University of Pennsylvania

Term paper for MSE 575

# Monte Carlo method for lattice gas systems

Yuding Ai

January 27, 2017

**Abstract**

Inspired by review articles [1], [4] and several works, such as [3] and [5], from Kurt Binder, and [8] from Daan Frenkel, we found that Monte Carlo method is closely related to our study of Statistical Mechanics. In order to get a better understanding and to appreciate the power of statistical mechanic, we wrote this term paper to apply our knowledge from statistical mechanics into some real computational tasks. This work will be focus on reviewing Monte Carlo method for both close and open lattice gas systems. There will be 3 main sections in this paper. For the first part, we will briefly review the statistical mechanics background behind the Monte Carlo method. In the second section, we will start to introduce the concept of Monte carlo simulation and make connection to the statistical mechanics that have been reviewed in section one. In section 3, we will apply the Monte carlo method to simulate both homogeneous (weakly/non-interacting particles) and inhomogeneous(interacting) lattice gas systems.

# 1 Introduction

Monte carlo methods is a well established tool for many field of computational studies such as fluids mechanics, thermal dynamics, computational biology and even in financial engineering. If one trace back to the origin and the fundamental basis of such powerful tool, not surprisingly, we get back to our statistical mechanics.

In the class MSE 575, we have covered fluctuation and desperation theorem and introduced three ensembles: micro-canonical (constant E,V,N), canonical (constant T,V,N) and grand canonical (constant T,V,$\mu$) ensembles corresponding to isolated, close and open system respectively. In order to further understand these concepts and apply them into developing some real (and 'cool') simulation model, we employ the idea of Monte carlo method and study the lattice gas model. These models could serve as a "virtual lab" and perform "virtual measurements" as to simulate physical quantities or behaviors. Particularly, Computer simulations can validate the theoretical simplifications applied in some theories and sometimes even predicts the outcomes with reasonable accuracy.

In this term paper, we will focus on reviewing and implementing the monte carlo model for both close and open lattice gas systems in an introductory level. To start with, we first review the partition functions in canonical and grand canonical ensembles and derive the probability of occurrence of the micro-state for each cases. Next we will briefly introduce the concept of Monte carlo simulation and then use the knowledge of statistical mechanics to construct the actual algorithm for both canonical and Grand canonical monte carlo method. Furthermore, inspired by reference [12] and [13] , we will also briefly introduce a more advanced Monte carlo method called biased sampling, together with one of the state-of-art approach, namely Wang-Laudau sampling. As one of the most popular application for those monte carlo models is to study the phase transition phenomenon, we will also briefly show how the gas-liquid translation took place on our lattice gas model.

Last, we will mention some more advanced and nice recent works based on the Monte carlo method for lattice system.

# 2 From statistical mechanics to Monte Carlo simulations

## 2.1 A short review of partition functions

The partition function[1] describes the statistical properties of a system and contains all the essential information about that system. For a closed system at a constant temperature T in a container with constant volume V and constant numbers of particle N, the discrete version of canonical partition function Q(T,V,N) is defines as

$$Q(T,V,N) = \sum_{allstates} e^{-\beta \mathcal{H}} \tag{1}$$

where $\beta$ is defined as $\beta = 1/k_B T$, $k_B$ is the Boltzmann constant, $\mathcal{H}$ is the Hamiltonian and T,V,N are absolute temperature, volume and number of particles respectively.
For a classical system, we could rewrite the $Q(T, V.N)$ into an integral form

$$Q(T, V.N) \sim \int\int ... \int e^{-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)} d\mathbf{p}_1 d\mathbf{p}_2 ... d\mathbf{p}_N d\mathbf{r}_1 d\mathbf{r}_2 ... d\mathbf{r}_N \tag{2}$$

where $\mathbf{r_i}$ and $\mathbf{p_i}$ are the position and momentum vectors for ith particles. In this study, we consider distinguishable particles that each particle has its unique position and momentum, and for a quantum treatment (say if we are dealing with nano-scale particles), a correction factor $\frac{1}{h^{3N}}$, where h is the Plank's constant, must be introduced. This gives

$$Q(T,V,N) = \frac{1}{h^{3N}} [\int e^{-\beta KE(\mathbf{p}^N)} d\mathbf{p}_1 d\mathbf{p}_2 ... d\mathbf{p}_N][\int e^{-\beta PE(\mathbf{r}^N)} d\mathbf{r}_1 d\mathbf{r}_2 ... d\mathbf{r}_N] \tag{3}$$

where KE and PE and kinetic energy and potential energy. Since kinetic energy can be written as

$$KE(\mathbf{p}^N) = \sum_{i=0}^{N} \frac{(\mathbf{p}_{i,x}^2 + \mathbf{p}_{i,y}^2 + \mathbf{p}_{i,z}^2)}{2m_i} \tag{4}$$

Assuming that $m_i = m = const$ (same type of particles), then one could solve for
$\int e^{-\beta KE(\mathbf{p}^N)} d\mathbf{p}_1 d\mathbf{p}_2 ... d\mathbf{p}_N$ and obtain

$$\int e^{-\beta KE(\mathbf{p}^N)} d\mathbf{p}_1 d\mathbf{p}_2 ... d\mathbf{p}_N = (\frac{2\pi m}{\beta})^{3N/2} = (2\pi m k_B T)^{\frac{3}{2}} \tag{5}$$

Then we can rewrite the partition function $Q(T,V,N)$ as

$$Q(T,V,N) = \frac{1}{h^{3N}} (\frac{2\pi m}{\beta})^{3N/2} \int e^{-\beta PE(\mathbf{r}^N)} d\mathbf{r}_1 d\mathbf{r}_2 ... d\mathbf{r}_N \tag{6}$$

[1] All the derivations in this section are following Reference [8]

We have introduced the quantum density $n_q = (\frac{2\pi m k_B T}{h^2})^{\frac{3}{2}}$ in MSE 575 and thus substituting $n_q$ into the above expression gives

$$Q(T,V,N) = n_q^N \int e^{-\beta PE(\mathbf{r}^N)} d\mathbf{r}_1 d\mathbf{r}_2 ... d\mathbf{r}_N \tag{7}$$

Recall that the configurational partition function Z(T,V,N) (the one we were dealing with in class 575) is defined as

$$\begin{aligned} Z &= \sum_{i=1}^{N_c} e^{-PE_i/k_B T} \\ Z(T,V,N) &= \int e^{-\beta PE(\mathbf{r}^N)} d\mathbf{r}_1 d\mathbf{r}_2 ... d\mathbf{r}_N \end{aligned} \tag{8}$$

And finally, we can write the partition function Q(T,V,N) as

$$Q(T,V,N) = n_q^N(T) \times Z(T,V,N) \tag{9}$$

Then, the partition function for an open system $\Xi(T,V,\mu)$ (grand canonical ensemble) with constant temperature T, volume V and chemical potential $\mu$ is given by

$$\Xi(T,V,\mu) = \sum_{N=0}^{\infty} Q(T,V,N) e^{\beta\mu N} \tag{10}$$

$$= \sum_{N=0}^{\infty} n_q^N(T) Z(T,V,N) e^{\beta\mu N} \tag{11}$$

## 2.2 Probability of occurrence for the microstate

### 2.2.1 Close system

For a close system with canonical ensemble(constant T ,or equivalently $< E >$,V,N), the total energy E fluctuated and the probability density follows Boltzmann's distribution

$$\wp(\mathbf{p}^N, \mathbf{r}^N) \propto e^{-\beta \mathcal{H}(\mathbf{p}^N \mathbf{r}^N)} \tag{12}$$

and one can write it precisely in terms of partition function Q(T,V,N) as

$$\wp(\mathbf{p}^N, \mathbf{r}^N) = \frac{e^{-\beta KE(\mathbf{p}^N)} e^{-\beta UE(\mathbf{r}^N)}}{Q(T,V,N)} \tag{13}$$

Together with Equation 9, one obtain the following expression:

$$\wp(\mathbf{p}^N, \mathbf{r}^N) = \frac{e^{-\beta KE(\mathbf{p}^N)} e^{-\beta UE(\mathbf{r}^N)}}{Q(T,V,N)} = \frac{e^{-\beta KE(\mathbf{p}^N)} e^{-\beta UE(\mathbf{r}^N)}}{n_q^N(T) \times Z(T,V,N)} \tag{14}$$

Since in canonical ensemble, we can separate the variable for momentum $\mathbf{p}$ and position $\mathbf{r}$ and obtain

$$\wp(\mathbf{p}^N, \mathbf{r}^N) = \wp(\mathbf{p}^N)\wp(\mathbf{r}^N) \tag{15}$$

where

$$\begin{aligned} \wp(\mathbf{p}^N) &= n_q^N e^{-\beta KE(\mathbf{p}^N)} \\ \wp(\mathbf{r}^N) &= \frac{e^{-\beta UE(\mathbf{r}^N)}}{Z(T,V,N)} \end{aligned} \tag{16}$$

Finally, as we will not involve kinetics in this study, there is no momentum here and hence the probability of occurrence for each micro-state "j"(configuration 'j'): $p_j(N)$ is solely depend on $\mathbf{r}^N$, given by

$$p_j = \wp(\mathbf{r}^N) d\mathbf{r}^N = \frac{e^{-\beta U_j} d\mathbf{r}^N}{Z(T,V,N)} \tag{17}$$

### 2.2.2 Open system

For an open system we have grand canonical ensemble (constant T,V,$\mu$) and both the energy and the number of particles fluctuate. The probability density of a system with N particles $\wp^{GC}(N)$ is given by

$$\wp^{GC}(N) = \frac{Q(T,V,N)e^{\beta\mu N}}{\sum_{N=0}^{\infty} Q(T,V,N)e^{\beta\mu N}} = \frac{Q(T,V,N)e^{\beta\mu N}}{\Xi(V,T,\mu)} = \frac{\int n_q^N(T) e^{-\beta U(r^N)} dr_1 dr_2 \ldots dr_N e^{\beta\mu N}}{\Xi(V,T,\mu)} \tag{18}$$

where $\Xi(V,T,\mu)$ is the grand partition function. As each micro-state in the GC system corresponds to both a set of configuational coordinates and a value of the particle number N. Therefore, the probability of occurrence for each micro-state "j"(configuration 'j' with N particles): $p_j^{GC}(N)$ is given by

$$\begin{aligned} p_j^{GC}(N) &= \wp^{GC}(\mathbf{r}^N, N) d\mathbf{r}^N \\ &= n_q(T)^N e^{-\beta U_j + \beta\mu N} \times \frac{d\mathbf{r}^N}{\Xi(V,T,\mu)} \end{aligned} \tag{19}$$

## 2.3 End note

In this section, we have reviewed the partition function in a general approach and found the formula for both canonical and grand canonical ensembles. Then followed by Reference [8], we have carefully derived the probability of occurrence for both closed (canonical) and open (grand canonical) systems and this probability of occurrence will play a crucial rule later in forming the actual algorithm for the Monte carlo simulation.

# 3 Monte Carlo Simulation and Markov process

## 3.1 Monte Carlo Simulation

**Monte Carlo Simulation**, which relies on repeated random sampling, is collection of computational methods that are used to describe any technique that approximates solutions to quantitative problems through statistical sampling. Utilizing monte carlo simulation, we divide a complicated event into a huge set of simple events (and sometimes even seems trivial) and then compute each simple event separately so that the modern computer can easily handle. In the case of simulating the thin film growth, the real situation is that millions of particles are depositing onto a silicon wafer almost simultaneously, and this is not computable for our modern computers. However, with employing the idea of monte carlo method, one can then separate this super complicated event into millions of simple events so that computer only deal with one movement with one particle at a time and then simulate the next movement for the next particle and so on. At the end of the monte carlo simulation, one could then get a statistical approximation that is consistent with the real outcome.

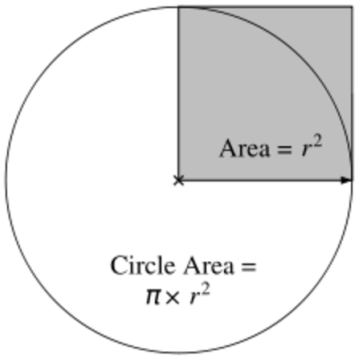
### 3.1.1 Simulate $\pi$

The simplest pedagogic example of Monte Carlo simulation would be the simulation of $\pi$. Suppose we have a circle inscribed in a square as depicted in Figure 1[2], we can estimate the value of $\pi$ by doing a series of experiments of throwing darts onto this square randomly and after big enough amount of 'throwing darts event', the value of $\pi$ could then be calculated through the following geometry recipe:

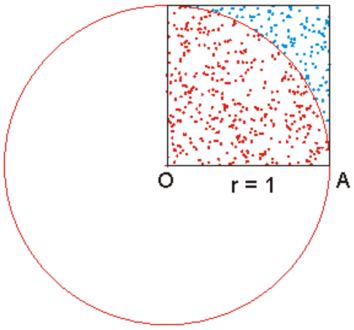
$$\begin{aligned} Area\ of\ Square &= r^2 \\ Area\ of\ Circle &= \pi r^2 \\ pi \approx 4 \times \frac{Number\ of\ Darts\ in\ Circle}{Number\ of\ Darts\ in\ Square} & \end{aligned} \tag{20}$$

Even though, it seems that throwing darts is a trivial way to estimate a value with infinite non-repeating decimals like $\pi$, nevertheless, as long as the amount of this random 'measurement' is huge enough, the value of $\pi$ could be calculated very accurately. Through such example, we see how the Monte carlo method could accurately simulate a physical quantity by doing a huge sets of random 'trivial' measurements.

[2]Picture obtained from online source: http://drewconway.com/zia/2013/3/26/happy-pi-day-now-go-estimate-it




(a) The square




(b) Throwing darts onto the square

Figure 1: Simulate $pi$

## 3.2 Markov process

Each step of a monte carlo simulation can be described as a Markov process. An $N_{iteration}$ of Markov process will generate a Markov chain that contains the statistical information of interest as the result from monte carlo simulations.Figure 2 describes how a Markov process works, in which $p^{mov}(i \to j)$ is the movement probability that

$$p^{mov}(i \to j) = p^{acc}(i \to j) \times p^{prop}(i \to j) \tag{21}$$

where where $p^{acc}(i \to j)$ is the movement acceptance probability and $p^{prop}(i \to j)$ is the movement proposal probability. Since a Markov process obeys detailed balance condition, we have

$$p_i p^{mov}(i \to j) = p_j p^{mov}(j \to i) \tag{22}$$

where $p_i$ and $p_j$ are probability of occurrence of state i and state j respectively. where $p_i \propto e^{-\beta E_i}$ Combine Equation 21 and Equation 22, we obtained a metropolis form expression for the movement acceptance probability

$$p^{acc}(i \to j) = min[1, \frac{p^{prop}(j \to i)p_j}{p^{prop}(i \to j)p_i}] \tag{23}$$

which is the so-called **Metropolis algorithm**, and this movement acceptance probability will serve as the key equation to the simulation in next section.

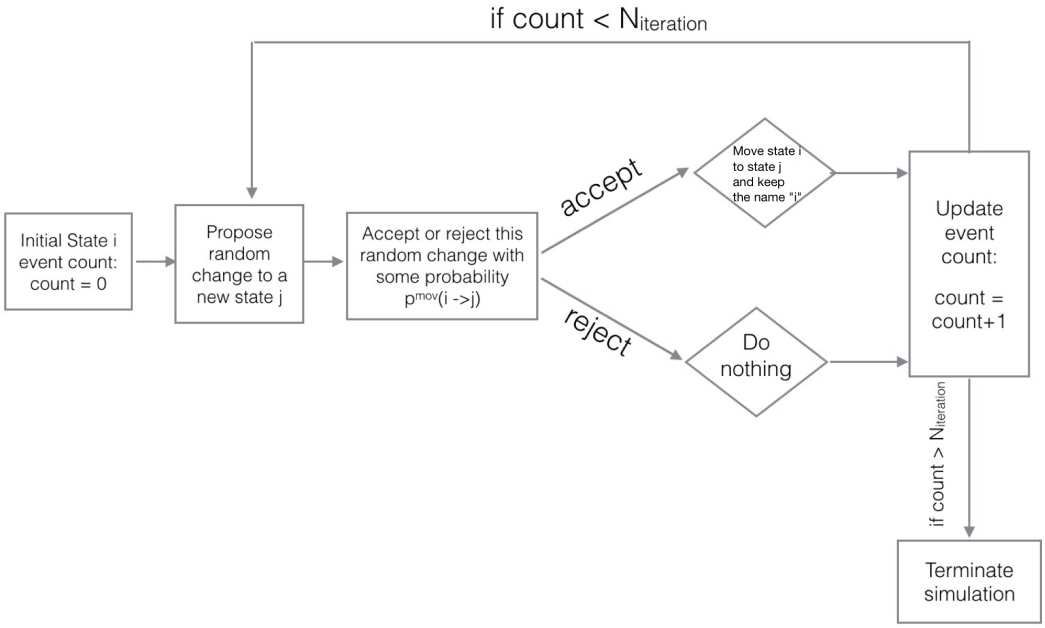



Figure 2: Flowchart for Markov process

## 3.3 The CMC (canonical monte carlo) method

The CMC method is used to simulate a closed system[3] and one good example is to simulate the Ising model. For simplicity, here we will stick with the simple Lattice gas model. The CMC method deals with fixing canonical ensembles: volume V, $< E >$ (equivalent to fix temperature T) and numbers of particles N while allowing the total Energy to fluctuate. Consider a large step configuration move (reaches equilibrium faster compared to small step move) by randomly displace an arbitrary lattice gas into an arbitrary site as the following:

1. Select one particle randomly
2. Place into a randomly chosen site
3. decide whether to accept or reject the move following the Metropolis algorithm.

Figure 3[4] graphically describes how such movement works. Since this large step configuration move has to be ergodic because each particle can reach any lattice site in principle, therefore

$$p^{prop}(i \to j) = p^{prop}(j \to i) = \frac{1}{N} \times \frac{1}{V} \tag{24}$$

[3]this section is followed by RLC Vink's lecture notes
[4]Again, from RLC Vink's lecture note

According to Equation 23, the acceptance probability for large step move is then

$$
\begin{aligned}
p^{acc}(i \to j) &= min[1, \frac{p^{prop}(j \to i)p_j}{p^{prop}(i \to j)p_i}] \\
p^{acc}(i \to j) &= min[1, \frac{p_j}{p_i}] \\
p^{acc}(i \to j) &= min[1, e^{-\beta(PE_j - PE_i)}]
\end{aligned} \tag{25}
$$

where PE (depending on configuration) is the potential energy. This movement acceptance law

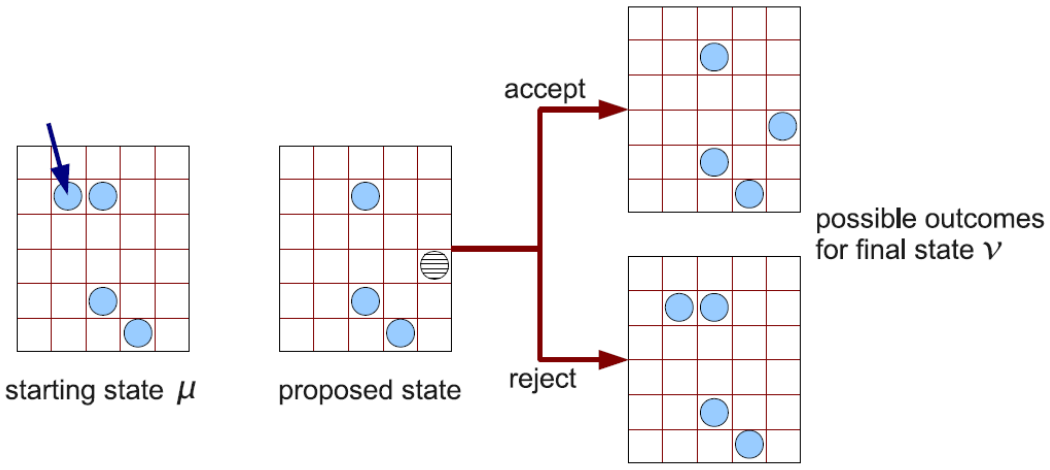



Figure 3: Large step configuration movement process

shall be followed on step 3 of each Monte carlo iteration and serve as the core equation for the Canonical monte carlo simulation.

## 3.4 The GCMC (grand canonical monte carlo) method

The GCMC method is used for simulating an open system and one best example is to simulate particles reversibly adsorb to a surface or interface from a bulk reservoir, i.e Thin film growth. The GCMC method[5] dealing with fixing the grand canonical ensembles: volume V, Temperature T and chemical potential $\mu$. Each iteration of the GCMC movement is described as the following:

1. decide whether to insert for remove a particle with 50% by 50% chance.
2. attempt to insert or remove the particle.
3. decide whether to accept or reject this movement following the Metropolis algorithm.

[5]Derivation followed by reference [10].

Consider the acceptance probabilities: $p^{acc}(N \to N+1)$ for addition and $p^{acc}(N+1 \to N)$ for deletion. Without loss of generosity, assume $N_j = N_i + 1$ such that we have insertion move: $i \to j$ and deletion move: $j \to i$. The proposal probability for inserting particles is $p^{prop}(i \to j) = dr/V$ (from the fact that we pick a random location in space and place the particle there) and for deletion $p^{prop}(j \to i) = 1/N_j$ (For each particle on the lattice, there is a $1/N_j$ chance to be chosen to remove). For GC system, combine Equation 19 and Equation 23, we derived the acceptance probability for insertion move as the following

$$
\begin{aligned}
p^{acc}(i \to j) &= min[1, \frac{p^{prop}(j \to i) p_j^{GC}(N_j)}{p^{prop}(i \to j) p_i^{GC}(N_i)}] \\
p^{acc}(i \to j) &= min[1, \frac{\frac{1}{N_j} n_q^{N_j} e^{-\beta U_j + \beta \mu_j N_j} \times \frac{d\mathbf{r}^{N_j}}{\Xi(V,T,\mu)}}{\frac{dr}{V} n_q^{N_i} e^{-\beta U_i + \beta \mu_i N_i} \times \frac{d\mathbf{r}^{N_i}}{\Xi(V,T,\mu)}}]
\end{aligned} \tag{26}
$$

Due to the fact that $\mu$ is a fixed constant, $\mu_i = \mu_j = \mu$; Substitute $N_j = N_i + 1$ into the above equations, and rearrange terms, we have

$$
\begin{aligned}
\frac{\frac{1}{N_j} n_q^{N_j} e^{-\beta U_j + \beta \mu_j N_j} \times \frac{d\mathbf{r}^{N_j}}{\Xi(V,T,\mu)}}{\frac{dr}{V} n_q^{N_i} e^{-\beta U_i + \beta \mu_i N_i} \times \frac{d\mathbf{r}^{N_i}}{\Xi(V,T,\mu)}} &= \frac{\frac{1}{N_i+1} n_q^{N_i+1} e^{-\beta U_j + \beta \mu_j (N_i+1)} \times \frac{d\mathbf{r}^{N_i+1}}{\Xi(V,T,\mu)}}{\frac{dr}{V} n_q^{N_i} e^{-\beta U_i + \beta \mu_i N_i} \times \frac{d\mathbf{r}^{N_i}}{\Xi(V,T,\mu)}} \\
&= \frac{V}{(N_i+1)} n_q e^{-\beta(U_j - U_i)} e^{\beta \mu}
\end{aligned} \tag{27}
$$

Hence

$$
p^{acc}(i \to j) = min[1, \frac{V}{(N_i+1)} n_q e^{-\beta(\Delta U)} e^{\beta \mu}] \quad \text{for insertion} \tag{28}
$$

Similarly, we obtained the acceptance probability for deletion move as

$$
p^{acc}(j \to i) = min[1, \frac{N_i}{V n_q} e^{-\beta(\Delta U)} e^{-\beta \mu}] \quad \text{for deletion} \tag{29}
$$

To simplify the equation and reduce the quantum density term $n_q$, we define a relative chemical potential

$$
\mu' = \mu + \frac{1}{\beta} \times \ln(n_q) \tag{30}
$$

Finally, suppose the system have $N$ particles on the grid site, we have derived a nice metropolis form expression for the movement acceptance probability $p^{acc}$. Such expression given as the following:

$$\begin{aligned} p_{\text{ins}}^{acc} &= min[1, \tfrac{V}{N+1} e^{-\beta\Delta U + \beta\mu'}] \quad \text{for insertions} \\ p_{\text{del}}^{acc} &= min[1, \tfrac{N}{V} e^{-\beta\Delta U - \beta\mu'}] \quad \text{for deletions} \end{aligned} \tag{31}$$

To further simplify the above expression and make it even easier for the computer to handle, we introduce the Thermal activity factor z, which is defined as

$$z = e^{\beta\mu'} \tag{32}$$

so that equation (23) could be rewrite as

$$\begin{aligned} p_{\text{ins}}^{acc} &= min[1, \tfrac{Vz}{N+1} e^{-\beta\Delta U}] \quad \text{for insertions} \\ p_{\text{del}}^{acc} &= min[1, \tfrac{N}{zV} e^{-\beta\Delta U}] \quad \text{for deletions} \end{aligned} \tag{33}$$

which will serve as the movement acceptance law to be followed on step 3 of each Grand canonical monte carlo iteration.

## 3.5 Biased Monte carlo method

Inspired by Vink's work on biased sampling[6], and his Lecture notes[7]., we here briefly introduce this advanced Monte carlo method. As is mentioned section 2, the probability distribution is proportional to $e^{-\beta E_i}$, denoted by:

$$p_i^u \propto e^{-\beta E_i} \tag{34}$$

where "u" signifies that we are in the "unweighted" ensemble. The general idea of biased sampling is to introduce a **weighting function** to the microstates in the exponential:

$$p_i^w \propto e^{-\beta E_i + W(\zeta)_i} \tag{35}$$

Combine Equation 34 and Equation 35

$$\frac{p_i^u}{p_i^w} \propto e^{-W(\zeta)_i} \tag{36}$$

[6]reference[12]
[7]reference[14]

The weighting function $W(\zeta)_i$, where $\zeta$ stands for the bias variable, modifies the distribution thus that some configuration have higher or lower probabilities would be expected. For example in reference [12], they picked both numbers of particles N and the order parameter S to be as the bias variable $\zeta$ and worked on two biased approaches: 1. biased on N and 2. biased on S. With the modification of weight function, a simulation with a modified potential $E'(\zeta) = E + k_B T W(\zeta)$ would yields a uniform probability distribution in the bias variable.

Here we pick the bias variable $\zeta$ to be the number of particles N and perform a GCMC simulation on the weighted ensemble as an example. Without loss of generosity, assume $N_j = N_i + 1$ such that we have insertion move: $i \to j$ and deletion move: $j \to i$. for symmetric moves, the acceptance rule for insertion move would then follow:

$$\frac{p^{acc}(i \to j)}{p^{acc}(j \to i)} = \frac{p_j^w}{p_i^w} = \frac{V}{N+1} e^{-\beta \Delta U + \beta \mu' + \Delta W} \tag{37}$$

With the Metropolis form:

$$p^{acc}(i \to j) = min[1, \frac{V}{N+1} e^{-\beta \Delta U + \beta \mu' + \Delta W}] \tag{38}$$

Similarly one could obtain the acceptance rule for deletion. We then summarize the acceptance rule as the following:

$$\begin{aligned} P_{\text{ins}}^{acc} &= min[1, \tfrac{V}{N+1} e^{-\beta \Delta U + \beta \mu' + W(N+1) - W(N)}] \quad \text{for insertions} \\ P_{\text{del}}^{acc} &= min[1, \tfrac{N}{V} e^{-\beta \Delta U - \beta \mu' + W(N-1) - W(N)}] \quad \text{for deletions} \end{aligned} \tag{39}$$

Setting the thermal activity factor $z_1 = 1$ and perform a GCMC simulation together with some Biased sampling techniques (for example Wang landau sampling, which will be introduced in next section) and followed by the metropolis rules(Equation 39), one could obtained the weighting function.

Once the weighting function is found, the distribution in the number of particles p(N) is trivially obtained via $P(N|z_1 = 1) = Ce^{W(N)}$ , where C is a normalization constant. One could then extrapolate the probability distribution of the biased parameter for all different thermodynamic activity factor z by using:

$$P(N|\mu_2) = P(N|\mu_1) e^{\beta(\mu_1 - \mu_2)N} \tag{40}$$

as $z = e^{\beta\mu'}$ and $P(N|z_1 = 1) = Ce^{W(N)}$ we have

$$P(N|z_2) = P(N|z_1 = 1)(\frac{z_2}{z_1})^N = C'e^{W(N)}z_2^N \tag{41}$$

with $P(N|z_i)$ the probability distribution P(N) at activity $z_i$ and C', a normalization constant. This means with this biased method, we only run for one simulation to obtain the weighting function for the target system, and then we could capture all the probability distributions for the biased parameter (more specifically, N here). This will significantly improve the time efficiency. To obtain the weighting function, *Wang-Laudau sampling* is the state-of-art ways to do so.

### 3.5.1 Wang-Landau sampling

The weight function can be adjusted "on the fly" and one the most well-known algorithm in this class is the so-called Wang-Landau sampling algorithm[8].
In this approach, we find the optimal W(N) that gives a flat histogram by modifying this weighting function at every MC step ("on the fly") such that enforces a flat distribution and allows it to converge to its optimal value. After every MC move, we update our running estimate for W(N) via:

$$W(N) \leftarrow W(N) + g \tag{42}$$

where g is the **modification factor** that we choose to gauge the rate as the weighting function updated. In our simulation, g begin with an initial value 1: g = 1; Then we form the Wang-Laudau sampling by the following procedure:

1. initially set W(N) = 0 and g = 1.
2. Perform a simulation and updating W(N) at every simulation step using Equation 42. Start a new histogram of N and collect observations throughout the run.
3. repeat the simulation and updating W(N) until the histogram is "flat enough" that: when the number of counts in the least visited histogram bin is no less than 80% of the average number of counts over all bins.
4. When the histogram is "flat enough", scale down the g value by: $g \leftarrow 0.5 \times g$
5. clear the histogram counts (re-zero it) and return to step 2. Repeat until g is less than $10^{-7}$

And after finishing the Monte carlo loop for Wang Landau sampling, one would obtained the weighting function W(N).

[8][13] D.P.Landau, 2001, and [10] Lecture notes from CHE210D, 2012

### 3.6 End note

In this section, we first introduced the general idea of Monte carlo simulation and established the Metropolis algorithm through Markov process. Then by using this Metropolis algorithm , we have showed 2 basic Monte carlo methods: CMC and GCMC and one advanced approach: the biased monte carlo method. In the next section, we will apply the 2 basic method to develop the Monte carlo 'vitual lab' to simulate the lattice gas system.

## 4 Simulate Hard Lattice Gas in a 2-D Grid

In Lattice gas model, each particle is uniquely fixed by a lattice point and for each cell at the lattice, at maximum only one particle can be placed into.(hard core exclusion) Such model is depicted in Figure 4. Assuming First nearest neighbor interaction (1NN) for our lattice model so

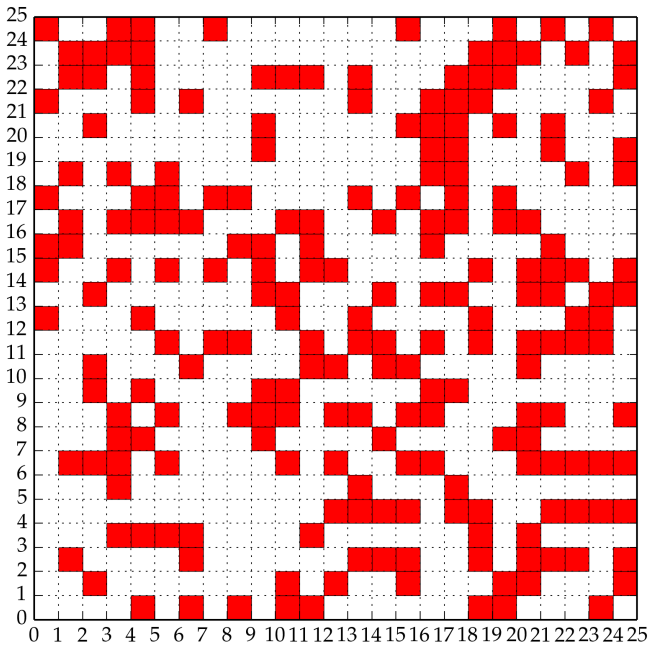



Figure 4: Hard Core Lattice Gas: Maximum one particle per cell

that whenever there is a pair of particles siting on the neighbor site, the internal energy will be lowered by $\epsilon$. Vink's[9] lecture note provides a nice schematic picture (see Figure 5 ). To describe

[9]Reference[14]

such interactions and one could write such interaction as the following:

$$\Delta PE_{ij} = \begin{cases} \infty & \text{if i and j particles overlap} \\ 0 & \text{if i and j particles are not first nearest neighbor} \\ -\epsilon & \text{if i and j particles are first nearest neighbor} \end{cases} \tag{43}$$

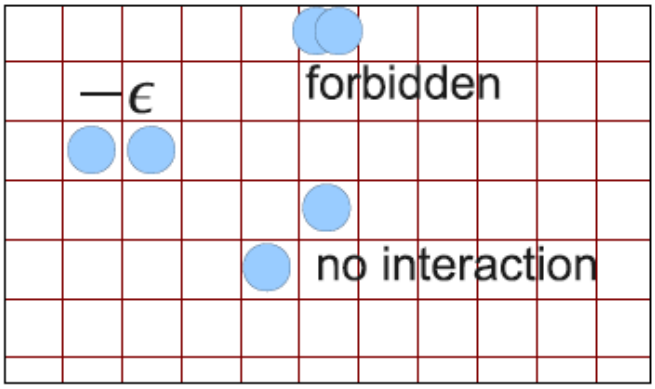



Figure 5: Interaction between particles

## 4.1 Canonical Monte Carlo simulation

### 4.1.1 Simulate Lattice gas with CMC method

The first simple (maybe the simplest) application on Canonical Monte carlo simulation would be simulating the average internal energy $\langle E \rangle$ (Potential Energy here since we do not consider kinetics) and heat capacity for lattice gas. In class MSE 575, we have learned that the heat capacity $C_V$ could been measured in two different ways. The first method is to verify the temperature while measuring the average energy $\langle E \rangle$ and then calculated heat capacity though its definition:

$$C_V = (\frac{\partial \langle E \rangle}{\partial T})_{const\ V} \tag{44}$$

The second way is keep the Temperature as a constant and use the Linear response fluctuation and dissipation (F-D) Theorem to obtained the heat capacity through the spontaneous fluctuation of energy E.

$$C_V = \frac{\Delta E^2}{k_B T^2} = \frac{\langle E^2 \rangle - \langle E \rangle^2}{k_B T^2} \tag{45}$$

Unlike the avarage Energy $< E >$, the variance of energy at constant temperature $\Delta E^2$ is rather difficult or tedious to measure in a lab but fairly simple for a computer to handle through Monte

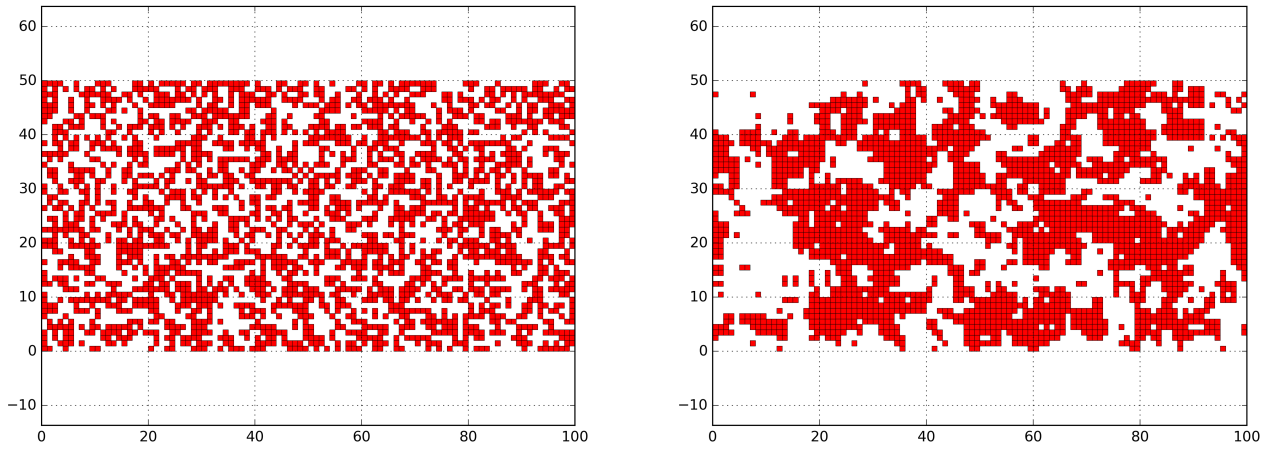
carlo simulations. So instead of actually going to an actually lab, one could also "measure" physical quantities though a 'virtual lab' in front of a computer.

Essentially what Monte carlo simulation do here is to simulate the whole partition function Z(T,V,N) and after millions or billions or even more Monte carlo steps, we could get all the informations through the markov chain, such as configurations (hence entropy), energy and Numbers of particles (useful for grand canonical case since N fluctuated) of each state. As is mentioned in the first section, once the partition function is known, we know all the statistical properties of this system and therefore we will know the variance of energy and heat capacity.

We have also mentioned in class that the whole entire partition function is usually too much of information, and so does it here because there will be millions or billions of states to record and this will be too much even for the smartest computer to-date to handle. However, Since we are only interested in the average energy and the variance of energy this specific example, we only need to record the energy for each equilibrium state.(instead of the configuration, which will clearly takes way much more computer memories to record for a big lattice system) And for this millions or billions of state, we don't need to keep track of every single one of them, but rather we record the sweep average value. For example, for a monte carlo simulation with $10^9$ steps, we divide this $10^9$ events/movements into $10^6$ 'sweeps', like a bundle each contains $10^3$ steps of data, and then only record the average value for each 'sweep' and this $10^6$ sweeps would be sufficient enough to represent the whole partition function.

### 4.1.2 Simulation results

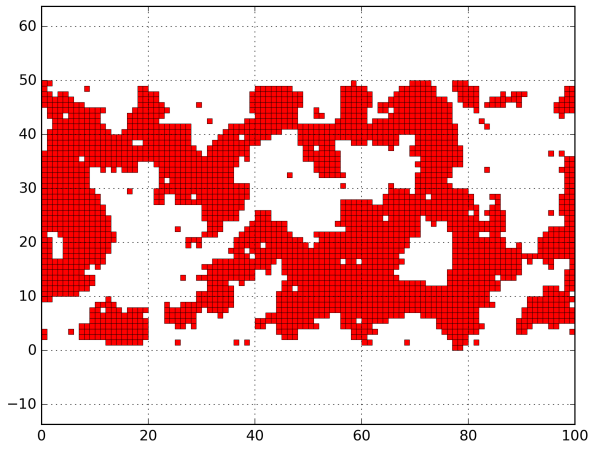
Even though the actual coding is not the purpose of this study, here we still provide a simple CMC simulation result of our own to 'visualize' and show that CMC model could behave just like the real gas: there is gas to liquid phase separation phenomenon when the temperature is low. Here we pick the lattice size to be 50 $\times$100 and setting the packing fraction $\eta = 1/2$, so that $M = 5000$ and $N = 2500$. Over the several sets of CMC simulation, we vary the $\epsilon\beta$ from 0 to 5.5. Since $\beta = \frac{1}{k_B T}$ and $\epsilon$ is a constant, so that basically we were decreasing the temperature and hence a phase transition from gas to liquid are expected to happened as $\epsilon\beta$ raised up. In Figure 6, we attached a 4 'snapshot' to graphically demonstrate this phase transition. The fig a) with lowest $\epsilon\beta$ (highest Temperature) represents gas phase and in fig b) and c), we see that with a decreasing T, the gas particles starts to form clusters due to higher interaction energy (attraction) $\epsilon\beta$ and finally in fig d), all gas particles stick together and form a liquid phase.



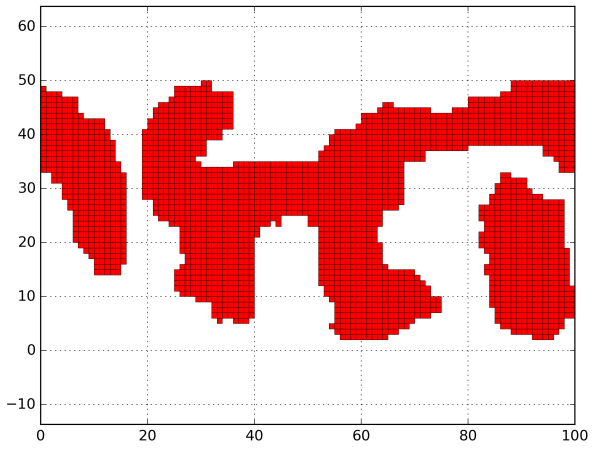
(a) snapshot for $\beta\epsilon = 0.2$



(b) snapshot for $\beta\epsilon = 1.8$



(c) snapshot for $\beta\epsilon = 2.5$



(d) snapshot for $\beta\epsilon = 5.5$

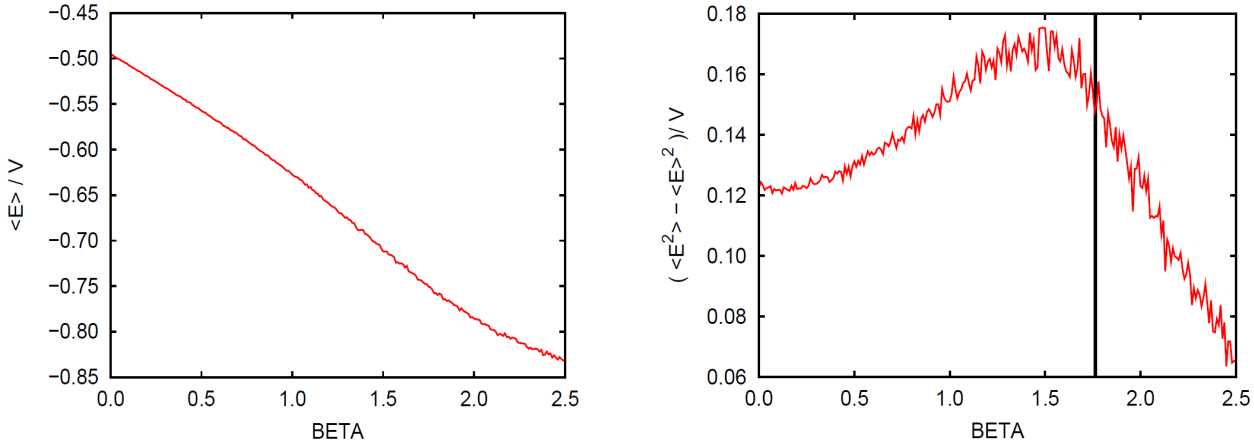
Figure 6: CMC lattice gas simulation snapshot

Moreover, one could obtained the $< E >$ and heat capacity $C_V$ as well and they behaved almost like the real data from the lab. For reference, here we attached vink's result for a Lattice size to be 10×10 and $\eta = 1/2$ in Figure 7. Through Figure 7, we see that the $< E >$ decreases as the Temperature decreases ($\beta$ increases) which is expected and for the heat capacity, since liquid to gas is a first-order transition and due to the discontinuity of entropy S, we expect to see a discontinuity peak from heat capacity as well when the phase transformation took place. Therefore it's reasonable to see such a peak at the heat capacity plot and it peaks at value very close to the exact result.

$$\beta_{exact} = 2\ln(1+\sqrt{2}) \approx 1.763 \tag{46}$$

which is the vertical line marked on Vink's $C_V$ plot. The reason for such small shift is because of the finite size effect[10] since $10 \times 10$ is a really small size. The peak would be getting closer to the

[10]Reference [1] section 3




(a) $< E >$ vs beta (b) heat capacity vs beta

Figure 7: CMC lattice gas $< E >$ and heat capacity

actual position if we increase the lattice size.

## 4.2 Grand Canonical Monte Carlo simulation

### 4.2.1 Chemical potential and Thermal activity

Unlike canonical ensemble fixes T,V,N, grand canonical ensemble have constant Temperature T, volume V and chemical potential $\mu'$. Since chemical potential is not as straight forward as T and V, we first establish an expression of chemical potential as a function of Number of particles N. To find the such expression $\mu'(N, M)$, we start with define the total number of configurations $N_c$ :

$$N_c = \binom{M}{N} = \frac{M!}{N!(M-N)!} \tag{47}$$

where $N_c$ is the total number of configurations, $M$ is the amount of lattice sites in the lattice grid and $N$ is The number of particles that sit in the lattice.

As we learned from MSE 575, the free energy for the system, denoted by F, could be written as the following:

$$F = -k_B T \ln(Z) = -k_B T \ln(\sum_{i=1}^{N_c} e^{-E_i/k_B T}) \tag{48}$$

where $N_c$ is the total number of configurations, $E_i$ is the Energy of ith particle that sits in the cell, $k_B$ is the Boltzmann constant, T is temperature.

Here we assume our Hard Lattice Gas do not interact with each other, therefore $E_{\mathrm{i}} = 0$. Then we can rewrite F as:

$$F = -k_B T \ln(\sum_{i=1}^{N_{\mathrm{c}}} 1) = -k_B T(\ln M! - \ln N! - \ln(M-N)!)$$
$$F = k_B T(N \ln N/M + (M-N)\ln(1 - N/M))$$

Then we obtained the relative chemical potential $\mu'$ by taking the partial derivative of the Free energy $F$ with respect to the number of particles $N$:

$$\mu' = \frac{\partial F}{\partial N} = k_B T(\ln \frac{N}{M} - \ln(1 - \frac{N}{M})) \tag{49}$$

In the previous section, we define the thermal activity $z \equiv e^{\beta\mu'}$, where $\beta \equiv 1/k_B T$, thus that we can rearrange Equation 49 to obtain Equation 50 and Equation 52 as to relate the activity z to packing fraction $\eta$ and we will make later comparison in next section between the simulated value and the theoretical value of $\eta$ (Equation 52) :

$$\beta\mu' = \beta\frac{\partial F}{\partial N} = \ln \tfrac{N}{M} - \ln(1 - \tfrac{N}{M}) \tag{50}$$
$$z = e^{\beta\mu'} \tag{51}$$
$$\eta = \tfrac{N}{M} \; and \; thus \; z = \tfrac{\eta}{1-\eta} \tag{52}$$

### 4.2.2 Simulate packing fraction $\eta$ (Number density) and isothermal compressibility

Utilizing the GCMC method, we implemented a tiny program to check the packing fraction $\eta$. To simplify the problem, here we assume no interacting between particles thus $\epsilon = 0$. After the GCMC simulations, not only the packing fraction $\eta$ is known, because GCMC indeed could give about $10^{10}$ sets (or even more if you are willing to wait) of equilibrium configurations datas (N, configuration, and the corresponding E), thus that the average packing fraction $\overline{\eta} = \frac{<N>}{M}$, $< N^2 >$ and $< N >^2$ are also known as well. Therefore one could also calculate the compressibility through

$$\chi = \frac{(< N^2 > - < N >^2)}{M} \tag{53}$$

which is once again, an application of the Fluctuation and dissipation theory and one could tell that it is very much like the formula for heat capacity in previous section. Due to the lack of time and as coding is not as important than to deliver the general idea for Monte carlo simulation in this study, here we only show the validity of simulating the $\langle N \rangle$ through comparing the simulated result of packing fraction $\eta$ to the theoretical result of $\eta$ (Equation 52).

According to Equation 33 (the movement acceptance probability):

$$p_{\text{ins}}^{acc} = min[1, \frac{Vz}{N+1}e^{-\beta\Delta U}] \quad \text{for insertions}$$
$$p_{\text{del}}^{acc} = min[1, \frac{N}{zV}e^{-\beta\Delta U}] \quad \text{for deletions}$$

Under the assumption of hard core interaction, we have

$$\Delta U = \begin{cases} \infty & \text{if two particles occupy the same lattice site} \\ 0 & \text{otherwise} \end{cases} \tag{54}$$

Now set the lattice size to be L, one could obtain the movement acceptance rule as the following:

$$p_{\text{ins}}^{acc} = min[1, \frac{z\times L\times L}{N+1}e^{-\beta\Delta U}] \quad \text{for insertions}$$
$$p_{\text{del}}^{acc} = min[1, \frac{N}{z\times L\times L}e^{-\beta\Delta U}] \quad \text{for deletions} \tag{55}$$

Utilizing the GCMC method, for each z value, perform a Monte Carlo simulation over $10^9$ Monte Carlo steps, and for each step:

- pick a random position from the 2-D lattice and them attempt to Add or Del particles with equal probability at that position;
- if it is the case of Addition, then attempt to Add the particle into the chosen lattice site following the movement acceptance rule;
- if it is the case of Deletion, then attempt to Delete the particle from the chosen lattice site following the movement acceptance rule;

As to test this model, we pick a specific lattice size of $100 \times 100$ and we then perform a set of Monte Carlo simulation to observe the relationship between $\eta$ and z. According to Equation 50, as M is fixed($100 \times 100$ ), the change of the number of particles: N is solely depend on $\beta\mu'$. Hence inplies that the change of the packing fraction $\eta$ is solely depend on z. We fluctuate our z value by slowly uniformly increasing it from 0 to 10 over 500 steps and for each simulation, we ran for $10^9$ MC steps. From each MC simulations (the $10^9$ MC steps) with a fixed z value, we record the average numbers of particles N in the system as to calculate for the corresponding $\eta = N/M$ under the fixed z value. At the same time, the value of $\eta$ could also be obtained by $\eta = z/1 + z$. Thus we plot z versus $\eta$ as to make comparison between the theoretical result and simulated result. The result of this comparison is depicted in Figure 8, the red line stands for the simulated data and the green line represents the theoretical result. As we see from Figure 8, the simulated result is consistent with the theoretical result and the simulated data is fluctuating within a reasonable range.

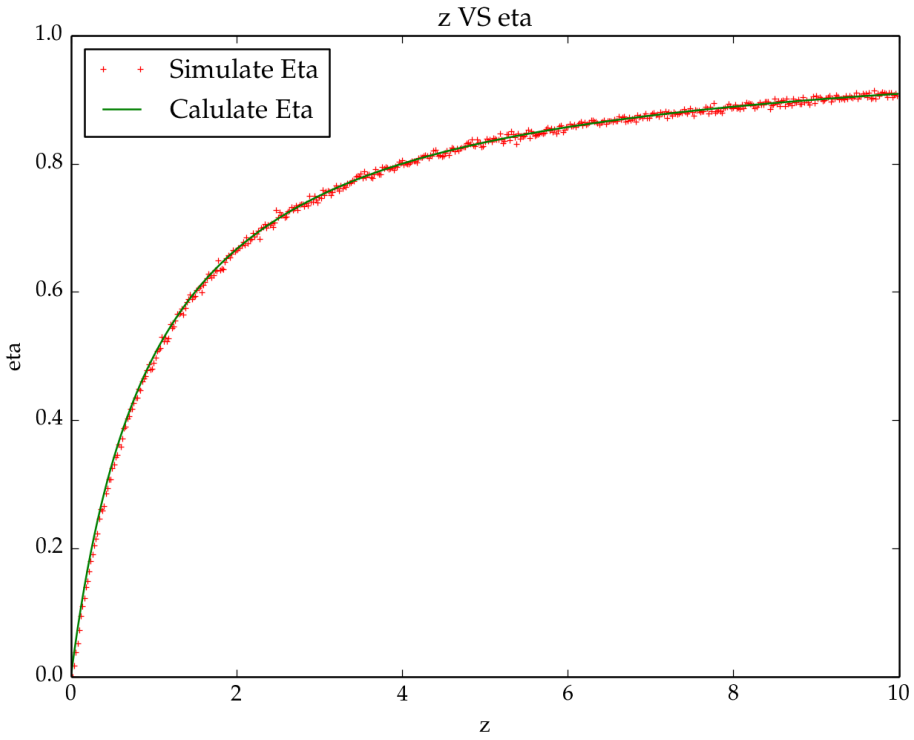


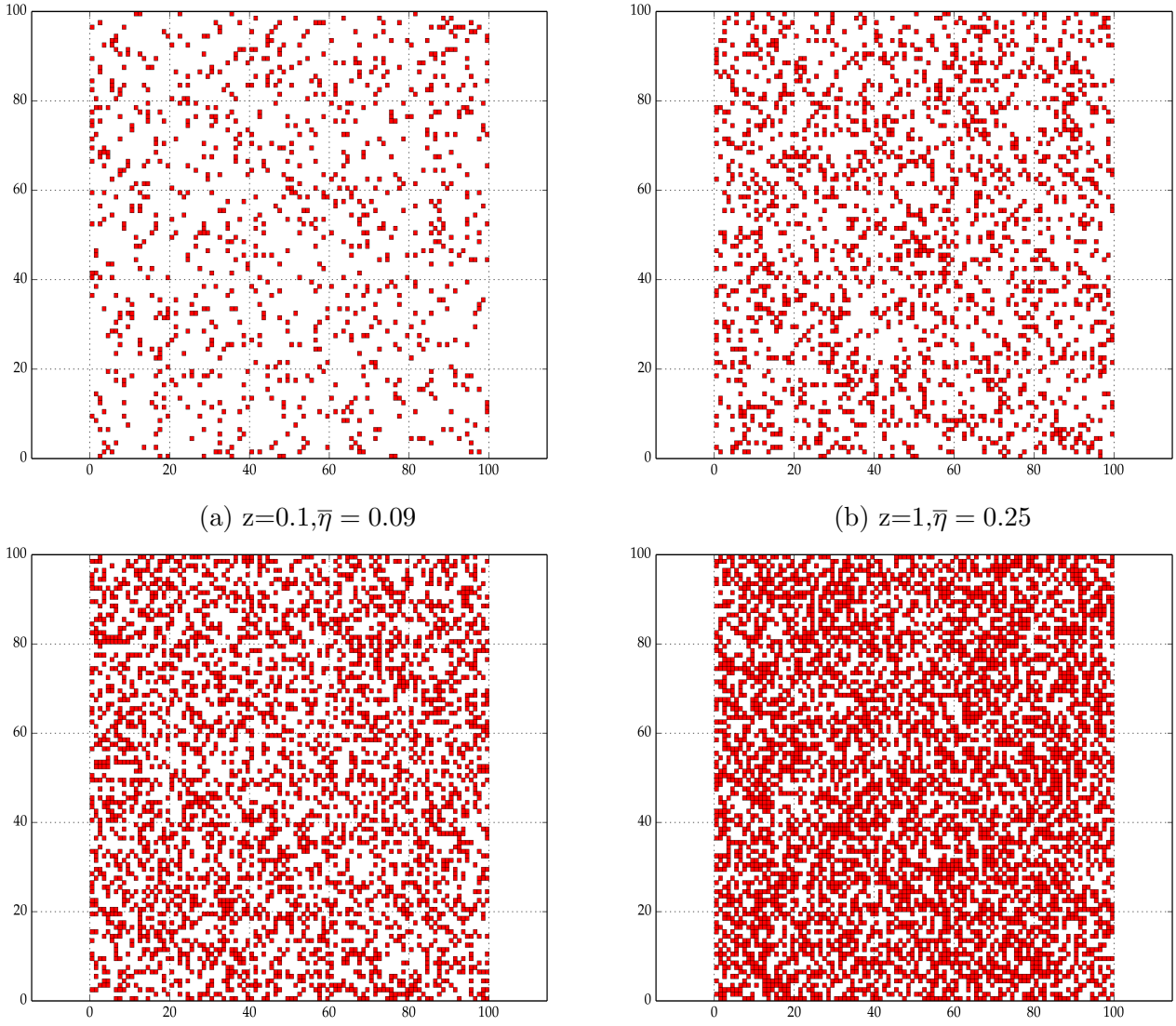

Figure 8: z VS $\eta$

### 4.2.3 Configuration Snapshots

The snapshot of the final configuration of the GCMC simulation on lattice gas with z = 0.1, 0.25, 0.5 and 1 is depicted in Figure 9. We divided the $10^9$ Monte Carlo steps into 10000 sweeps, record the total number of particles at the end of each sweep and obtained the corresponding packing fraction(density) $\eta \equiv \frac{N}{L \times L}$. Then we calculate the average packing fraction(density) $\overline{\eta}$ under each z value by taking the average of the packing fraction $\eta$s over all the 10000 sweeps. As shown in Figure 4, we see that the the packing fraction $\overline{\eta}$ increases as z increases and they are not linearly correlated as expected and since we did not involve interaction energy in this GCMC demo there is no clustered formed no matter how strong the activity z are which is also as expected.

## 4.3 End note

In this section, we have applied the Monte carlo method onto both close and open lattice gas system and they both yield a pretty good result (at least qualitative here) compared to the real physical systems. we have also briefly introduced a few application of fluctuation and dissipation theory such as simulating heat capacity $C_V$ and compressibility $\chi$ and Professor Vink's simulated result for $C_V$ shows a nice agreement with the actual data and hence we see the validity of F-D theorem on Monte Carlo approach.




(a) z=0.1,$\overline{\eta} = 0.09$

(b) z=1,$\overline{\eta} = 0.25$

(c) z=0.1,$\overline{\eta} = 0.50$

(d) z=1,$\overline{\eta} = 1$

Figure 9: Snapshot of the final configuration of Lattice gas model

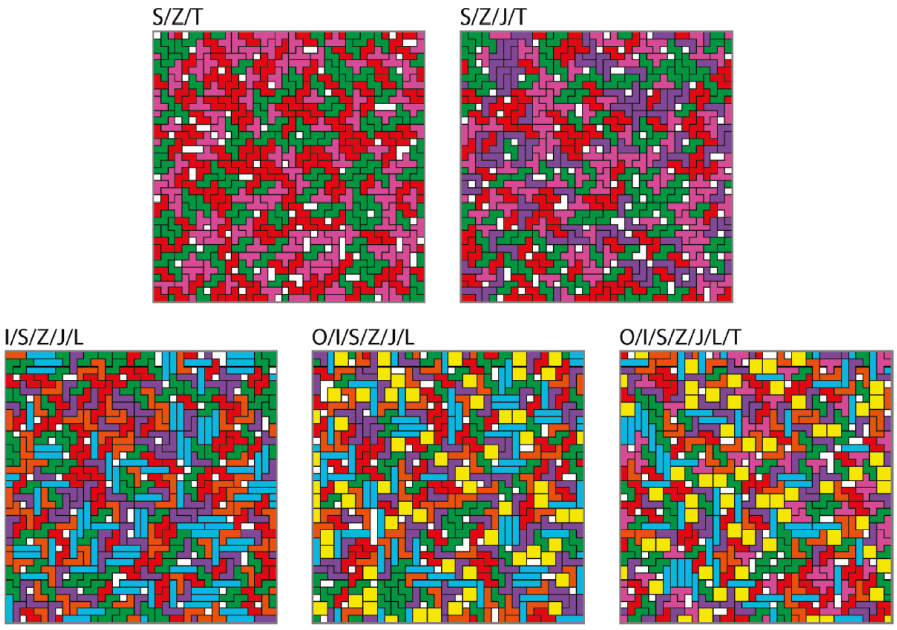



Figure 10: Tetromino Fluids

# 5 Outlooks

After finishing this introductory level work about mathod carlo method on lattice gas model, we are now ready to look at some more advanced Monte carlo techniques and be able to study journal articles in such field. Here I would like especially to point out two very good articles that are definitely worth studying next. In both works, their Monte Carlo simulations are performed in the grand canonical ensemble (the method we've introduced in section 3.2) to simulate molecules or nanoparticles reversibly adsorb to a surface or interface from a bulk reservoir. The first one is Reference [12], a work about simulation of the on lattice rods system through the biased sampling method, which we have briefly mentioned at the end of section 3. They employed a very modern and cool techniques called the "successive umbrella sampling" (rather than the Wang Landau method) and their results agrees with theoretical predictions. The other work we found very cool is a study about the Tetromino Fluids from Reference [11], that instead of the boring lattice gas particles, they took a research on the Tetris-like fluids. (I took a picture from their work just to show how cool this model looks like, see Figure 10).

All in all, through writing this term paper, although there are a lot of materials we did not covered as planned (such as the study of Ising model and the full implementation of coding the Monte carlo simulation of lattice gas model), we have learned a lot about the basics of Monte carlo simulations. Overall, we found that the field of Monte carlo simulations is fascinating and we are eager to achieve more in our future study.

# References


[1] K. Binder, *Journal of Computational Physics* 59, 1-55, 1985.

[2] Selinger.J, *Introduction to the Theory of Soft Matter From Ideal gas to Liquid crystals*, 2016, ISBN:978-3-319-21053-7

[3] K. Binder, *Thin Solid Films*, 20 (1974) 367-381

[4] K. Binder, *Rep. Prog. Phys.* 60, (1997) 487–559.

[5] K. Binder, *Eur. Phys. J.B.* 64, (2008) 307–314

[6] D.P.Laudau, K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, ISBN: 978-1107074026

[7] D.P.Laudau, I.M. KIM, *Surface Science* 110 (1981) 415-422

[8] D. Frenkel, K. Binder, H. Grubmuller,K. Kremer, *Computational soft matter: From Synthetic Polymers to Proteins*, NIC series, vol,23, ISBN 3-00-012641, pp.29-60, 2004

[9] D. Frenkel, B. Smit, *Understanding Molecular Simulation: From algorithms to Applications*, ISBN: 978-0-12-267351-1

[10] M.Scott Shell, UCSB CHE210D course materials: https://engineering.ucsb.edu/ shell/che210d/

[11] B. C. Barnes, D. W. Siderius, and L.D. Gelb, *Structure, Thermodynamics, and Solubility in Tetromino Fluids*, Langmuir 2009, 25(12), 6702–6716

[12] R.L.C. Vink, S. Wolfsheimer, and T. Schilling, The Journal of Chemical Physics 123, 074901 (2005)

[13] D. P. Landau, Shanho Tsai, M.Exler, A new approach to Monte Carlo simulation in statistical physics: Wang Landau sampling, Am. J. Phys. 72(10) October 2004

[14] R.L.C. Vink. Monte Carlo methods in Statistical Physics lecture notes, 2011.