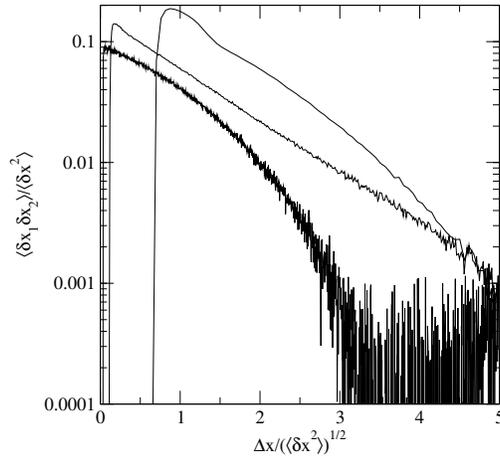
$$\sigma^+ = \langle \delta x^2 \rangle + \langle \delta x_1 \cdot \delta x_2 \rangle \tag{3.44}$$

$$\sigma^- = \langle \delta x^2 \rangle - \langle \delta x_1 \cdot \delta x_2 \rangle. \tag{3.45}$$

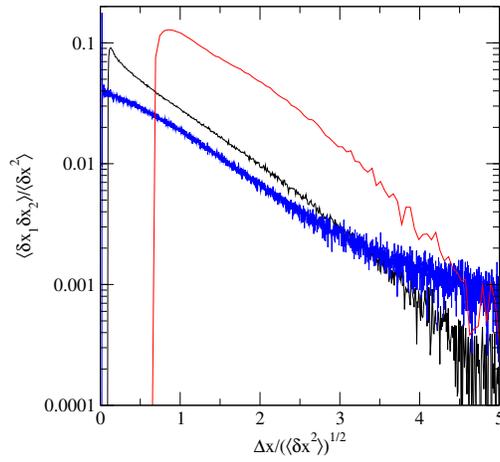
Measurements of these correlators are shown in Fig. (3.3). With these measurements we can predict the two free parameters  $\sigma^+$  and  $\sigma^-$  as a function of  $\Delta X$ .

In the other hand we should be able to capture the correlation between two particles by measuring the value of  $\langle v_i v_j \rangle$  at the same timestep. Fig. (3.2) shows  $\langle v_i v_j \rangle$  for three different densities. Blue data represent the same density as black data just for longer simulation time. This graph suggests the value of  $\langle v_i v_j \rangle$  does not depend on correlation of two particles. In theory for very short timestep ( $\tau = 0.2$ ),  $\langle v_i v_j \rangle = \langle \delta x_1 \delta x_2 \rangle$ .

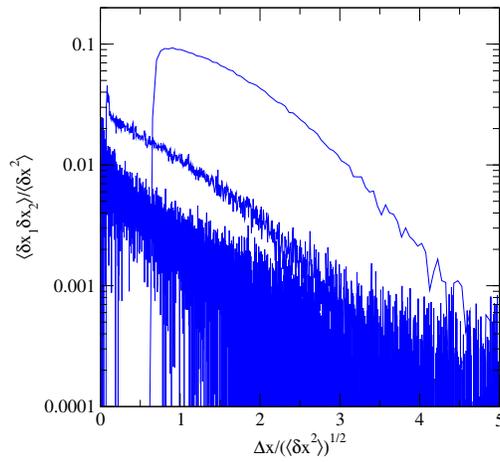
Now we examine what our ansatz (Eq. (3.33)) predicts for the correlators for the occupation numbers. The system is still translationally invariant, so the probability will only depend on the distance between the two particles. We can pick two new variables  $X = (x_1 + x_2)/2$  and  $\Delta X = (x_2 - x_1)/2$ . With this change in variables, and noting that  $P$



(a)



(b)



(c)

Figure 3.3. Measured correlators  $\langle \delta x_1 \delta x_2 \rangle$  for different timesteps  $\Delta t$  and different mean densities (a) high density (b) medium density (c) low density.

has no dependence on  $X$ , we can now write

$$\begin{aligned}
\langle n_i(x)n_j(x') \rangle &= \int dX \int d\Delta X \int d(\delta x^+) \int d(\delta x^-) \\
&\quad \Delta_x[X + \Delta X] \Delta_{x-v_i}[X + \Delta X + \delta x^+ + \delta x^-] \\
&\quad \Delta'_x[X - \Delta X] \Delta_{x'-v_j}[X - \Delta X + \delta x^+ - \delta x^-] \\
&\quad P^+(\delta x^+) P^-(\delta x^-) P^\Delta(\Delta X) \\
&= \int d\Delta X \int d(\delta x^+) \int d(\delta x^-) \\
&\quad Z(\delta x^+ - v_i - v_j, \delta x^- - v_j + v_i, \Delta x + x - x') \\
&\quad P^+(\delta x^+) P^-(\delta x^-) P^\Delta(\Delta X)
\end{aligned} \tag{3.46}$$

where we introduced the  $Z$  function which is a generalization of the wedge function  $W$  defined through

$$\begin{aligned}
&Z(\delta x^+, \delta x^-, \Delta X) \\
&= \int_{-\infty}^{\infty} dX \Delta_0[X] \Delta_0[X + \delta x^+ + \delta x^-] \\
&\quad \Delta_0[X - 2\Delta X] \Delta_0[X - 2\Delta X + \delta x^+ - \delta x^-] \\
&= \int_{\max(0, \delta x^+ + \delta x^-, -2\Delta X, -2\Delta X + \delta x^+ - \delta x^-)}^{\min(\Delta x, \Delta x + \delta x^+ + \delta x^-, \Delta x - 2\Delta X, \Delta x + 2\Delta X + \delta x^+ - \delta x^-)} 1 dX \\
&= \min(\Delta x, \Delta x + \delta x^+ + \delta x^-, \Delta x - 2\Delta X, \Delta x + 2\Delta X + \delta x^+ - \delta x^-) \\
&\quad - \max(0, \delta x^+ + \delta x^-, -2\Delta X, -2\Delta X + \delta x^+ - \delta x^-) \text{ if } > 0.
\end{aligned} \tag{3.47}$$

Based on Fig. (3.3), we can assume correlators  $\langle \delta x_1 \delta x_2 \rangle$  approximately follow an exponential function for large distances. Then by plotting  $\sigma^+$  and  $\sigma^-$ , we are able to predict the joint probability. After measuring  $Z$  function for the given value, we could predict two particles

fluctuation by the following equation

$$\begin{aligned}
\langle n_i(x)n_j(x') \rangle &= \int d(\delta x_1) \int d\Delta X \int d(\delta x_2) \\
&\quad Z(\delta x_1 - v_i, \delta x_2 - v_j, \Delta x + x - x') \\
&\quad P(\Delta x, \delta x_1, \delta x_2) \\
&= \int d(\delta x_1) \int d\Delta X \int d(\delta x_2) \\
&\quad Z(\delta x_1 - v_i, \delta x_2 - v_j, \Delta x + x - x') \\
&\quad P(\delta x_1)P(\delta x_2) \left[ 1 + \delta x_1 \cdot \delta x_2 A \exp\left(-\frac{\Delta x}{\xi}\right) \right] \\
&= f_i^{eq} f_j^{eq} + \int d(\delta x_1) \int d(\delta x_2) \int d\Delta x \\
&\quad Z(v_i + \delta x_1, v_j + \delta x_2, \Delta x) P(\delta x_1) P(\delta x_2) \\
&\quad \delta x_1 \cdot \delta x_2 A \exp\left(-\frac{\Delta X}{\xi}\right). \tag{3.48}
\end{aligned}$$

However, there is an additional difficulty since the  $\langle \delta x_1 \delta x_2 \rangle$  correlator has a strong negative contribution for distances smaller than the Lenard-Jones interaction radius (1 in the units used here). We believe that a full model of this correlation function would allow us to predict the observed correlations of the  $n_i$ . Unfortunately we were unable to complete this calculation in time.

## 4. FURTHER RESEARCH

The motivation for this work was to gain a better understanding of the lattice Boltzmann and lattice gas methods through an explicit coarse-graining procedure that relates lattice Boltzmann densities to Molecular Dynamics particles moving between coarse-graining boxes.

The coarse-graining procedure gave us a lattice gas method with a Molecular Dynamics collision operator (MDLG). This is in principle an exact representation for the underlying physics that is rendered by the Molecular Dynamics simulation. Therefore we can take this new MDLG method as a standard to evaluate other lattice gas and lattice Boltzmann methods.

The main contribution of this research has been to propose and analyze, for the first time, an averaged equilibrium behavior and were able to relate the lattice Boltzmann equilibrium distribution to the behavior of the Molecular Dynamics simulation.

### 4.1. Examination of the Fluctuating Behavior of the MDLG Model

Our initial analyses have assumed no correlation between particles that led us to the expected Poisson distribution behavior for dilute gases. Our later results indicated this assumption is not true for denser gases. Strong correlation during the simulation process was observed for higher density that proved the need for a reformulation of the probability distribution function of the particles. This important piece of information could lead us to true understanding of lattice Boltzmann fluctuation. It should be noticed that previous models neglected any correlation between particles due to the Boltzmann molecular chaos assumption. Based on direction, particles show positive and negative correlation for higher density. Positive correlation for same direction and negative correlation for opposite direction indicate the importance of local equilibrium for higher densities. Our results suggest, strong correlation between a group of particles create a local equilibrium that keeps these particles together.

On the other hands, there is much discussion about the correct way of introducing fluctuations into non-ideal systems. Non of the existing methods appears entirely satisfactory. The MDLG method could provide a useful benchmark to evaluate the correct fluctuations for such non-ideal systems. These systems are of particular interest, since all nucleation phenomena rely on the interplay of a non-ideal phase-separating system with fluctuations.

#### **4.2. Formulation of the Collision Operator of the MDLG Model**

During the simulation, particles are moving after each time-step and the new distribution represents the connection between the current and previous position of particles. This looks like the collision process in a lattice Boltzmann method. Ideally this new proposed collision operator could be fomulised like as lattice gas. The main goal of this part would to relate the transport coefficients of a liquid to this collision process and to evaluate how good an approximation the current LB collision operators are to this MDLG collision operator.

#### **4.3. Multi-Phase Behavior**

The application of lattice Boltzmann methods to non-ideal gases, and in particular phase-separating systems is of intense interest. However, there is much debate about the correct implementation of non-ideal terms. Several methods have been proposed from altering the equilibrium distribution to include a non-ideal Pressure tensor, over the introduction of pseudo-potentials who's gradient terms result in a forcing term derived from gradients of a chemical potential. The MDLG method could allow us to extract the correct method for including the non-ideal parts by examining which, if any, of the existing methods is consistent with the behavior of the MDLG method for a phase-separating Lennard-Jones system.

## REFERENCES

- [1] Stephen Wolfram. High-speed computing: Scientific applications and algorithm design. chapter Cellular automation supercomputing, pages 40–48. University of Illinois Press, Champaign, IL, USA, 1988.
- [2] Xiaoyi He and Gary Doolen. Lattice boltzmann method on curvilinear coordinates system: flow around a circular cylinder. *Journal of Computational Physics*, 134(2):306–315, 1997.
- [3] Jie Bao and Laura Schaefer. Lattice boltzmann equation model for multi-component multi-phase flow with high density ratios. *Applied Mathematical Modelling*, 37(4):1860–1871, 2013.
- [4] Jianhui Yang and Edo S Boek. A comparison study of multi-component lattice boltzmann models for flow in porous media applications. *Computers & Mathematics with Applications*, 65(6):882–890, 2013.
- [5] MR Kamali, S Sundaresan, HEA Van den Akker, and JJJ Gillissen. A multi-component two-phase lattice boltzmann method applied to a 1-d fischer–tropsch reactor. *Chemical engineering journal*, 207:587–595, 2012.
- [6] Xiaowen Shan and Gary Doolen. Multicomponent lattice-boltzmann model with interparticle interaction. *Journal of Statistical Physics*, 81(1):379–393, 1995.
- [7] M Reza Parsa and Alexander J Wagner. Lattice gas with molecular dynamics collision operator. *Physical Review E*, 96(1):013314, 2017.
- [8] James E Broadwell. Study of rarefied shear flow by the discrete velocity method. *Journal of Fluid Mechanics*, 19(3):401–414, 1964.

- [9] J Hardy, Yv Pomeau, and O De Pazzis. Time evolution of a two-dimensional classical lattice system. *Physical Review Letters*, 31(5):276, 1973.
- [10] Uriel Frisch, Brosl Hasslacher, and Yves Pomeau. Lattice-gas automata for the navier-stokes equation. *Physical review letters*, 56(14):1505, 1986.
- [11] Stephen Wolfram. Cellular automaton fluids 1: Basic theory. *Journal of statistical physics*, 45(3):471–526, 1986.
- [12] Dominique d’Humieres, Pierre Lallemand, and Uriel Frisch. Lattice gas models for 3d hydrodynamics. *EPL (Europhysics Letters)*, 2(4):291, 1986.
- [13] Guy R McNamara and Gianluigi Zanetti. Use of the boltzmann equation to simulate lattice-gas automata. *Physical review letters*, 61(20):2332, 1988.
- [14] Xiaowen Shan and Hudong Chen. Lattice boltzmann model for simulating flows with multiple phases and components. *Physical Review E*, 47(3):1815, 1993.
- [15] Manjusha Namburi, Siddharth Krithivasan, and Santosh Ansumali. Crystallographic lattice boltzmann method. *Scientific reports*, 6:27172, 2016.
- [16] Muhammad Izham, Tomohiro Fukui, and Koji Morinishi. Application of regularized lattice boltzmann method for incompressible flow simulation at high reynolds number and flow with curved boundary. *Journal of Fluid Science and Technology*, 6(6):812–822, 2011.
- [17] R Benzi, MV Struglia, and R Tripicciono. Extended self-similarity in numerical simulations of three-dimensional anisotropic turbulence. *Physical Review E*, 53(6):R5565, 1996.
- [18] G Amati, R Benzi, and S Succi. Extended self-similarity in boundary layer turbulence. *Physical Review E*, 55(6):6985, 1997.

- [19] Xiaoyi He, Xiaowen Shan, and Gary D Doolen. Discrete boltzmann equation model for nonideal gases. *Physical Review E*, 57(1):R13, 1998.
- [20] SH Kim and H Pitsch. On the lattice boltzmann method for multiphase flows. *Annual Research Briefs, Center for Turbulence Research*, 2009.
- [21] D Arumuga Perumal and Anoop K Dass. A review on the development of lattice boltzmann computation of macro fluid flows and heat transfer. *Alexandria Engineering Journal*, 54(4):955–971, 2015.
- [22] HNV Temperley. Statistics of a two-dimensional lattice gas. i. *Proceedings of the Physical Society*, 80(4):813, 1962.
- [23] J Hardy, O De Pazzis, and Yves Pomeau. Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. *Physical review A*, 13(5):1949, 1976.
- [24] Bruce M Boghosian. Exact hydrodynamics of the lattice bgk equation. *arXiv preprint arXiv:0810.2344*, 2008.
- [25] Bastien Chopard, Alexandre Dupuis, Alexandre Masselot, and Pascal Luthi. Cellular automata and lattice boltzmann techniques: An approach to model and simulate complex systems. *Advances in complex systems*, 5(02n03):103–246, 2002.
- [26] Bruce M Boghosian. Lattice gases and cellular automata. *Future Generation Computer Systems*, 16(2):171–185, 1999.
- [27] Daniel H Rothman and Stiphane Zaleski. *Lattice-gas cellular automata: simple models of complex hydrodynamics*, volume 5. Cambridge University Press, 1997.
- [28] Christopher Adler, Bruce Boghosian, Eirik G Flekkøy, Norman Margolus, and Daniel H Rothman. Simulating three-dimensional hydrodynamics on a cellular automata machine. *Journal of Statistical Physics*, 81(1):105–128, 1995.

- [29] James E Broadwell. Shock structure in a simple discrete velocity gas. *The Physics of Fluids*, 7(8):1243–1247, 1964.
- [30] Stephen Wolfram. *Cellular automata and complexity: collected papers*, volume 1. Addison-Wesley Reading, MA, 1994.
- [31] P Douglas Tougaw and Craig S Lent. Dynamic behavior of quantum cellular automata. *Journal of Applied Physics*, 80(8):4722–4736, 1996.
- [32] Tommaso Toffoli and Norman Margolus. *Cellular automata machines: a new environment for modeling*. MIT press, 1987.
- [33] Sauro Succi. Numerical solution of the schrödinger equation using discrete kinetic theory. *Physical Review E*, 53(2):1969, 1996.
- [34] Daniel H Rothman and Stéphane Zaleski. Lattice-gas models of phase separation: interfaces, phase transitions, and multiphase flow. *Reviews of Modern Physics*, 66(4):1417, 1994.
- [35] Nicos S Martys and Hudong Chen. Simulation of multicomponent fluids in complex three-dimensional geometries by the lattice boltzmann method. *Physical review E*, 53(1):743, 1996.
- [36] Alexei D Kotelnikov and David C Montgomery. A kinetic method for computing inhomogeneous fluid behavior. *Journal of Computational Physics*, 134(2):364–388, 1997.
- [37] K Komater, G Zandler, and P Vogl. Lattice-gas cellular-automaton method for semi-classical transport in semiconductors. *Physical Review B*, 46(3):1382, 1992.
- [38] Shuling Hou, Qisu Zou, Shiyi Chen, Gary Doolen, and Allen C Cogley. Simulation of cavity flow by the lattice boltzmann method. *Journal of computational physics*, 118(2):329–347, 1995.

- [39] Xiaoyi He and Li-Shi Luo. A priori derivation of the lattice boltzmann equation. *Physical Review E*, 55(6):R6333, 1997.
- [40] Xiaoyi He and Li-Shi Luo. Theory of the lattice boltzmann method: From the boltzmann equation to the lattice boltzmann equation. *Physical Review E*, 56(6):6811, 1997.
- [41] J-P Rivet and Jean Pierre Boon. *Lattice gas hydrodynamics*, volume 11. Cambridge University Press, 2005.
- [42] Jean Pierre Boon and Sidney Yip. *Molecular hydrodynamics*. Courier Corporation, 1991.
- [43] Gary D Doolen. *Lattice gas methods for partial differential equations: a volume of lattice gas reprints and articles, including selected papers from the workshop on large nonlinear systems, held August, 1987 in Los Alamos, New Mexico*, volume 4. Addison-Wesley Longman, 1990.
- [44] Chopard Bastien and Droz Michel. *Cellular automata modeling of physical systems*. Cambridge University Press, 1998.
- [45] Gary D Doolen. *Lattice gas methods: theory, applications, and hardware*. MIT press, 1991.
- [46] Mark E Tuckerman and Glenn J Martyna. *Understanding modern molecular dynamics: techniques and applications*, 2000.
- [47] Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*, volume 1. Elsevier, 2001.
- [48] Martin Karplus and J Andrew McCammon. Molecular dynamics simulations of biomolecules. *Nature Structural and Molecular Biology*, 9(9):646, 2002.

- [49] Michael P Allen and Dominic J Tildesley. *Computer simulation of liquids*. Oxford university press, 2017.
- [50] Steve Plimpton, Paul Crozier, and Aidan Thompson. Lammmps-large-scale atomic/molecular massively parallel simulator. *Sandia National Laboratories*, 18:43, 2007.
- [51] Steve Plimpton et al. Lammmps user’s manual. *Sandia National Laboratory*, 2005.
- [52] ZD Sha, RQ Wu, YH Lu, L Shen, M Yang, YQ Cai, YP Feng, and Y Li. Glass forming abilities of binary cu 100- x zr x (34, 35.5, and 38.2 at.%) metallic glasses: A lammmps study. *Journal of Applied Physics*, 105(4):043521, 2009.
- [53] Christina Grindon, Sarah Harris, Tom Evans, Keir Novik, Peter Coveney, and Charles Laughton. Large-scale molecular dynamics simulation of dna: implementation and validation of the amber98 force field in lammmps. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 362(1820):1373–1386, 2004.
- [54] Benjamin FrantzDale, Steven J Plimpton, and Mark S Shephard. Software components for parallel multiscale simulation: an example with lammmps. *Engineering with Computers*, 26(2):205–211, 2010.
- [55] Sauro Succi. *The lattice Boltzmann equation: for fluid dynamics and beyond*. Oxford university press, 2001.
- [56] Christopher M Teixeira. Incorporating turbulence models into the lattice-boltzmann method. *International Journal of Modern Physics C*, 9(08):1159–1175, 1998.
- [57] Huidan Yu, Sharath S Girimaji, and Li-Shi Luo. Dns and les of decaying isotropic turbulence with and without frame rotation using lattice boltzmann method. *Journal of Computational Physics*, 209(2):599–616, 2005.

- [58] Michael R Swift, WR Osborn, and JM Yeomans. Lattice boltzmann simulation of nonideal fluids. *Physical Review Letters*, 75(5):830, 1995.
- [59] AJ Briant, AJ Wagner, and JM Yeomans. Lattice boltzmann simulations of contact line motion. i. liquid-gas systems. *Physical Review E*, 69(3):031602, 2004.
- [60] Peng Yuan and Laura Schaefer. Equations of state in a lattice boltzmann model. *Physics of Fluids*, 18(4):042101, 2006.
- [61] Jonas Tölke. Lattice boltzmann simulations of binary fluid flow through porous media. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 360(1792):535–545, 2002.
- [62] Qinjun Kang, Peter C Lichtner, and Dongxiao Zhang. Lattice boltzmann pore-scale model for multicomponent reactive transport in porous media. *Journal of Geophysical Research: Solid Earth*, 111(B5), 2006.
- [63] MM Dupin, I Halliday, and CM Care. Multi-component lattice boltzmann equation for mesoscale blood flow. *Journal of Physics A: Mathematical and General*, 36(31):8517, 2003.
- [64] Fabian Jansen and Jens Harting. From bijels to pickering emulsions: A lattice boltzmann study. *Physical Review E*, 83(4):046707, 2011.
- [65] FJ Higuera and J Jimenez. Boltzmann approach to lattice gas simulations. *EPL (Europhysics Letters)*, 9(7):663, 1989.
- [66] FJ Higuera, S Succi, and R Benzi. Lattice gas dynamics with enhanced collisions. *EPL (Europhysics Letters)*, 9(4):345, 1989.
- [67] YH Qian, Dominique d’Humières, and Pierre Lallemand. Lattice bgk models for navier-stokes equation. *EPL (Europhysics Letters)*, 17(6):479, 1992.

- [68] Jeffrey J Potoff and Athanassios Z Panagiotopoulos. Critical point and phase behavior of the pure fluid and a lennard-jones mixture. *The Journal of chemical physics*, 109(24):10914–10920, 1998.
- [69] George E Uhlenbeck and Leonard S Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.
- [70] Tung Tsang and Hwa Tang. Velocity autocorrelation functions of lennard-jones fluids. *Physical Review A*, 15(4):1696, 1977.
- [71] Melville S Green. Markoff random processes and the statistical mechanics of time-dependent phenomena. *The Journal of Chemical Physics*, 20(8):1281–1295, 1952.
- [72] Melville S Green. Markoff random processes and the statistical mechanics of time-dependent phenomena. ii. irreversible processes in fluids. *The Journal of Chemical Physics*, 22(3):398–413, 1954.
- [73] Ryogo Kubo. Statistical-mechanical theory of irreversible processes. i. general theory and simple applications to magnetic and conduction problems. *Journal of the Physical Society of Japan*, 12(6):570–586, 1957.
- [74] DA Weitz, DJ Pine, PN Pusey, and RJA Tough. Nondiffusive brownian motion studied by diffusing-wave spectroscopy. *Physical review letters*, 63(16):1747, 1989.
- [75] Santosh Ansumali and Iliya V Karlin. Single relaxation time model for entropic lattice boltzmann methods. *Physical Review E*, 65(5):056312, 2002.
- [76] Santosh Ansumali, Iliya V Karlin, and Hans Christian Öttinger. Minimal entropic kinetic models for hydrodynamics. *EPL (Europhysics Letters)*, 63(6):798, 2003.
- [77] Santosh Ansumali and Iliya V Karlin. Consistent lattice boltzmann method. *Physical review letters*, 95(26):260605, 2005.

- [78] Martin Geier, Martin Schönherr, Andrea Pasquali, and Manfred Krafczyk. The cumulant lattice boltzmann equation in three dimensions: Theory and validation. *Computers & Mathematics with Applications*, 70(4):507–547, 2015.
- [79] AJ Wagner and Qi Li. Investigation of galilean invariance of multi-phase lattice boltzmann methods. *Physica A: Statistical Mechanics and its Applications*, 362(1):105–110, 2006.
- [80] GA Bird. Direct simulation and the boltzmann equation. *The Physics of Fluids*, 13(11):2676–2681, 1970.
- [81] GA Bird. Monte carlo simulation of gas flows. *Annual Review of Fluid Mechanics*, 10(1):11–31, 1978.
- [82] Alejandro L Garcia and Wolfgang Wagner. Time step truncation error in direct simulation monte carlo. *Physics of Fluids*, 12(10):2621–2633, 2000.
- [83] Nicolas G Hadjiconstantinou. Analysis of discretization in the direct simulation monte carlo. *Physics of Fluids*, 12(10):2634–2638, 2000.
- [84] R Adhikari, K Stratford, ME Cates, and AJ Wagner. Fluctuating lattice boltzmann. *EPL (Europhysics Letters)*, 71(3):473, 2005.
- [85] Anthony JC Ladd. Short-time motion of colloidal particles: Numerical simulation via a fluctuating lattice-boltzmann equation. *Physical Review Letters*, 70(9):1339, 1993.
- [86] Burkhard Dünweg, Ulf D Schiller, and Anthony JC Ladd. Statistical mechanics of the fluctuating lattice boltzmann equation. *Physical Review E*, 76(3):036704, 2007.
- [87] Goetz Kaehler and Alexander J Wagner. Derivation of hydrodynamics for multi-relaxation time lattice boltzmann using the moment approach. *Communications in Computational Physics*, 13(3):614–628, 2013.

- [88] A Kuzmin, ZL Guo, and AA Mohamad. Simultaneous incorporation of mass and force terms in the multi-relaxation-time framework for lattice boltzmann schemes. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 369(1944):2219–2227, 2011.
- [89] Hiroaki Yoshida and Makoto Nagaoka. Multiple-relaxation-time lattice boltzmann model for the convection and anisotropic diffusion equation. *Journal of Computational Physics*, 229(20):7774–7795, 2010.
- [90] Frank J Alexander, Shiyi Chen, and JD Sterling. Lattice boltzmann thermohydrodynamics. *Physical Review E*, 47(4):R2249, 1993.
- [91] G Kaehler and AJ Wagner. Fluctuating ideal-gas lattice boltzmann method with fluctuation dissipation theorem for nonvanishing velocities. *Physical Review E*, 87(6):063310, 2013.
- [92] Alexander J Wagner and Kyle Strand. Fluctuating lattice boltzmann method for the diffusion equation. *Physical Review E*, 94(3):033302, 2016.
- [93] D Belardinelli, M Sbragaglia, L Biferale, M Gross, and F Varnik. Fluctuating multi-component lattice boltzmann model. *Physical Review E*, 91(2):023313, 2015.
- [94] Mordechai Bixon and Robert Zwanzig. Boltzmann-langevin equation and hydrodynamic fluctuations. *Physical Review*, 187(1):267, 1969.
- [95] EM Foard and AJ Wagner. Enslaved phase-separation fronts in one-dimensional binary mixtures. *Physical Review E*, 79(5):056710, 2009.
- [96] Jill P Dahlburg, David Montgomery, and Gary D Doolen. Noise and compressibility in lattice-gas fluids. *Physical Review A*, 36(5):2471, 1987.
- [97] EM Foard and AJ Wagner. Survey of morphologies formed in the wake of an enslaved phase-separation front in two dimensions. *Physical Review E*, 85(1):011501, 2012.

- [98] Dieter Forster. *Hydrodynamic fluctuations, broken symmetry, and correlation functions*. CRC Press, 2018.
- [99] Ronald Forrest Fox and George E Uhlenbeck. Contributions to nonequilibrium thermodynamics. ii. fluctuation theory for the boltzmann equation. *The Physics of Fluids*, 13(12):2881–2890, 1970.
- [100] Irina Ginzburg. Prediction of the moments in advection-diffusion lattice boltzmann method. i. truncation dispersion, skewness, and kurtosis. *Physical Review E*, 95(1):013304, 2017.
- [101] Thomas Blommel and Alexander J Wagner. Integer lattice gas with monte carlo collision operator recovers the lattice boltzmann method with poisson-distributed fluctuations. *Physical Review E*, 97(2):023310, 2018.

# APPENDIX. SUPPLEMENTARY CODE

## A.1. Code for Molecular Dynamic Lattice Gas

The input file contains Molecular Dynamics data for 99 856 particles with initial box size of 1 000 and temperature of 50 unit of Lennard-Jones for 2 000 000 time steps. This box was divided by  $25 \times 25$  lattice (length of each lattice is 40).

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <errno.h>

#define LX 25
#define LY 25
#define velvecX 7
#define velvecY 7
#define maxatoms 99856
#define Box 1000

typedef struct pos {int x; int y;} pos;
pos c[2][maxatoms];
int atoms=0;
int velvec[velvecX][velvecY][2];
double n[2000][velvecX][velvecY][LX][LY];
double f[velvecX][velvecY];
double pr[velvecX][velvecY];
```

```

double xmin,xmax,ymin,ymax,zmin,zmax,p,sum,velvecn,a1;

void ReadFile(double ns,int dt){
    int first=0;
    FILE *fp;
    char *str;
    int it=0,er=0,pno,type;
    memset(&n[0][0][0][0][0],0,2000*velvecX*velvecY*LX*LY*sizeof(double));
    for (int i=0; i<velvecX; i++)
        for (int j=0; j<velvecY; j++)
            for (int k=0; k<2; k++)
                velvec[i][j][k]=(1-k)*(i-((velvecX-1)/2))+k*(j-((velvecY-1)/2));
    a1=(26.1068*(exp(-1.96*tau)-1)+51.1694*tau);
    fp=fopen("dump.relax","r");
    for (int j=0; j<20001; j++){
        errno = 0;
        er=fscanf(fp,"%ms\n",&str);
        free(str);
        errno = 0;
        er=fscanf(fp,"%ms\n",&str);
        free(str);
        errno = 0;
        er=fscanf(fp,"%i",&it);
        printf("iteration: %i\n",it);
        errno = 0;
        er=fscanf(fp,"%ms\n",&str);
        free(str);
    }
}

```



```

er=fscanf(fp,"%ms\n",&str);
free(str);
for (int i=0; i<maxatoms; i++){
    fscanf(fp,"%i",&pno);
    fscanf(fp,"%i",&type);
    fscanf(fp,"%le",&p);
    if (it % dt==0) {
        c[0][pno-1].x=(int) c[1][pno-1].x;
        c[1][pno-1].x=(int) fmod(floor(p*LX),LX);
    }
}

```

```

}
fscanf(fp,"%le",&p);
if (it % dt==0){
    c[0][pno-1].y=(int) c[1][pno-1].y;
    c[1][pno-1].y=(int) fmod(floor(p*LY),LY);
}
if ((it % dt==0)){
    if (first==1){
        int dx= c[1][pno-1].x-c[0][pno-1].x;
        if (abs(dx+LX)<abs(dx)) dx=dx+LX;
        if (abs(dx-LX)<abs(dx)) dx=dx-LX;
        int dy= c[1][pno-1].y-c[0][pno-1].y;
        if (abs(dy+LY)<abs(dy)) dy=dy+LY;
        if (abs(dy-LY)<abs(dy)) dy=dy-LY;
        for (int i=0; i<velvecX; i++)
            for (int j=0; j<velvecY; j++)
                if ((dx==velvec[i][j][0])&&(dy==velvec[i][j][1]))
                    n[(it/dt)-1][i][j][c[1][pno-1].x][c[1][pno-1].y]++;
    }
}
fscanf(fp,"%le",&p);
fscanf(fp,"%le",&p);
fscanf(fp,"%le",&p);
fscanf(fp,"%le",&p);
}
first=1;
}

```

```

memset(&f[0][0],0,velvecX*velvecY*sizeof(double));
memset(&pr[0][0],0,velvecX*velvecY*sizeof(double));
for (int i=0; i<velvecX; i++){
    for (int j=0; j<velvecY; j++){
        for (int t=0; t<(ns); t++){
            for (int k=0; k<LX; k++){
                for (int l=0; l<LY; l++){
                    f[i][j]+=n[t][i][j][k][l];
                    if (n[t][i][j][k][l]>0) pr[i][j]+=1;
                }
            }
        }
    }
}

memset(&velvecn,0,sizeof(double));
for (int i=0; i<velvecX; i++)
    for (int j=0; j<velvecY; j++)
        if (f[i][j]>0) velvecn+=1;
memset(&sum,0,sizeof(double));
for (int i=0; i<velvecX; i++)
    for (int j=0; j<velvecY; j++)
        sum+=f[i][j];
}

void writefiles(double ns,int dt){
    FILE *res;
    res=fopen("output.dat","a");

```

```

fprintf(res,"%g %g",dt*0.0001,a1/((Box/LX)*(Box/LX)));
for (int i=0; i<velvecX; i++)
    for (int j=0; j<velvecY; j++)
        fprintf(res," %g",f[i][j]/(LX*LY*(ns)));
fprintf(res," %g",sum/(LX*LY*(ns)));
for (int i=0; i<velvecX; i++)
    for (int j=0; j<velvecY; j++)
        fprintf(res," %g",f[i][j]/(LX*LY*(ns)*(maxatoms/(LX*LY))));
fprintf(res," %g",sum/(LX*LY*(ns)*(maxatoms/(LX*LY))));
fprintf(res," %g",velvecn);
for (int i=0; i<velvecX; i++)
    for (int j=0; j<velvecY; j++)
        fprintf(res," %g",pr[i][j]/(LX*LY*(ns)));
fprintf(res,"\n");
fclose(res);
}

void main(){
    for (int i=1 ;i<2000; i=ceil(i*1.1)){
        int timestep=(i*1000);
        double n=(floor(2000000/timestep));
        ReadFile(n,timestep);
        writefiles(n,timestep);
    }
}

```

## A.2. Code for Mean Square Displacement

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <errno.h>

#define maxatoms 99856
#define SaveSteps 100

double sum[SaveSteps];
int atoms=0,TimeCount=0;
typedef struct state {int it; double r[maxatoms][2];} state;
state c[SaveSteps];
double xmin,xmax,ymin,ymax,zmin,zmax,p;

void ReadFile(){
    int st=-1;
    FILE *fp;
    char *str;
    int it=0,er=0,pno,type;
    memset(&sum[0],0,SaveSteps*sizeof(double));
    fp=fopen("dump.relax","r");
    for (int j=0;j<20001;j++){
        errno = 0;
        er=fscanf(fp,"%ms\n",&str);
        free(str);
    }
}
```

```
errno = 0;
er=fscanf(fp,"%ms\n",&str);
free(str);
errno = 0;
er=fscanf(fp,"%i",&it);
printf("iteration: %i\n",it);
errno = 0;
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
errno = 0;
er=fscanf(fp,"%i",&atoms);
printf("no atoms: %i\n",atoms);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
```

```
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
fscanf(fp,"%le",&xmin);
fscanf(fp,"%le",&xmax);
fscanf(fp,"%le",&ymin);
fscanf(fp,"%le",&ymax);
fscanf(fp,"%le",&zmin);
fscanf(fp,"%le",&zmax);
er=fscanf(fp,"%ms\n",&str);
free(str);
```

```

er=fscanf(fp,"%ms\n",&str);
free(str);
if (st==SaveSteps-1){
    memmove(&c[0].it,&c[1].it,(SaveSteps-1)*sizeof(state));
    TimeCount++;
}
else st++;
c[st].it=it;
double Dx=xmax-xmin;
double Dy=ymax-ymin;
for (int i=0;i<maxatoms;i++){
    fscanf(fp,"%i",&pno);
    fscanf(fp,"%i",&type);
    fscanf(fp,"%le",&p);
    c[st].r[pno-1][0] = p;
    fscanf(fp,"%le",&p);
    c[st].r[pno-1][1] = p;
    if (st==SaveSteps-1){
        for (int dt=0; dt<SaveSteps; dt++){
            double dist=1*1+1*1;
            for (int dx=-1;dx<2;dx++){
                for (int dy=-1;dy<2;dy++){
                    double dist_tmp=
                    pow((c[st].r[pno-1][0]-c[st-dt].r[pno-1][0]+dx)*1,2)
                    +pow((c[st].r[pno-1][1]-c[st-dt].r[pno-1][1]+dy)*1,2);
                    if (dist_tmp<dist) dist=dist_tmp;
                }
            }
        }
    }
}

```

```

        sum[dt] +=dist;
    }
}
fscanf(fp,"%le",&p);
fscanf(fp,"%le",&p);
fscanf(fp,"%le",&p);
fscanf(fp,"%le",&p);
}
}
}

void writefiles(){
    FILE *res;
    res=fopen("output.dat","w");
    for (int dt=1;dt<SaveSteps;dt=ceil(dt*1.1))
        fprintf(res,"%g %g\n",((double)dt*(c[1].it-c[0].it)/10000),
                (sum[dt]/(2.*maxatoms*TimeCount))*1000*1000);
    fclose(res);
}

void main(){
    ReadFile();
    writefiles();
}

```

### A.3. Code for Velocity Autocorrelation Function

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <errno.h>

#define maxatoms 99856
#define SaveSteps 100

int TimeCount=0;
double sum[SaveSteps][2],R[SaveSteps];
int atoms=0;
double p;
typedef struct state {int it; double r[maxatoms][2];} state;
state c[SaveSteps];
double xmin,xmax,ymin,ymax,zmin,zmax;

void ReadFile(){
    int st=-1;
    FILE *fp;
    char *str;
    int it=0,er=0,pno,type;
    memset(&sum[0],0,SaveSteps*sizeof(double));
    fp=fopen("dump.relax","r");
    for (int j=0;j<2001;j++){
        errno = 0;
```

```
er=fscanf(fp,"%ms\n",&str);
free(str);
errno = 0;
er=fscanf(fp,"%ms\n",&str);
free(str);
errno = 0;
er=fscanf(fp,"%i",&it);
printf("iteration: %i\n",it);
errno = 0;
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
errno = 0;
er=fscanf(fp,"%i",&atoms);
printf("no atoms: %i\n",atoms);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
```

```
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
fscanf(fp,"%le",&xmin);
fscanf(fp,"%le",&xmax);
fscanf(fp,"%le",&ymin);
fscanf(fp,"%le",&ymax);
fscanf(fp,"%le",&zmin);
fscanf(fp,"%le",&zmax);
er=fscanf(fp,"%ms\n",&str);
free(str);
```

```

er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
if (st==SaveSteps-1){
    memmove(&c[0].it,&c[1].it,(SaveSteps-1)*sizeof(state));
    TimeCount++;
}
else st++;
c[st].it=it;
for (int i=0;i<maxatoms;i++){
    fscanf(fp,"%i",&pno);
    fscanf(fp,"%i",&type);
    fscanf(fp,"%le",&p);
    fscanf(fp,"%le",&p);
    fscanf(fp,"%le",&p);
    fscanf(fp,"%le",&p);
    fscanf(fp,"%le",&p);
    c[st].r[pno-1][0] = p;
    fscanf(fp,"%le",&p);
    c[st].r[pno-1][1] = p;
    if (st==SaveSteps-1){
        for (int dt=0; dt<SaveSteps; dt++){
            sum[dt][0]+=(c[dt].r[pno-1][0]*c[0].r[pno-1][0]);
            sum[dt][1]+=(c[dt].r[pno-1][1]*c[0].r[pno-1][1]);
        }
    }
}
fscanf(fp,"%le",&p);

```

```

    }
}
memset(R,0,SaveSteps*sizeof(double));
for (int i=0;i<SaveSteps;i++){
    for (int j=0;j<SaveSteps;j++){
        if ((i-j)>0) R[i]+=(sum[j][0]+sum[j][1]
            +sum[j+1][0]+sum[j+1][1])*(i-j)*.01;
    }
}
}

void writefiles(){
    FILE *res;
    res=fopen("output.dat","w");
    for (int dt=1;dt<SaveSteps;dt=ceil(dt*1.1))
        fprintf(res,"%g %g %g\n",((double)dt*(c[1].it-c[0].it)/10000),
            sum[dt][0]/(maxatoms*TimeCount),R[dt]/(maxatoms*TimeCount));
    fclose(res);
}

void main(){
    ReadFile();
    writefiles();
}

```

#### A.4. Code for Measured Correlators $\langle \delta x_1 \delta x_2 \rangle$

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <errno.h>

#define maxatoms 99856
#define nohist 10000
#define tau .2

double c[2][maxatoms][2],n[nohist],delx[nohist],r[4];
int atoms=0,time=0;
double xmin,xmax,ymin,ymax,zmin,zmax,p,sum,a1,delt;

void ReadFile(){
    FILE *fp;
    char *str;
    int it=0,er=0,pno,type;

    memset(&n[0],0,nohist*sizeof(double));
    memset(&delx[0],0,nohist*sizeof(double));
    a1=(26.1068*(exp(-1.96*tau)-1)+51.1694*tau);
    delt=tau*10000;
    fp=fopen("dump.relax","r");
    for (int t=0; t<20001; t++){
        errno = 0;
```

```
er=fscanf(fp,"%ms\n",&str);
free(str);
errno = 0;
er=fscanf(fp,"%ms\n",&str);
free(str);
errno = 0;
er=fscanf(fp,"%i",&it);
printf("iteration: %i\n",it);
errno = 0;
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
errno = 0;
er=fscanf(fp,"%i",&atoms);
printf("no atoms: %i\n",atoms);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
```

```
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
fscanf(fp,"%le",&xmin);
fscanf(fp,"%le",&xmax);
fscanf(fp,"%le",&ymin);
fscanf(fp,"%le",&ymax);
fscanf(fp,"%le",&zmin);
fscanf(fp,"%le",&zmax);
er=fscanf(fp,"%ms\n",&str);
free(str);
```

```

er=fscanf(fp,"%ms\n",&str);
free(str);
er=fscanf(fp,"%ms\n",&str);
free(str);
for (int i=0;i<maxatoms;i++){
    fscanf(fp,"%i",&pno);
    fscanf(fp,"%i",&type);
    fscanf(fp,"%le",&p);
    c[0][pno-1][0]= p;
    fscanf(fp,"%le",&p);
    c[0][pno-1][1]= p;
    fscanf(fp,"%le",&p);
    fscanf(fp,"%le",&p);
    fscanf(fp,"%le",&p);
    fscanf(fp,"%le",&p);
}
if ((it>0)&&((fmod(it,delt))==0)){
    for (int i=0;i<maxatoms;i++){
        for (int j=i+1;j<maxatoms;j++){
            double dx= c[0][j][0]-c[0][i][0];
            if (fabs(dx+1)<fabs(dx)) dx=dx+1;
            if (fabs(dx-1)<fabs(dx)) dx=dx-1;
            dx=fabs(dx);
            double dy= c[0][j][1]-c[0][i][1];
            if (fabs(dy+1)<fabs(dy)) dy=dy+1;
            if (fabs(dy-1)<fabs(dy)) dy=dy-1;
            dy=fabs(dy);

```

```

double d=sqrt(dx*dx+dy*dy);
int k=d*sqrt(2)*nohist;
if (k<nohist){
    n[k]++;
    r[0]=c[0][i][0]-c[1][i][0];
    if (fabs(r[0]+1)<fabs(r[0])) r[0]=r[0]+1;
    if (fabs(r[0]-1)<fabs(r[0])) r[0]=r[0]-1;
    r[1]=c[0][j][0]-c[1][j][0];
    if (fabs(r[1]+1)<fabs(r[1])) r[1]=r[1]+1;
    if (fabs(r[1]-1)<fabs(r[1])) r[1]=r[1]-1;
    r[2]=c[0][i][1]-c[1][i][1];
    if (fabs(r[2]+1)<fabs(r[2])) r[2]=r[2]+1;
    if (fabs(r[2]-1)<fabs(r[2])) r[2]=r[2]-1;
    r[3]=c[0][j][1]-c[1][j][1];
    if (fabs(r[3]+1)<fabs(r[3])) r[3]=r[3]+1;
    if (fabs(r[3]-1)<fabs(r[3])) r[3]=r[3]-1;
    delx[k]+=(( r[0])*(r[1]) )+( (r[2])*(r[3]) ) )
                *(xmax-xmin)*(xmax-xmin);
}
}
}
time+=1;
}
if((fmod(it,delt))==0){
    for (int i=0;i<maxatoms;i++){
        c[1][i][0]=c[0][i][0];
        c[1][i][1]=c[0][i][1];
    }
}

```

```

    }
}
}
for (int k=0; k<nohist; k++) if (n[k]>0) delx[k]/=n[k];
for (int k=0; k<nohist; k++)
    sum+=n[k];
}

void writefiles(){
    FILE *res;
    res=fopen("output.dat","a");
    for (int i=0;i<nohist;i++) {
        fprintf(res,"%g %g %g %g\n", (xmax-xmin)*i*sqrt(2)/((2*nohist)),
            delx[i],n[i]/time,sum/time);
    }
    fclose(res);
}

void main(){
    ReadFile();
    writefiles();
}

```