Monte Carlo Molecular Simulation with Isobaric-Isothermal and Gibbs-NPT Ensembles

Thesis by

Shouhong Du

In Partial Fulfillment of the Requirements
For the Degree of
Master of Science

King Abdullah University of Science and Technology, Thuwal,
Kingdom of Saudi Arabia

May, 2012

EXAMINATION COMMITTEE APPROVALS FORM

The thesis of Shouhong Du is approved by the examination committee.

Committee Chairperson: Shuyu Sun

Committee Member: Ying Wu

Committee Member: Zhiping Lai

ABSTRACT

Monte Carlo Molecular Simulation with Isobaric-Isothermal and Gibbs-NPT Ensembles

Shouhong Du

This thesis presents Monte Carlo methods for simulations of phase behaviors of Lennard-Jones fluids. The isobaric-isothermal (NPT) ensemble and Gibbs-NPT ensemble are introduced in detail. NPT ensemble is employed to determine the phase diagram of pure component. The reduced simulation results are verified by comparison with the equation of state by by Johnson et al. and results with L-J parameters of methane agree considerably with the experiment measurements. We adopt the blocking method for variance estimation and error analysis of the simulation results. The relationship between variance and number of Monte Carlo cycles, error propagation and Random Number Generator performance are also investigated. We review the Gibbs-NPT ensemble employed for phase equilibrium of binary mixture. The phase equilibrium is achieved by performing three types of trial move: particle displacement, volume rearrangement and particle transfer. The simulation models and the simulation details are introduced. The simulation results of phase coexistence for methane and ethane are reported with comparison of the experimental data. Good agreement is found for a wide range of pressures. The contribution of this thesis work lies in the study of the error analysis with respect to the Monte Carlo cycles and number of particles in some interesting aspects.

ACKNOWLEDGMENTS

I would like to sincerely thank my supervisor Prof. Shuyu Sun for his continuous guidance and encouragement throughout the course of this work and the committee members, Prof. Ying Wu and Prof. Zhiping Lai for their valuable feedback.

I appreciate a lot for the help and suggestions from Dr. Jun Li, Ahmad Kadoura and Zhiwei Ma.

I thank my family for their continuous encouragement and my fiance for the deep moral support at all times.

Lastly, I would like to thank the people at KAUST, Thuwal, Makkah Province, Saudi Arabia for providing support and resources for this research work.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

EoS	Equation of State
L-J	Lennard-Jones Model
LCG	Linear Congruential Generator
MC	Monte Carlo
MD	Molecular Dynamics
NPT	Isobaric-Isothermal Ensemble
NVT	Canonical Ensemble
RNG	Random Number Generator

LIST OF SYMBOLS

o	Old configuration
n	New configuration
$\text{acc}(o \to n)$	Acceptance probability of a move from o to n
$\alpha(o \to n)$	Probability of generating conf. n from o
$\pi(o \to n)$	Transition probability from o to n
$k(o \to n)$	Flow of configuration from o to n
$U(o)$	Potential energy of configuration o
E	Total energy
$H(p, r)$	Hamiltonian
k_B	Boltzmann's constant
β	Reciprocal temperature$(1/k_BT)$
M	Total number of Monte Carlo samples
N	Number of particles
V	Volume
ρ	Number density
ρ_L	Density of liquid phase
ρ_V	Density of vapor phase
P	Pressure
T	Temperature
Q(N,P,T)	Isothermal isobaric partiton function
Π(N,P,T)	Probability density for NPT ensemble
$\Pi(o)$	Probability finding a system in conf. o
μ	Chemical potential
r_c	Cut off radius of the potential
ϵ	Characteristic energy in potential model

σ	Characteristic distance in potential model
Ranf	Random number uniform distributed in [0,1]
vir	Virial
Super *	Reduced units

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter I

Introduction

Molecular simulation is a computational experiment based on a molecular model with the character of both theory and experiment. Typically 10 to 100,000 or more atoms or molecules are simulated. It is widely used in the fields of computational chemistry, drug design, computational biology and materials science. Molecular Dynamics (MD) and Monte Carlo (MC) simulation are two popular methods.

There are two types of MD approaches [1]. One is the classical molecular dynamics, which directly solves systems from Newton's equation. The other is the quantum molecular dynamics, where the system is governed by Schrodinger equation. There are two basic steps in MD method, the first is to compute the forces on each particle, which contributes significantly to the algorithm complexity and consumes most of the computation time. The second step is to solve a system of nonlinear ordinary differential equations with many existing efficient numerical methods. MD is applied today mostly in materials science and the modeling of biomolecules.

Monte Carlo methods [2, 3, 4] are a class of computational algorithms which obtain the results by repeated random sampling and are used extensively in mathematics, economics, physics and engineering. They are useful to study the nondeterministic processes, deterministic problems which are too complex to solve analytically, and systems with high dimensionality. In molecular simulation, Monte Carlo uses importance sampling to compute the equilibrium properties of N-body systems.

The basic Monte Carlo process goes as following: starting from an initial configuration of the system, we perform a Monte Carlo trial move. We accept or reject this trial move based on an acceptance rule that ensures the configurations are sampled in the simulation from a statistical mechanics ensemble distribution. After many such cycles, we estimate the expected value of quantities of interest by average values. In Monte Carlo simulations, one important and necessary condition must be satisfied, which is the scheme employed must be ergodic. Ergodic means that we can access every point in the configuration space by performing a finite number of Monte Carlo moves from any other point.

The main differences between MD and MC are listed as follows.

- MD solves equations of motion while MC obtains result by ensemble average.
- MD is deterministic while MC is stochastic.
- MD retains time element while there is no time element in MC.

The reason to choose Monte Carlo in some cases is that it could allow us to perform unphysical trial moves, but essential for the equilibration of system. What is more, the force evaluation is not necessary in MC, but not trivial to compute in MD. Monte Carlo is mainly used for small molecules, because for chain molecules, the acceptance probability is quite small. For chain molecules, other modified MC methods like Configurational Biased Monte Carlo are employed.

There are a lot of books, papers and review articles in the area of molecular simulation. Among them, the book by Frenkel and Smit [5] is outstanding. It provides an excellent introduction to Monte Carlo and Molecular Dynamics methods and covers many major recent methods and techniques in detail.

In this thesis, we studied the phase diagram for one component system and phase equilibrium for a binary mixture system. The material we focused on are methane and ethane. They are the most important hydrocarbons in oil and gas industry, and also for environment issues. The investigation of their thermodynamic properties and phase behaviors is very important. Beyond molecular simulation method, there are two other approaches to study the phase behavior: equation of state (EoS) models [6, 7] and empirical correlations based on experimental data. Over the years, a large amount of experimental data has been collected and many theoretically or empirically based models are developed. However, the experimental measurements are expensive and time consuming. The correlated empirical and theoretical models have limited predictive capabilities either confined by the range of experimental data or where they differ from the experimental data. In this sense, the molecular based simulation methodology is of much interest and importance to investigate.

The thesis is organized as follows. Monte Carlo method and technical details in molecular simulations are thoroughly reviewed in chapter II. In chapter III, two ensembles are presented: NPT ensemble for phase diagram of single component one phase system and Gibbs-NPT ensemble for phase equilibrium of multi components. Blocking method is investigated in chapter IV for variance calculation and error analysis in our simulation results. Chapter V is devoted to the application of simulations of the phase behaviors and phase equilibrium. The conclusion with some discussions as well as the outlook of future work are presented in Chapter IV. The contribution of this thesis are the error analysis with respect to the Monte Carlo cycles, error propagation and Random Number Generator performance comparison besides the phase behavior simulation.

Chapter II

Monte Carlo Molecular Simulation

In this chapter, we will give a detailed description of MC method. The reader may refer to Appendix A for the relevant physical concepts of statistical thermodynamics and deviation for the partition function. We now describe below the Monte Carlo method [5].

II.1 Metropolis Method

In most cases, the goal of a Monte Carlo molecular simulation is to calculate the equilibrium thermodynamics and physical properties of the system, like pressure, volume, density and composition. Let us consider the canonical ensemble with N identical atoms, where the temperature and volume are kept constant. The reason to take the canonical ensemble as example is that it is the most easy one to analyze involving only one trial move. Suppose we need to calculate the thermal average $\langle A\rangle$ of some quantity A. This could be done by evaluating its ensemble average, which is given by equation (A.4) in Appendix A

$$\langle A\rangle = \frac{\int d\mathbf{r}^N \exp\left[-\beta U(\mathbf{r}^N)\right] A(\mathbf{r}^N)}{\int d\mathbf{r}^N \exp\left[-\beta U(\mathbf{r}^N)\right]}, \tag{II.1}$$

where $\mathbf{r}^N$ denotes the coordinates of all N particles, U is the potential energy of the system, $\beta = 1/k_B T$, k_B is the Boltzmann constant and $\exp[-\beta U]$ is know as the Boltzmann factor.

Hence, we wish to know the ratio of two integrals as equation (II.1) illustrates. It is not hard to observe that the weight Boltzmann factor $\exp[-\beta U]$ before $A(\mathbf{r}^N)$ will determine the contribution of A to the integral, the larger the weight is, the more contribution will be made. In order to evaluate the integral ratio efficiently without wasting considerable effort, the importance sampling technique is adopted. The basic idea behind importance sampling is to increase the sample density in the region where the Boltzmann factor is large and sample few points elsewhere and hence maximize the sampling efficiency.

Therefore, the probability density of finding system in configuration $\mathbf{r}^N$ is:

$$\Pi(\mathbf{r}^N) = \frac{\exp\left[-\beta U(\mathbf{r}^N)\right]}{\int d\mathbf{r}^N \exp\left[-\beta U(\mathbf{r}^N)\right]}. \tag{II.2}$$

where the denominator is the partition function, serving as a normalizing constant guaranteeing the probabilities of all possible configurations sum up to one.

The basic simulation procedure of Monte Carlo method goes as following: starting from configuration o (old), a new trial configuration n (new) is generated by adding some random perturbation to o. There exist many methods to construct the Markov chain making the average probability of finding the system in configuration n is proportional to $\Pi(n)$, so that the configurations are distributed according to the desired probability distribution (II.2). Here, we demonstrates a simple and generally applicable scheme: Metropolis scheme [8].

Once the system reaches equilibrium after a very large number of Monte Carlo cycles, to maintain the equilibrium state, we do no want the subsequent Monte Carlo moves destroy this equilibrium distribution. This means in equilibrium, the number of accepted trial moves leaving state o must be equal to the the number of accepted trial moves from all the other states to state o, which is called the balance condition. In practice, it is convenient to impose a much stricter condition, the so called detailed balance: the number of accepted trial moves from state o to state n must be equal to the the number of reverse moves. The detailed balance condition is sufficient, but not necessary for the validity of a Monte Carlo scheme. However, there is no example in the literature where a balance only algorithm can be shown to be much faster than the detailed balance algorithm. Hence, we impose the detailed balance condition for the Monte Carlo algorithm for safety and efficiency.

The detailed balance condition can be formulated as

$$k(o \rightarrow n) = k(n \rightarrow o), \tag{II.3}$$

where $k(o \rightarrow n)$ is the flow of configuration from o to n, which is equal to the product of the probability of being in configuration o, $\Pi(o)$, and the transition probability from configuration o to n: $\pi(o \rightarrow n)$. This detailed balance can be expanded as:

$$\Pi(o) \times \pi(o \rightarrow n) = \Pi(n) \times \pi(n \rightarrow o). \tag{II.4}$$

Many possible choices satisfying the above equation are available for the transition matrix $\pi(o \rightarrow n)$. Since in a Monte Carlo move, first of all, we need to perform a trial move from o to n. We denote the probability of generating configuration n from o by $\alpha(o \rightarrow n)$. The second step is to determine whether we should accept or reject this trial move, and denote the acceptance probability by $\mathrm{acc}(o \rightarrow n)$. This results in

$$\pi(o \rightarrow n) = \alpha(o \rightarrow n) \times acc(o \rightarrow n). \tag{II.5}$$

α is chosen to be symmetric, i.e. $\alpha(o \rightarrow n) = \alpha(n \rightarrow o)$ in many cases but not necessary for all moves. As a result, equation (II.4) can be rewritten as

$$\Pi(o) \times acc(o \rightarrow n) = \Pi(n) \times acc(n \rightarrow o). \tag{II.6}$$

which , together with equation (II.2), leads to

$$\frac{acc(o \to n)}{acc(n \to o)} = \frac{\Pi(n)}{\Pi(o)} = \exp\{-\beta[U(n) - U(o)]\}. \tag{II.7}$$

There are again many possible choices for acc$(n \to o)$ satisfying the above condition. Metropolis proposed the following scheme: if the new configuration n has lower energy than the old configuration n, we then accept the move. Otherwise, accept the move with acc$(n \to o) = \exp(-\beta[U(o) - U(n)]) < 1$. Therefore, the Metropolis acceptance criterion [9] can be summarize as

$$acc(o \to n) = \min\{1, \exp(-\beta[U(n) - U(o)])\}. \tag{II.8}$$

Equation (II.8) could be also interpreted in the sense of energy minimization. Since in the canonical ensemble, there is only one trial move: the particle displacement to equilibrate the temperature. The system has tendency to be more stable in the low energy, where the energy function is generally non convex, implying there are many local minimums for the energy function. In order to walk to the global minimum, after one Monte Carlo move, if the new energy is lower than the current energy, we accept the new configuration to decrease the energy. On the other hand, if the new energy is higher than the current energy, in order to jump out the potential local minimum, it is still possible to accept the new configuration even if the energy increase. The characteristic of non convexity for most real problems and the essence of the Metropolis acceptance makes Monte Carlo a very effective tool in optimization.

II.2 Basic Algorithm Structure

In conclusion, the major steps of numerical algorithm for Monte Carlo methods described in the previous section include:

1. Initialize the configuration.
2. For each cycle
   (a) Generate a new configuration n by making a perturbation to the present configuration o.
   (b) Accept the new configuration based on acceptance rule. The new configuration n is accepted if Ranf (random number uniformly distributed in $[0, 1]$) is less than the acceptance probability $\text{acc}(o \to n)$.
   (c) If the trial is rejected, the present configuration is taken as the next one in the Markov chain.
3. Terminate the simulation process after getting sufficient samples and accumulate sums for averages.

A great variety of trial moves can be chosen and basic selection of trial moves is determined by the choice of ensemble. Commonly used ones are particle displacement, volume change, particle transfer, molecule rotation, etc. For a specific ensemble, we must decide how to generate trial moves which could guarantee the symmetry of the underlying Markov Chain to ensure the detailed balance condition is satisfied.

II.3 Potential Model

As we could see from the deviation of the Metropolis method, the kinetic energy is not considered in the Monte Carlo simulation, cause what we only care about is the configurational information, which means the positions of particles, not the velocity. To calculate the potential energy of the system, we need to provide a reliable potential model based on the molecules we are dealing with. For the non-polar molecules of methane and ethane we are interested to investigate in this thesis, there is no Columbic energy attached, what we only need to consider is the intermolecular energy.

The simplest choice, employed in many popular force field is the pair potential, where the total energy can be evaluated by summing the potential energy contributed by pairs of particles. An example of such pair potential is the Lennard Jones potential (also referred to as the L-J potential) . We assume that we are dealing with the hard sphere system where all intermolecular interactions are pairwise additive. Ethane is modeled with two Lennard Jones sites and methane could be modeled with one Lennard Jones site.

II.3.1 Lennard-Jones Potential

The Lennard-Jones potential is quite a simple model providing a good approximation for the interaction between a pair of neutral particles. The most common expression is

$$u_{i,j}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right], \tag{II.9}$$

where r is the distance between the particles (or sites) i and j, ϵ is the depth of the potential well, σ is the finite distance at which the inter-particle potential is zero. The Lennard-Jones potential is widely used as a force field in thermodynamic and material simulations because of its computational simplicity and relatively good approximation capability. The parameters ϵ and σ in the model are different for different materials and can be adjusted from the experimental data.

II.3.2 Truncated L-J Potential

The time complexity for the pairwise energy calculation is $O(N^2)$, computation time would be significant when N is large. To save computation time, we often truncate the L-J potential model at a cut-off distance r_c. The simplest method to truncate the potential is to ignore all interactions beyond r_c.

$$u_{i,j}^{\text{trunc}}(r) = \begin{cases} u_{i,j}(r) & r \leq r_c \\ 0 & r > r_c \end{cases} \tag{II.10}$$

There are two important aspects of potential truncation we should consider seriously. The first one is whether the truncation is reasonable. The other is how should we eliminate the effect caused by the truncation.

Assume the system we are coping with is three dimensional, for any given particle i, the average potential energy contributed by interacting with all other particles is

$$\begin{aligned} u_i &= \frac{1}{2}\int_0^\infty dr 4\pi r^2 \rho(r) u(r) \\ &= u_{\text{trunc}} + \frac{1}{2}\int_{r_c}^\infty dr 4\pi r^2 \rho(r) u(r). \end{aligned} \tag{II.11}$$

where $\rho(r)$ is the average number density at distance r from particle i. For simplicity, we assume $\rho(r) = \rho$ for $r > r_c$ in the preceding computation. As we could see from the above equation, the potential energy function $u(r)$ has to decrease more rapidly than r^{-3} to guarantee the tail contribution is finite, which the L-J potential could satisfy. The tail for the L-J potential for correction can be computed as:

$$\begin{aligned} u_{\text{tail}} &= \frac{\rho}{2}\int_{r_c}^\infty dr\ 4\pi r^2 u(r) \\ &= \frac{\rho}{2} 4\pi \int_{r_c}^\infty dr\ r^2 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \\ &= \frac{8}{3}\pi\rho\epsilon\sigma^3 \left[\frac{1}{3}\left(\frac{\sigma}{r_c}\right)^{9} - \left(\frac{\sigma}{r_c}\right)^{3}\right]. \end{aligned} \tag{II.12}$$

II.3.3 Reduced Units

In simulations, it would be convenient for us to express the fundamental quantities like temperature, density, pressure or others in dimensionless units. In the system of a

Lennard-Jones Potential system, a natural choice for the basic units is the following:

- Unit of length, σ
- Unit of energy, ϵ
- Unit of mass, m

In this way, the other quantities such as energy can be expressed in terms of the basic units and relative to the L-J parameters. In terms of the above reduced units, denoted with superscript *,the reduced potential $u^* \equiv u/\epsilon$ is a dimensionless function of reduced distance $r^* \equiv r/\sigma$. The normalized number density $\rho^* \equiv N\sigma^3/V$, pressure $P^* \equiv P\sigma^3/\epsilon$, temperature $T^* \equiv Tk_B/\epsilon$.

The reduced form of the Lennard-Jones Potential is

$$u^*_{i,j}(r^*) = 4\left[\left(\frac{1}{r^*}\right)^{12} - \left(\frac{1}{r^*}\right)^{6}\right]. \tag{II.13}$$

The reduced form of the tail correction is

$$u^*_{\text{tail}} = \frac{8}{3}\pi\rho^*\left[\frac{1}{3}\left(\frac{1}{r^*_c}\right)^{9} - \left(\frac{1}{r^*_c}\right)^{3}\right]. \tag{II.14}$$

There are several benefits of use of reduced units.

- Unitary: Different combinations of ρ, T, ϵ and σ may all correspond to the same state in reduced units. Thus, by running a single simulation, we could obtain results for many different systems.
- Error reduction: Because of the finite digits computing, when very large numbers and very small numbers are computed together, the rounding error might be significant, while in reduced units, the quantities are of order 1.
- Bug detection: if we find a very large or a very small number during simulation, there are chances that error exits in the program.

II.4 Boundary Conditions

Through molecular simulation, we would like to gather information about the properties of a macroscopic sample. However, for the time being, most simulations can only handle a few hundred to a few thousand particles. To eliminate the effect of small size system and surface effect and mimic the infinite bulk surrounding the N particles system, the periodic boundary condition is employed in our simulation to achieve the target in Monte Carlo simulations.

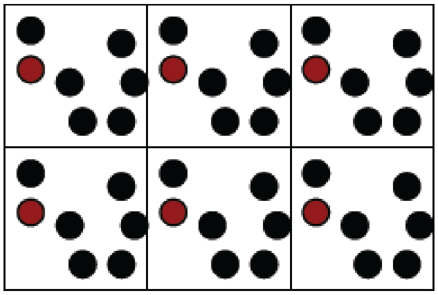

Figure II.1: Periodic boundary conditions

The periodic boundary conditions is demonstrated in Figure II.1. Particles have mirror images in all other boxes. A particle now interacts with all other particles in the infinite system rather than a finite system. If a particle moves a box, then the image in a neighboring box comes in this box.

For periodic boundary conditions, the cut off radius r_c is taken to be less than $L/2$ (half of the box length). For a given particle, only the interaction with the nearest periodic image of any other particles is taken into account.

II.5 Random Number Generator

Random number generator (RNG) is a key component in Monte Carlo simulation. In reality, there is no perfect random number generator. As a matter of fact, the computer programs for generating random numbers are actually deterministic, yielding the so called pseudo-random sequence.

In stochastic simulations, to verify the simulation results, it is always a good idea to run a Monte Carlo simulation with several different random number generators [10]. The reason is that the intrinsic structure of the RNG might interfere with simulation problem and produce incorrect results.

There are two big families of random number generators, linear ones and nonlinear ones. Nonlinear generators are generally much slower than the linear ones, but with the capability producing larger samples. In Monte Carlo simulations, linear RNGs are the best-known and most widely used ones, they produce linear point structures in every dimension. One of the most common generators is linear congruential generator with the following recurrence expression:

$$X_{n+1} = (aX_n + b) \mod m. \tag{II.15}$$

X_0 is called the seed, the modulus m determines the maximum number of random numbers the expression can generate, a and b are parameters.

TT800 and Mersenne Twister (MT) from Makoto Matsumoto [11] based on matrix linear recurrence can provide fast generation of very high-quality pseudorandom numbers, with a period of 2^{800} for TT800 and a quite long period of $2^{19937} - 1$ for MT.

In this thesis, we apply the intrinsic Fortran random number generator (call Random_Seed, Random_Number), linear congruential generator and TT800 RNG to validate our simulation results.

Chapter III

Monte Carlo Simulations in Two Ensembles

There are several popular ensembles for MC simulations, categorized by different thermodynamic constraints. Canonical ensemble (NVT) [12] refers to all states consistent with fixed temperature, volume and number of particles. Grand canonical ensemble (μVT) [13] refers to the system with fixed volume, temperature and chemical potential, which can be used to compute the average molecule number in adsorption studies. Isobaric-isothermal ensemble (NPT) is constrained by constant pressure, temperature and number of molecules. Gibbs-NVT ensemble accounts for the phase equilibrium with constant temperature, volume and number of particles. Gibbs-NPT ensemble deals with systems under constant temperature, pressure and number of molecules for phase equilibrium. In this chapter, NPT ensemble and Gibbs-NPT ensemble will be introduced.

III.1 Isobaric-Isothermal Ensemble

The isobaric−isothermal ensemble, also called NPT ensemble, is a statistical mechanical ensemble widely used in Monte Carlo simulations, which was first proposed by Wood [14] and generalized to systems with continuous intermolecular forces by McDonald [15]. In an NPT ensemble, the number of particles N, pressure P and temperature T are kept as constant. The NPT ensemble is very important in chemistry since chemical reactions are usually conducted under the constant pressure condition [16, 17]. In addition, constant NPT ensemble is convenient for the systems in the vicinity of first order phase transition, because the system is free to transform completely into the state of lowest Gibbs free energy at constant pressure. This is different from other ensembles. For example, in a constant NVT ensemble, the phase separation may be prevented by the finite size effect.

III.1.1 Statistical Mechanical Basis

Assume the system we are dealing with is composed by N identical particles. The expression of the partition function for constant NPT ensemble can be formulated as the weighted sum of the partition function of canonical ensemble described in the previous chapter as following.

$$Q(N, P, T) = c \int dV V^N \exp(-\beta PV) \int d\mathrm{r}^N \exp[-\beta U(\mathrm{r}^N)]. \tag{III.1}$$

where the partition function Q is now a function of N, P and T. c is a normalizing factor.

Now, the probability of finding the system in configuration with volume V and position r^N is:

$$\begin{aligned}\Pi(V) &\propto V^N \exp(-\beta PV) \exp[-\beta U(\mathrm{r}^N)] \\ &= \exp\{[-\beta[U(\mathrm{r}^N) + PV - N \ln(V)/\beta]\}.\end{aligned} \tag{III.2}$$

III.1.2 Trial Moves

In NPT ensemble, we perform two types of trial moves, particle displacement and volume change.

Particle Displacement

Particle displacement consists of two types of displacements: translational and orientational. We confine ourselves to the trial moves only involving the particle centers of mass, which means the orientation is not within our consideration. In the preceding review, we only investigate how the translational displacement works. An obvious translational trial move is by adding some random numbers between $-\Delta s/2$ and $+\Delta s/2$ to the x, y and z coordinates of the location of the center of mass of the particle, where Δs is the maximum displacement determining the size of trial move. The new coordinates for particle i after translational trial move is:

$$\begin{aligned} x_i' &= x_i + (Ranf - 0.5)\Delta s, \\ y_i' &= y_i + (Ranf - 0.5)\Delta s, \\ z_i' &= z_i + (Ranf - 0.5)\Delta s. \end{aligned} \tag{III.3}$$

Where Ranf is a random number uniformly distributed between 0 and 1. Obviously, the reverse trail move from x_i' back to x_i is equally probable. For the sake of sampling efficient, we adopt the approach that move one particle at a time instead of moving all particles simultaneously.

Pertaining to how large Δs should be made, we should take it seriously since the choice of Δs affects the acceptance ratio of the trial move directly. If Δs is too small, the new configuration is not far from the current configuration, hence the acceptance probability is high, the trial move is probably accepted.Vice versa, if Δs is very large,the new configuration will possibly have a high energy and the trial move is likely to be rejected. The smaller Δs is, the higher the acceptance ratio will be.

The acceptance probability for particle displacement move is

$$\mathrm{acc}(o \to n) = \min\left\{1, \exp[-\beta[U(n) - U(o)]]\right\}. \tag{III.4}$$

Volume Change

In constant NPT ensemble, the volume of simulation box could be attempted to change. We perform random walk in the logarithm of the volume ln$V$ [18] by the following rule:

$$\ln V' = \ln V + (Ranf - 0.5)\Delta V_{max}.$$

where ΔV_{max} is the maximum volume change. Under the constraint of symmetry of the underlying Markov Chain, the acceptance probability of this trial move in the Metropolis scheme is

$$\begin{aligned} \mathrm{acc}(o \to n) = \min(1, \exp\{-\beta[U(n) - U(o) + P(V' - V) \\ - (N+1)\ln(V'/V)/\beta]\}). \end{aligned} \tag{III.5}$$

The pressure of the system is calculated using the virial

$$P = \frac{\rho}{\beta} + \frac{\mathrm{vir}}{3V}. \tag{III.6}$$

where the viral is defined as

$$\mathrm{vir} = \sum_i \sum_{j>i} f(r_{ij}) \cdot r_{ij} \tag{III.7}$$

where $f(r_{ij})$ is the intermolecular force. For the L-J model (in reduced units),

$$f(r_{ij}^*) = -\nabla U_{ij}^* = -4\left[-12\left(\frac{1}{r^*}\right)^{13} - (-6)\left(\frac{1}{r^*}\right)^{7}\right]. \tag{III.8}$$

Therefore,

$$\mathrm{vir}_{ij}^* = f(r_{ij}^*) \cdot r_{ij}^* = 24\left[2\left(\frac{1}{r^*}\right)^{12} - \left(\frac{1}{r^*}\right)^{6}\right]. \tag{III.9}$$

At equilibrium, the calculated pressure should be the same as the predefined pressure by minor statistical error.

III.2 Gibbs-NPT Ensemble

The Gibbs ensemble Monte Carlo methodology is a method to simulate phase equilibrium, which was first proposed by Panagiotopoulos et al. in 1987 for pure component [19] and Gibbs-NPT ensemble for multi components was developed and applied a year later [20]. Since then, Panagiotopoulos et al. have extended their phase equilibria studies in many aspects [21, 22, 23].

In the literature, there are a lot of successful applications of the Gibbs-NPT ensemble based simulations for many important binary or ternary mixtures equilibrium studies [24, 25, 26, 27].

In Gibbs-NPT ensemble, suppose there are two phases: liquid phase and vapor phase. The thermodynamic conditions for phase equilibrium are that each phase should be in internal equilibrium, and also the temperature, pressure and chemical potential of each species in all phases are equal, respectively. The phase equilibrium in Gibbs-NPT ensemble can be achieved by performing three types of Monte Carlo trial moves: particle displacement within each simulation box to satisfy the internal equilibrium, volume rearrangement to equilibrate the pressure and particle transfer between two phases to equilibrate the chemical potential.

A schematic diagram of the three types of trial move [28] is illustrated in Figure III.2.

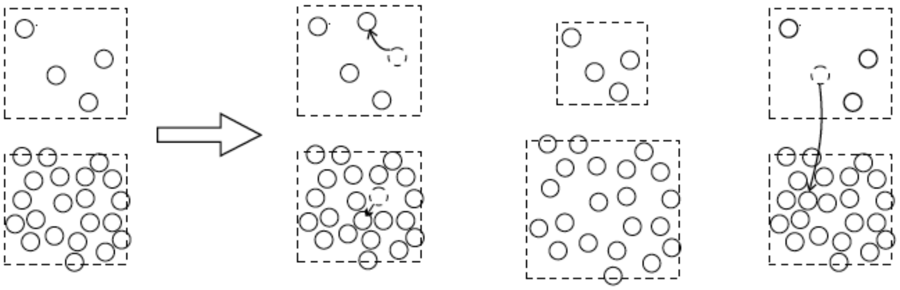

Figure III.1: A schematic diagram of the Gibbs-NPT ensemble simulation

The computational procedure goes as following: two simulation boxes are constructed. The first box contains N^Iparticles with volume V^I and the second box contains N^{II} particles with volume V^{II}. $N = N^I + N^{II}$ remains constant, so is the pressure and temperature. After the initialization of the configuration, the three types of Monte Carlo trial move are performed in random sequence with predefined probability.

III.2.1 Trial Moves

The rule for particle displacement is the same as discussed in constant NPT ensemble.

Volume Rearrangement

Differing from the single box volume change in NPT ensemble, in Gibbs-NPT ensemble, we change the volume of each simulation box by ΔV_1 and ΔV_2 independently. The acceptance probability is

$$\begin{aligned}\mathrm{acc}(o \to n) =& \min\{1, \exp[-\beta[\Delta U^I + \Delta U^{II} - N^I k_B T \ln \frac{V^I + \Delta V^I}{V^I} \\ &- N^{II} k_B T \ln \frac{V^{II} + \Delta V^{II}}{V^{II}} + P(\Delta V^I + \Delta V^{II})]]\}.\end{aligned} \tag{III.10}$$

where ΔU is the change in potential energy. For the sake of efficiency, in practice, a volume change for only one box is attempted at a time.

Particle Transfer

Firstly, randomly pick up box I or box II with equal probability, then choose the type of particle to transfer at random, and finally, move a randomly selected particle of the chosen type i from one simulation box to the other. Since the total number of the particles is conserved, the transfer consists of two parts: particle insertion in one box and particle destruction in the other box. Suppose a particle of component i is transferred from box II into box I. The trial move is accepted with probability

$$\mathrm{acc}(o \to n) = \min\left\{1, \exp\left[-\beta[\Delta U^{I} + \Delta U^{II} + k_B T \ln\left(\frac{V^{II}(N_i^{I}+1)}{V^{I} N_i^{II}}\right)]\right]\right\}. \quad \text{(III.11)}$$

where N_i^I is the number of particles of component i in box I.

Chapter IV

Statistical Errors

Monte Carlo simulation generally produces finite time series of correlated data and the result always contains statistical errors. The estimator based on the correlation function is popular for the error calculation on the time average of correlated data. However, the efforts required to compute a time correlation function is more significant than what required for a static average. In this thesis, we adopt the blocking method [29] proposed by H.Flyvbjerg and H.G.Petersen.

IV.1 Notation

In Monte Carlo simulation, we are interested in quantity $A = A(X)$, where $A_1, A_2, \cdots,$ A_n are n consecutive measurements taken after the system reaches equilibrium with exact but unknown probability distribution function under the condition that we know the probability distribution function of X. Our aim is to calculate the ensemble average and statistical error.

$\langle A\rangle$ denotes the expected value of A and $\bar{A}$ is the time average value over the set $\{A_1, A_2, \cdots, A_n\}$.

$$\langle A\rangle \equiv \int dt P(t) A(t), \tag{IV.1}$$

$$\bar{A} \equiv \frac{1}{n}\sum_{i=1}^{n} A_i, \tag{IV.2}$$

$\langle A\rangle$ is good for theoretical consideration and $\bar{A}$ is what we can compute in practice. Under the assumption of the ergodicity, which means the dynamic system has the same behavior average over time as averaged over space, we know that$\langle A\rangle$ is equal to the time average $\bar{A}$ in the limit $n \to \infty$.

Therefore, we are able to approximate the expected value $\langle A \rangle$ by the average value $\bar{A}$. Since $\bar{A}$ is a fluctuating quantity, we need to estimate the variance of $\bar{A}$

$$\sigma^2(\bar{A}) = \langle \bar{A}^2 \rangle - \langle \bar{A} \rangle^2. \tag{IV.3}$$

Substitute equation (IV.2) into (IV.3) and expand, we obtain

$$\sigma^2(\bar{A}) = \frac{1}{n^2} \sum_{i,j=1}^{n} \gamma(i,j), \tag{IV.4}$$

where

$$\gamma(i,j) \equiv \langle A_i A_j \rangle - \langle A_i \rangle \langle A_j \rangle, \tag{IV.5}$$

is the correlation function. By the stationary property of the random process

$$\gamma(i,j) = \gamma(i+T, j+T) = \gamma(i-j, 0). \tag{IV.6}$$

We define

$$\gamma(t) \equiv \gamma(i,j), \quad t = |i-j| \tag{IV.7}$$

With equation (IV.7),equation (IV.4) can be reformulated as

$$\sigma^2(\bar{A}) = \frac{1}{n}[\gamma(0) + 2\sum_{t=1}^{n-1}(1 - \frac{t}{n})\gamma(t)]. \tag{IV.8}$$

If the measurements A_i can be taken as independent from each other, then the variance $\sigma^2(\bar{A}) = \frac{\sigma^2(A)}{n}$, will be inversely proportional to the size of the data set. However, the samples are correlated in Markov chains and the correlation has to be taken into account.

IV.2 Blocking Method

The idea behind blocking method is to group data into consecutive blocks and compute average for each block. As the size of block is made larger, the block average will show less and less correlation. In the limit case, the block averages are independent Gaussian variables by the central limit theorem, then we can apply standard statistical formulas to obtain a reliable estimate of the standard deviation.

We transform the original data set $A_1, A_2, \cdots, A_n$ into $A'_1, A'_2, \ldots, A'_n$ by the following blocking transformation

$$\begin{aligned} A'_i &= \frac{1}{2}(A_{2i-1} + A_{2i}), \\ n' &= \frac{1}{2}n. \end{aligned} \tag{IV.9}$$

The new data set is half size of the original set and it is clear that the average value remains unchanged during above transformation.

$$\bar{A}' = \bar{A} \tag{IV.10}$$

Now let us investigate the correlation function for the new data set.

$$\begin{aligned} \gamma'(0) =& \gamma(A'_1, A'_1) = \langle A_1'^2 \rangle - \langle A'_1 \rangle^2 \\ =& \langle (\frac{1}{2}(A_0 + A_1))^2 \rangle - \langle \frac{1}{2}(A_0 + A_1) \rangle^2 \\ =& \frac{1}{4}[\langle A_0^2 \rangle + \langle A_1^2 \rangle + 2\langle A_0 A_1 \rangle - \langle A_0 \rangle^2 - \langle A_1 \rangle^2 - 2\langle A_0 \rangle \langle A_1 \rangle] \\ =& \frac{1}{2}[\gamma(0) + 2\gamma(1)]. \end{aligned} \tag{IV.11}$$

Similarly for $\gamma'(t), t > 0$, we have

$$\gamma'(t) = \frac{1}{4}[\gamma(2t-1) + 2\gamma(2t) + \gamma(2t+1)]. \tag{IV.12}$$

Insert equation (IV.11) and (IV.12) into the formulation of $\sigma^2(\bar{A}')$, we can prove that

$$\sigma^2(\bar{A}') = \frac{1}{n'^2} \sum_{i,j=1}^{n'} \gamma'(i,j) = \sigma^2(\bar{A}). \tag{IV.13}$$

Therefore, the average value and the variance of the new data set are both invariant under the blocking transformation, which makes the blocking method reasonable to employ.

From equation (IV.8), we know that

$$\sigma^2(\bar{A}) \geq \frac{\gamma(0)}{n}. \tag{IV.14}$$

The most obvious estimator for $\gamma(0)$ is

$$c_0 \equiv \frac{1}{n} \sum_{i=1}^{n} (A_i - \bar{A})^2, \tag{IV.15}$$

which is a biased estimator, and whose expectation value is not $\gamma(0)$, but

$$\langle c_0 \rangle = \gamma(0) - \sigma^2(\bar{A}). \tag{IV.16}$$

Combine equation (IV.14) and (IV.16), we obtain

$$\sigma^2(\bar{A}) \geq \left\langle \frac{c_0}{n-1} \right\rangle. \tag{IV.17}$$

The identity is satisfied when $\gamma(t) = 0, t > 0$, which means the block averages are uncorrelated.

Start from the original data set, we compute $c_0/(n-1)$, and apply blocking transformation, $c_0'/(n'-1)$ is computed again until $n' = 2$. The sequence of $c_0'/(n'-1)$ will increase until the plateau is reached. If there is no convergence, then we take the largest value of $c_0'/(n'-1)$ in the sequence as a lower bound of $\sigma^2(\bar{A})$.

Since $\sigma^2(\bar{A})$ is also a random variable, we can estimate the statistical error of $\sigma^2(\bar{A})$ in the similar way. This gives as estimate of the variance in the ensemble average

$$\sigma^2(\bar{A}) \approx \frac{c_0}{n'-1} \pm \sqrt{\frac{2}{n'-1}}\frac{c_0}{n'-1}. \tag{IV.18}$$

$$\sigma(\bar{A}) \approx \sqrt{\frac{c_0}{n'-1}}\left(1 \pm \frac{1}{\sqrt{2(n'-1)}}\right). \tag{IV.19}$$

The essence of blocking method is to reduce correlation via blocking transformation. The blocking method is more computationally economical than the correlation function method. In addition, the blocking method can give us extra information about whether we have simulated enough measurements. If we can not find the plateau, we should enlarge the simulation steps.

Chapter V

Results and Discussion

V.1 Simulation Based on NPT Ensemble

Computational Procedure

One-component Lennard-Jones fluid system is considered for the NPT Monte Carlo simulation. One simulation box is constructed for this purpose. 500 particles are used for simulation and the initial density for the box is taken as 0.3. The initial configuration is acquired by putting the desired number of particles on a face centered cubic lattice in the simulation box as illustrated in Figure V.1. The cutoff distance is chosen to be 0.45 of the box size with periodic boundary conditions. To impose the detailed balance condition, the two kinds of trial moves are selected randomly, the probability for selecting the displacement trial move is 0.9 and 0.1 for the volume change. The maximum displacement is adjusted to give an average acceptance ratio of around 50%. Tail correction for the potential energy and pressure are taken into account [30].

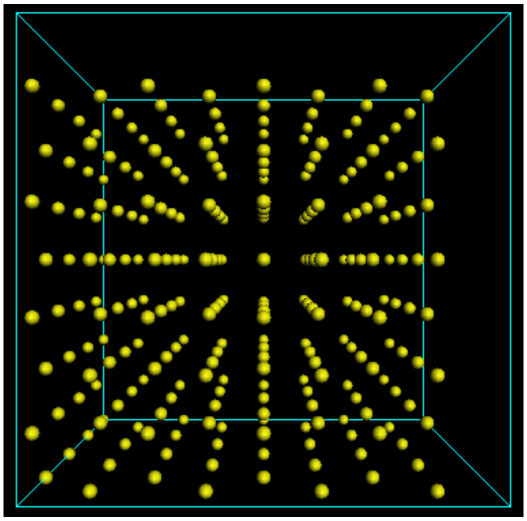

Figure V.1: Initial configuration

The program is implemented in Fortran 95 since Fortran is especially suited to numeric computation and scientific computing. There are six main subroutines: initialization, mcmove for particle displacement, mcvol for volume change, energy for energy calculation, RF for random number generation and sample for data collection. We have attached the pseudo code of mcmove and mcvol in Appendix B. Figure V.2 illustrates how the system look like when it reaches equilibrium.

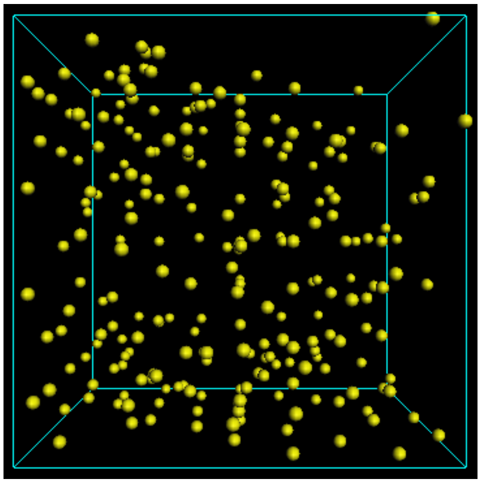

Figure V.2: Equilibrium configuration

Figure V.3 is the density evolution of the system for running 10^6 Monte Carlo cycles.

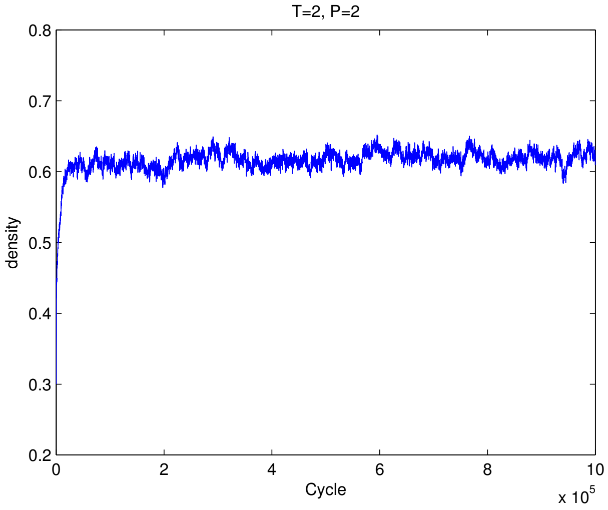

Figure V.3: Density evolution for $T^* = 2, P^* = 2$

V.1.1 Equation of State

Repeat the simulation for different pressures and compare the obtained density with a sophisticated equation of state for Lennard-Jones model.

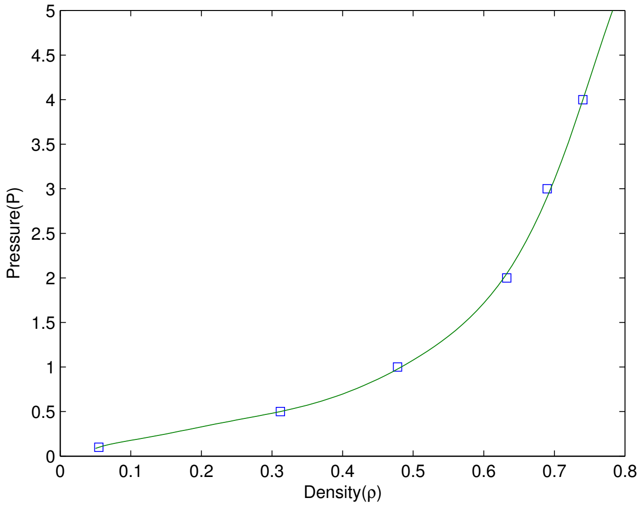

Figure V.4: Equation of State for normalized units

Figure V.4 illustrates the NPT simulation at $T^* = 2.0$ (higher than the critical temperature $T_c^* \approx 1.3$). The squares are the simulation results and the solid line is the equation of state by Johnson et al.[31]. It shows for the Lennard-Jones fluid, the results of NPT simulation agree very well with equation of state.

Methane (CH_4) molecule is a non-polar symmetric molecule made of a central carbon atom covalently bonded to four hydrogen atoms forming a tetrahedral structure. By convention, the force center is located at the center of the carbon atom. Therefore, each methane molecule could be represented by a point in the simulation box and what we only need to consider is the Lennard-Jones intermolecular interaction without taking Columbic effects into account.

The Lennard-Jones parameters for methane are $\epsilon/k_B = 147.5K$, $\sigma = 3.73 \times 10^{-10}\text{m} = 3.73\mathring{A}$ and M= 0.01604 Kg/mol [32]. We convert our simulation results to real values using the translation rule given in Table V.1.

Table V.1: Translation of reduced units to real units for methane

Quantity	Reduced units	Real units
Temperature	$T^* = 1$	T= 147.5 K
Density	$\rho^* = 1.0$	$\rho = 513.3\text{kg/m}^3$
Pressure	$P^* = 1$	P= 39.24 MPa

The comparison of simulation result with the experimental data of methane [33] is illustrated in Figure V.5. The results from simulation agree considerably with the experimental results with slight deviation when pressure is high.

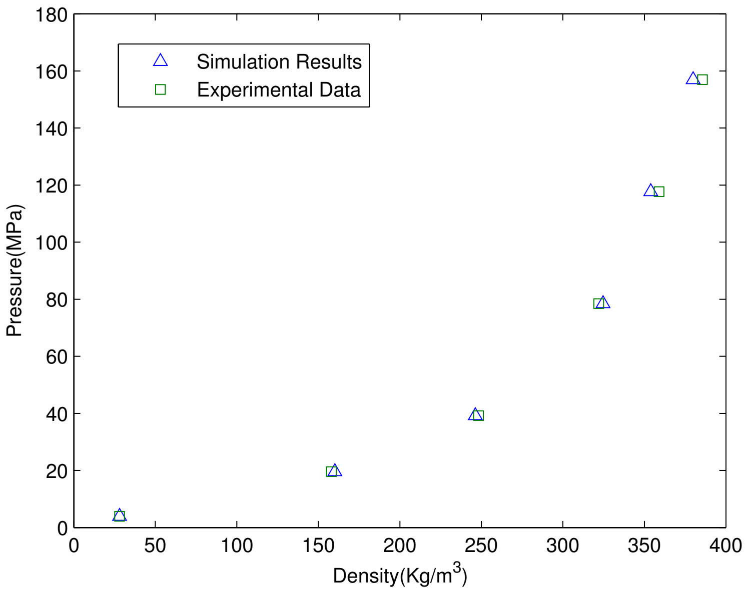

Figure V.5: Equation of State for methane

V.1.2 Error Analysis

Error Estimation

We estimate the statistical error in the average value of the simulation results using blocking method [29]. After 10^6 equilibrium cycles, 2^{23} production cycles are performed to generate samples for this variance analysis. The detailed data is shown in Appendix B. A demonstration of how the standard deviation is obtained by performing blocking transformation repeatedly is shown in Figure V.6. As we could see, a plateau in $c_0/(n-1)$ is observed between 19 and 22 block transformations and from this we find $\sigma = 4.8 \times 10^{-4}$. As expected, for a small value of block operations, the error increases until a plateau is reached. For high value of block operations, we have only a few data points, as a result, the statistical error in our estimate for standard deviation will be large.

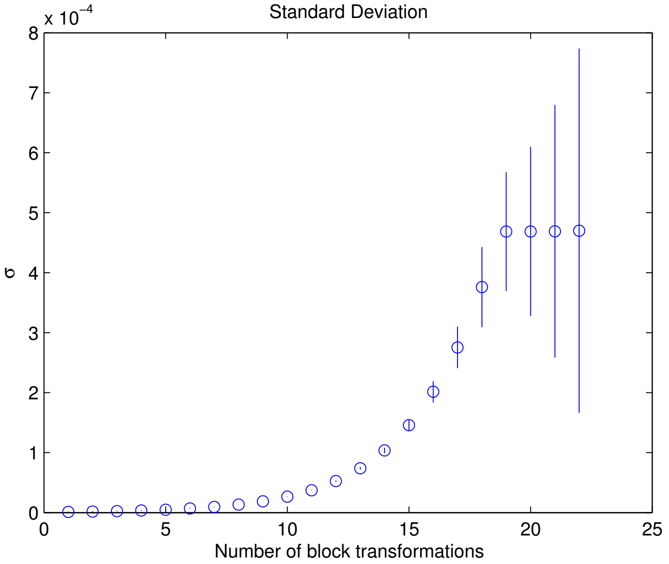

Figure V.6: Estimate for standard deviation

Apply the blocking process repeatedly to results obtained by running different Monte Carlo cycles. A comparison of how Monte Carlo cycles (M) could affect the standard deviation is illustrated in Figure V.7. Generally, the more cycles we run, the smaller standard deviation is.

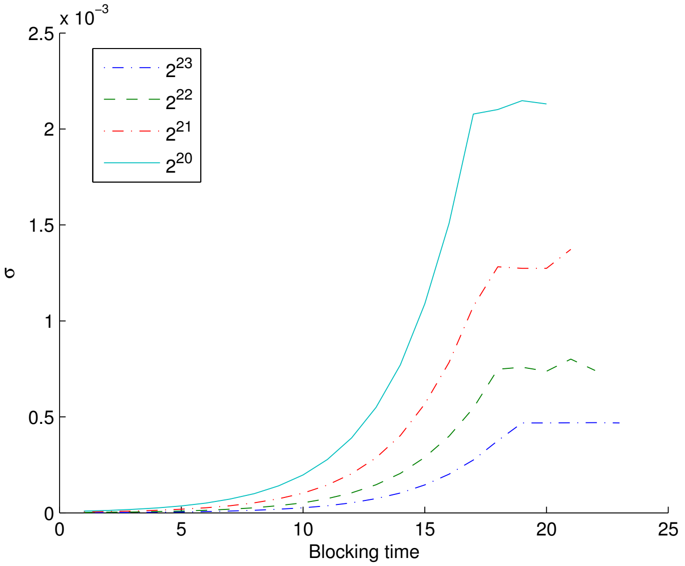

Figure V.7: Standard deviation at different cycles

We take the plateau values as estimate for the standard deviation in the blocking process and plot the standard deviation with respect to M in the logarithmic scale as illustrated in figure V.8.

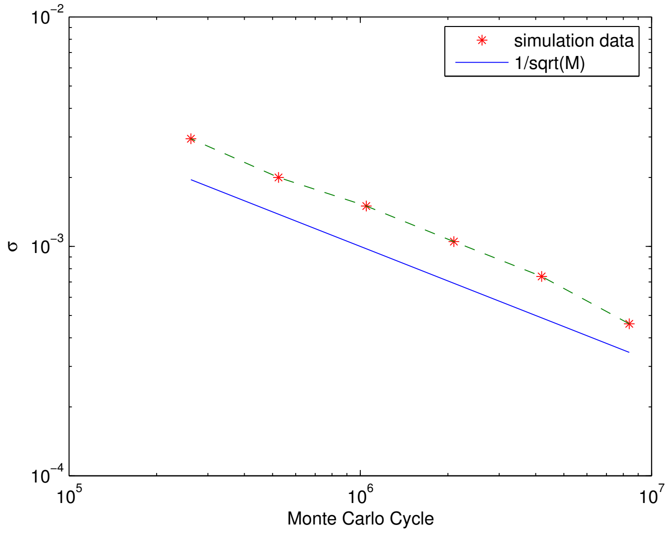

Figure V.8: Standard deviation at different values of M in logarithmic scale

The red stars are the computed data and is connected by the green line. The blue line is with slope $1/\sqrt{M}$ for comparison. It could be seen that the standard deviation is approximately proportional to $1/\sqrt{M}$, which also verifies that the convergence in Monte Carlo simulation.

Sample Interval

Let us take a look at the correlation between standard deviation and the sampling interval d. As we know, in simulation, two consecutive data points would be (highly) correlated. As the sample interval is made to be larger, the data would be less correlated. For the simulation result of size 2^{23}, take the sample interval to be $d = 2^i$, the corresponding data set is of size 2^{23-i}.

As we could see that if we take $d = 2^i$, the standard deviation grows as d increases since the data become less correlated. If the simulation data are independent from

Table V.2: Standard deviation vs sample interval

d	1	2	4	8	16	32	64
$\sigma/10^{-4}$	4.682541	4.682541	4.684837	4.686446	4.687606	4.689011	4.689102

each other, when the sample interval increases by a factor of n, i.e. the number of data decreases by a factor of n, the standard deviation will increase by a factor of $\sqrt{n}$. However, since the data from MC simulation are highly correlated, the increase in the standard deviation is not substantial. Therefore, we can obtain the desired estimate for standard deviation using proper size of interval d by storing much less data.

Error Propagation

In the early implementation, we adopt the scaling method for energy calculation in volume change trial moves in [5]. The scaling rule is as following:

$$\begin{aligned} U' &= U * a(2-a) - \text{vir} * a(1-a)/6, \\ \text{vir}' &= -12U * a(a-1) + \text{vir} * a(2a-1). \end{aligned} \tag{V.1}$$

where U' represents the potential energy after the volume change from V to V' and vir$'$ is the new virial. $a = (L/L')^6$, L is the simulation box length.

However, divergence for the simulation results is observed after around 10^6 MC cycles, the volume becomes very small and the potential energy goes to infinity. As a result, the error propagation in the simulation process needs to be serious investigated.

First of all, let us take a look at how the scaling method works. For the volume change, assume the box length changes from L to L', the coordinates of the molecules is rescaled by the same ratio and the cut off radius will change correspondingly too. Therefore, those molecules had intermolecular interactions still have the interactions.

The molecules which make contributions to the potential energy and the virial remain the same. After the volume change, the distance between molecules i and j is $r\frac{L'}{L} = rf$, the new potential becomes

$$\begin{aligned} u'_{i,j}(rf) &= 4\left[\left(\frac{1}{rf}\right)^{12} - \left(\frac{1}{rf}\right)^{6}\right] \\ &= 4\frac{1}{f^6}\left[\frac{1}{f^6}\left(\frac{1}{r}\right)^{12} - \left(\frac{1}{r}\right)^{6}\right]. \end{aligned} \tag{V.2}$$

while

$$\begin{aligned} &U_{i,j} * a(2-a) - \mathrm{vir}_{i,j} * a(1-a)/6 \\ &= 4\left[\left(\frac{1}{r}\right)^{12} - \left(\frac{1}{r}\right)^{6}\right] * \frac{1}{f^6}(2 - \frac{1}{f^6}) - 24\left[2\left(\frac{1}{r}\right)^{12} - \left(\frac{1}{r}\right)^{6}\right] * \frac{1}{f^6}(1 - \frac{1}{f^6})/6 \\ &= 4\frac{1}{f^6}\left[\frac{1}{f^6}\left(\frac{1}{r}\right)^{12} - \left(\frac{1}{r}\right)^{6}\right]. \end{aligned} \tag{V.3}$$

which verifies the validation of the scaling method. We find out $a = \frac{1}{f^6} = (\frac{L}{L'})^6$ is the mainly used factor for computation. The error propagation rule for $a = \frac{1}{f^6}$ is

$$\Delta a = \mathrm{abs}(\nabla(\frac{1}{f^6})) = 6\frac{\Delta f}{f^7}. \tag{V.4}$$

$$\Delta a^2 = \mathrm{abs}(\nabla(\frac{1}{f^{12}})) = 12\frac{\Delta f}{f^{13}}. \tag{V.5}$$

when $f < 1$, i.e. the simulation box shrinks, the absolute error in a will be enlarged and so is for the potential energy. As we perform MC cycles, the error will accumulate and eventually blow off, which result in the phenomena we observed.

Alternative approaches avoiding the error accumulation is to recompute the new energy for all molecules, which is safe but needs more computation efforts or recompute the new energy for all molecules after a certain number of moves, like 10^3, before the accumulated error is non negligible.

RNG Performance Comparison

In this thesis work, we apply the intrinsic Fortran random number generator, the linear congruential generator (LCG) with modulus 10^9 and TT800 random number generator in our simulations. We run the Monte Carlo simulations with 10^6 equilibrium cycles and 2^{23} production cycles. Other system parameters remain the same as what we have defined in the previous section. The CPU time is recorded and the standard deviation is calculated for performance comparison.

Table V.3: RNG performance comparison

	Intrinsic RNG	LCG	TT800
CPU Time/s	5392	4832	5812
σ	2.77×10^{-5}	4.68×10^{-4}	7.78×10^{-7}

As we could see from Table V.3, there is no best choice among this three Random Number Generators. TT800 can obtain the most accurate results with consuming most time, while LCG is the most time saving with low quality results. The intrinsic Fortran random number ranks second in both metrics. In the previous error and CPU time analysis, LCG is applied for the sake of time saving.

V.1.3 Number of Particles

CPU Time

It is easy to conclude that the CPU time is proportional to the total number of MC cycles M. Now, let us investigate how the CPU time would change as the number of particles N vary In MC, the energy calculation is the most time consuming part. If the potential model is pairwise additive interactions for particles i, without truncation, we need to calculate the interactions with the other N-1 particles. In total, the number of interactions is $\frac{N(N-1)}{2}$, the complexity is $O(N^2)$. Even with truncation, we still first need to calculate $\frac{N(N-1)}{2}$ times distance to determine the particles within the cutoff distance with particle i.

In our NPT simulation, we need to conduct energy calculation in every trial move. For molecule displacement trial move, we only need to calculate the energy difference caused by displacement of one particle, hence the complexity of calculating energy is $O(N)$. For volume change trial move, we have to compute the total energy before and after the volume change, and the complexity is $O(N^2)$.

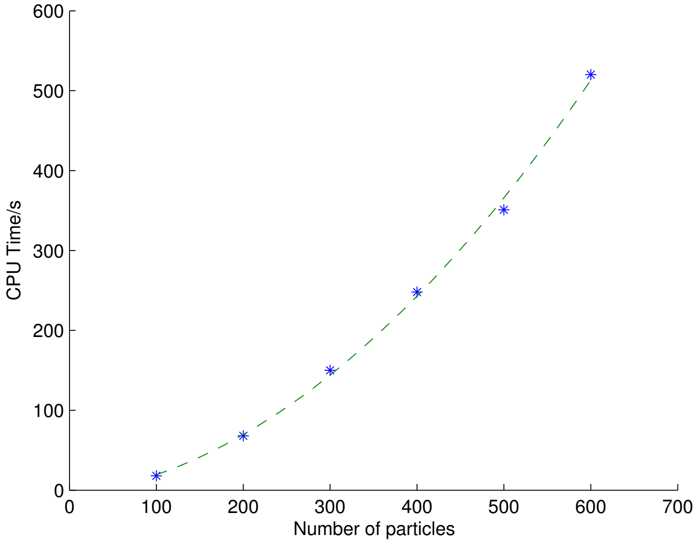

Figure V.9: CPU time vs N

The above figure illustrates the CPU time fits well as the quadratic function

$$t = 0.12 \times 10^{-2} N^2 + 0.14N.$$

of number of particles. The coefficients of quadratic function are connected to the parameters: number of Monte Carlo cycles, the probability set, computer performance. Here, the probability of selecting the displacement trial move is 0.9 and 0.1 for the volume changes.

Error Comparison

We have analyzed the running CPU time with respect to the number of particles N in the previous section, the time grows quadratically with N. A good simulation is expected to consuming proper running time as well as obtaining high accurate results. For the accuracy analysis, ten simulations are conducted at different values of N ranging from 100 to 1000 at constant pressure $P^* = 1.2743$, temperature $T^* = 2.034$ for 10^5 equilibrium cycles and 2^{18} production cycles.

The reduced densities ($\rho^* = N/V$) are collected and the relative error in the densities can be evaluated by comparing to the real experimental data $\rho^*_{\text{true}} = 0.5293$ [33]. The relative error is computed by

$$e = \frac{|\rho^* - \rho^*_{\text{true}}|}{\rho^*_{\text{true}}}.$$

As illustrated in Figure V.10, when $N < 300$, the relative error in the density is unacceptably high, as the N increases to 300, the relative error has a sharp decrease. For $N \geq 300$, there are fluctuations with the general trend of decline.

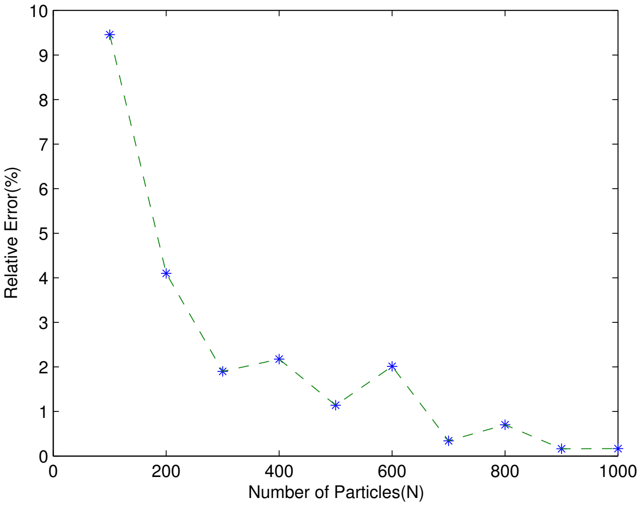

Figure V.10: Relative error in density at different values of N

Table V.4 displays the reduced density and standard deviation of simulation results with respect to $N \geq 300$, where / means there is no convergence observed in the blocking process, which implies that the imposed the number of Monte Carlo cycles M for $N = 300, N = 400$ is not sufficient.

Table V.4: Reduced density and standard deviation of results with different N

N	300	400	500	600	700	800	900	1000
ρ^*	0.5192	0.5177	0.5232	0.5186	0.5311	0.5255	0.5301	0.5284
$\sigma/10^{-3}$	/	/	6.1693	13.092	7.0937	4.5571	1.9240	2.7878

As it could be observed on the table V.4, there is a general trend of decline in the standard deviation for fixed M as N increases, however, no exact formula for the relation between σ and N could be discovered.

V.2 Simulation Based on Gibbs-NPT Ensemble

Two components system is considered in the Gibbs-NPT ensemble Monte Carlo simulation for phase equilibrium. We choose methane CH_4 and ethane C_2H_6 as binary mixtures since they are very important hydrocarbons in the oil industry and simple to use the Lennard-Jones model.

We run the simulations using the Gibbs module of MedeA platform. MedeA could be used to predict the materials properties and thermodynamics by simulations based on classical and quantum mechanics, statistical thermodynamics and correlation methods involving the experimental data. In particular, the Gibbs module could be employed to calculate the properties of single and multi-phase fluids and adsorption isotherms of fluids.

The computational procedure goes as following: build the methane and ethane molecules first, and then set up system variables (temperature, pressure) and determine the values of parameters. 500 molecules for each simulation box is initialized and the initial composition for the mixture is set between the liquid and vapor compositions to allow phase split. The three kinds of trial moves are selected randomly, and the probabilities for particle displacement, volume rearrangement and particle transfer are 0.5, 0.1 and 0.4 respectively.

V.2.1 Phase Equilibrium

In Figure V.11, we plot our simulation results as well as the experimental data [34, 35] for comparison.

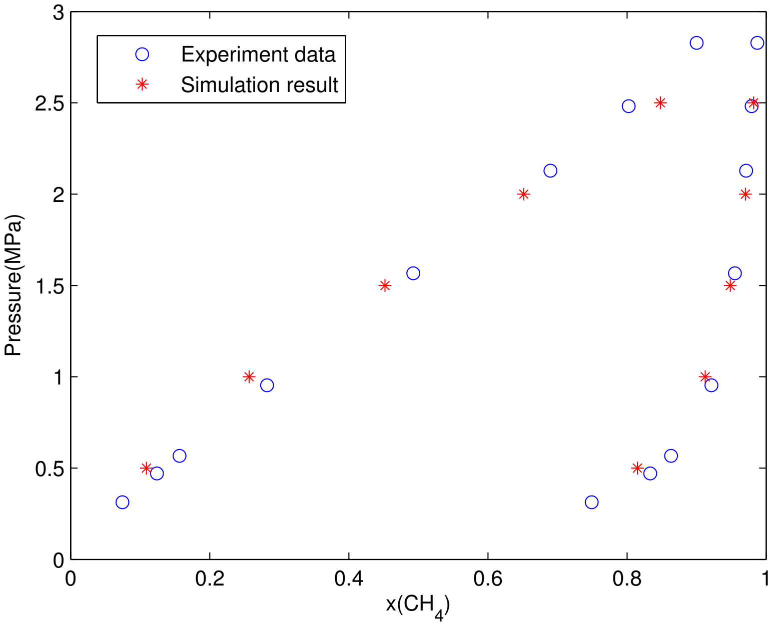

Figure V.11: Pressure composition diagram at 180K

It could be seen from the figure that the simulation results are generally in good agreement with the experimental data. Detailed simulated data of thermodynamic properties are shown in Table V.5, where A represents CH_4 and B denotes C_2H_6, x_A is the mole fraction of CH_4 in liquid phase and y_A is the mole fraction of CH_4 in vapor phase. ρ_L is the density of the mixture in liquid phase and ρ_V is the density of the mixture in vapor phase.

Although Gibbs NPT simulation does not require information about the chemical potential μ, we know that when the system reaches equilibria, for each component, the chemical potential in two phases should be identical. By computing μ, we could test the convergence of simulation and verify the chemical potential equilibrium. μ is calculated using equation (V.6) by Widom method [36]

$$\mu_{ji} = -k_B T \ln \langle \frac{V_j}{N_{ji}+1} \exp(-\frac{U_{ji}}{k_B T}) \rangle, \tag{V.6}$$

where μ_{ji} denotes the chemical potential for component i in box j, V_j is the volume of simulation box j. N_{ji} is the number of molecules of component i in simulation box j. U_{ji} is the potential of component i in simulation box j.

Table V.5: Thermodynamic properties for binary mixtures CH_4/C_2H_6 at 180K

Pressure (MPa)	ρ_L (g/cm^3)	ρ_V	y_A	x_A	μ_A^L	μ_B^L	μ_A^V	μ_B^V
2.5	0.3706	0.03703	0.982	0.848	-1286.1	-2073.9	-1286.22	-2073.4
2.0	0.4372	0.0285	0.9703	0.6516	-1315.8	-2042.3	-1315.4	-2043.2
1.5	0.5482	0.01975	0.9485	0.452	-1366	-1929.1	-1366.72	-1925.6
1.0	0.6267	0.012812	0.9124	0.2567	-1439.1	-1889	-1437.88	-1890.5
0.5	0.6842	0.006551	0.815	0.109	-1578	-1861	-1574.7	-1861.2

As we could see on table V.5, in equilibrium, the chemical potential of each component in two phases are the same within tolerant error. The density and mole fraction results agree with the experimental data considerably. Since the critical temperature of ethane ($T_c = 305.4K$) is higher than the critical temperature of methane ($T_c = 190.5K$), as the pressure increases, ethane is first compressed into liquid phase and the mole fraction of ethane in the gas phase decreases. As the pressure keeps increasing, more methane are compressed into liquid phase and almost all ethane molecules exist in the liquid phase.

Chapter VI

Concluding Remarks

The pressure density diagram we obtained by conducting NPT ensemble Monte Carlo simulations for Lennard-Jones fluid using reduced units fits well with the equation of state by Johnson et al. With the reduced units to be transformed real units, the results also agree considerably with the experimental data of methane.

We apply the blocking method to estimate the standard deviation of simulation result. The main idea is to reduce correlation of consecutive blocks by performing blocking transformation and in the end the block averages become uncorrelated. After calculating the standard deviation for different sizes of MC cycles, we found out that the standard deviation is inversely proportional to the square root of the MC cycles, i.e. $\sigma \propto 1/\sqrt{M}$. Since the result data are highly correlated, we can take proper size of sample interval for the variance estimate. Random Number Generator is also a key component in MC simulations, therefore a comparison of three RNGs on CPU time and standard deviation is conducted. We should choose the RNG based on what we would like to optimize in the simulation, better results or less time. The CPU time is recorded for different numbers of particles, which verifies that the complexity of the Monte Carlo with NPT ensemble algorithm is O(N^2).

For the Gibbs-NPT ensemble method employed for phase equilibrium of methane and ethane. Good agreement is found between the simulation results of phase coexistence and the experiment data over a wide range of pressure. In addition, we compute the chemical potential for each component in both phases, which could provide us extra information about the convergence of the equilibrium and criteria to test whether the equilibrium is reached.

Monte Carlo simulation, and the molecular based simulation is very promising and effective to determine phase behaviors and material properties. One future work would be to focus more on the efficiency of the Monte Carlo algorithm using advanced computing facilities or by parallelization. Another future work is the investigation and implementation of complex or polar molecules, where the Lennard-Jones is not sufficient to model the molecular interactions, requiring more accurate and sophisticated potential models.

BIBLIOGRAPHY

[1] J. Haile, “Molecular dynamics simulation: elementary methods,” *Physical Review Letters*, vol. 104, no. 19, 1992.

[2] K. Binder, *Monte-Carlo Methods.* Wiley Online Library, 1979.

[3] J. Liu, *Monte Carlo strategies in scientific computing.* Springer Verlag, 2008.

[4] C. Geyer, “Practical markov chain monte carlo,” *Statistical Science*, vol. 7, no. 4, pp. 473–483, 1992.

[5] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications.* Academic Press, Inc., 1996.

[6] D. Peng and D. Robinson, “A new two-constant equation of state,” *Industrial & Engineering Chemistry Fundamentals*, vol. 15, no. 1, pp. 59–64, 1976.

[7] G. Soave, “Equilibrium constants from a modified redlich-kwong equation of state,” *Chemical Engineering Science*, vol. 27, no. 6, pp. 1197–1203, 1972.

[8] N. Meteopolis and S. Ulam, “The monte carlo method,” *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.

[9] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The journal of chemical physics*, vol. 21, p. 1087, 1953.

[10] P. Hellekalek, “Good random number generators are (not so) easy to find,” *Mathematics and Computers in Simulation*, vol. 46, no. 5, pp. 485–505, 1998.

[11] M. Matsumoto and T. Nishimura, “Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator,” *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, 1998.

[12] S. Nosé, “A molecular dynamics method for simulations in the canonical ensemble,” *Molecular Physics*, vol. 52, no. 2, pp. 255–268, 1984.

[13] D. Adams, “Grand canonical ensemble monte carlo for a lennard-jones fluid,” *Molecular Physics*, vol. 29, no. 1, pp. 307–311, 1975.

[14] W. Wood, “Monte carlo calculations for hard disks in the isothermal-isobaric ensemble,” *The Journal of Chemical Physics*, vol. 48, p. 415, 1968.

[15] I. McDonald, “Npt-ensemble monte carlo calculations for binary liquid mixtures,” *Molecular Physics*, vol. 23, no. 1, pp. 41–58, 1972.

[16] J. Owicki and H. Scheraga, “Monte carlo calculations in the isothermal-isobaric ensemble. 2. dilute aqueous solution of methane,” *Journal of the American Chemical Society*, vol. 99, no. 23, pp. 7413–7418, 1977.

[17] D. Corti, “Isothermal-isobaric ensemble for small systems,” *Physical Review E*, vol. 64, no. 1, p. 016128, 2001.

[18] R. Eppenga and D. Frenkel, “Monte carlo study of the isotropic and nematic phases of infinitely thin hard platelets,” *Molecular physics*, vol. 52, no. 6, pp. 1303–1334, 1984.

[19] A. Panagiotopoulos, “Direct determination of phase coexistence properties of fluids by monte carlo simulation in a new ensemble,” *Molecular Physics*, vol. 61, no. 4, pp. 813–826, 1987.

[20] A. Panagiotopoulos, N. Quirke, M. Stapleton, and D. Tildesley, “Phase equilibria by simulation in the gibbs ensemble,” *Molecular Physics*, vol. 63, no. 4, pp. 527–545, 1988.

[21] J. Errington and A. Panagiotopoulos, “A fixed point charge model for water optimized to the vapor-liquid coexistence properties,” *The Journal of Physical Chemistry B*, vol. 102, no. 38, pp. 7470–7475, 1998.

[22] A. Panagiotopoulos, “Direct determination of fluid phase equilibria by simulation in the gibbs ensemble: a review,” *Molecular simulation*, vol. 9, no. 1, pp. 1–23, 1992.

[23] J. Errington and A. Panagiotopoulos, “A new intermolecular potential model for the n-alkane homologous series,” *The Journal of Physical Chemistry B*, vol. 103, no. 30, pp. 6314–6322, 1999.

[24] J. De Pablo and J. Prausnitz, “Phase equilibria for fluid mixtures from monte-carlo simulation,” *Fluid phase equilibria*, vol. 53, pp. 177–189, 1989.

[25] S. Moon, “Monte carlo simulation for vapor-liquid equilibrium of binary mixtures,” *BULLETIN-KOREAN CHEMICAL SOCIETY*, vol. 23, no. 6, pp. 811–816, 2002.

[26] M. van Leeuwen and B. Smit, “Molecular simulation of the vapor-liquid coexistence curve of methanol,” *The Journal of Physical Chemistry*, vol. 99, no. 7, pp. 1831–1833, 1995.

[27] B. Smit, S. Karaborni, and J. Siepmann, “Computer simulations of vapour-liquid phase equilibria of n-alkanes.” *Journal of chemical physics*, vol. 102, p. 2126, 1995.

[28] A. Panagiotopoulos, “Monte carlo methods for phase equilibria of fluids,” *Journal of Physics: Condensed Matter*, vol. 12, p. R25, 2000.

[29] H. Flyvbjerg and H. Petersen, “Error estimates on averages of correlated data,” *The Journal of Chemical Physics*, vol. 91, p. 461, 1989.

[30] J. Li, S. Sun, and V. Calo, “Monte carlo molecular simulation of phase-coexistence for oil production and processing,” in *SPE Reservoir Characterisation and Simulation Conference and Exhibition*, 2011.

[31] J. Johnson, J. Zollweg, and K. Gubbins, “The lennard-jones equation of state revisited,” *Molecular Physics*, vol. 78, no. 3, pp. 591–618, 1993.

[32] J. Errington and A. Panagiotopoulos, “Phase equilibria of the modified buckingham exponential-6 potential from hamiltonian scaling grand canonical monte carlo,” *The Journal of chemical physics*, vol. 109, p. 1093, 1998.

[33] U. Setzmann and W. Wagner, “A new equation of state and tables of thermodynamic properties for methane covering the range from the melting line to 625 k at pressures up to 1000 mpa,” *J. Phys. Chem. Ref. Data*, vol. 20, no. 6, pp. 1061–1151, 1991.

[34] R. Miller, A. Kidnay, and M. Hiza, “Liquid+ vapor equilibria in methane+ ethene and in methane+ ethane from 150.00 to 190.00 k,” *The Journal of Chemical Thermodynamics*, vol. 9, no. 2, pp. 167–178, 1977.

[35] I. Wichterle and R. Kobayashi, “Vapor-liquid equilibrium of methane-ethane system at low temperatures and high pressures,” *Journal of Chemical and Engineering Data*, vol. 17, no. 1, pp. 9–12, 1972.

[36] B. Widom, “Some topics in the theory of fluids,” *The Journal of Chemical Physics*, vol. 39, p. 2808, 1963.

[37] S. Sandler, *An Introduction to Applied Statistical Thermodynamics.* Wiley, 2010.

APPENDICES

Appendix A

Statistical Mechanics

Here we review one basic postulate of statistical thermodynamics[37].

Postulate 1. *(Ergodic hypothesis) The long time average of any mechanical property in a real macroscopic system is equal to the average value of that property over all the microscopic states of the system.*

The ergodic hypothesis means that any experimental measurement is a long time measurement on a molecular time scale. As a result, we can use a statistical average to replace time average.

The partition functions is a description of the statistical properties of a system when it is in thermodynamic equilibrium. Most of the thermodynamic properties, such as internal energy, entropy and pressure, can be formulated in terms of the partition function or the derivatives.

Assume we are dealing with the canonical ensemble, which means the temperature, volume and number of particles of the system are fixed. In the quantum statistical mechanics, the energy of molecular states are discrete values, not continuous. An energy state is a particular specification of quantum numbers. Label the micro states that a system can occupy with $s(s = 1, 2, 3, ...)$ and E_s represents the total energy or the energy level of system when it is in micro state s. The partition function for the canonical ensemble is defined as

$$Z = \sum_s \exp[-\beta E_s]. \tag{A.1}$$

where $\beta = 1/k_B T$, k_B is the Boltzmann constant.

However, in classical statistical mechanics, the momentum and position variables of particles can change continuously, the micro states become unaccountable. Therefore, the summation in the above equation is replaced with an integral. Suppose we have N identical particles in the system, the partition function can be reformulated as

$$Z = c \int dr^N dp^N \exp[-\beta H(r^N, p^N)]. \tag{A.2}$$

where c is a constant. r^N denotes the coordinates of all N particles, p^N is for the momenta. H is the Hamiltonian, it express the total energy of the system by summing the kinetic energy and potential energy.

It should be noted that for different ensembles, there are corresponding partition functions since the the partition function is a function of temperature and other parameters, like volume.

If we want to compute the ensemble average of property A, then we have

$$\langle A \rangle = \frac{\int dr^N dp^N \exp\left[-\beta H(r^N, p^N)\right] A(\mathbf{r}^N)}{\int dr^N dp^N \exp\left[-\beta H(r^N, p^N)\right]}. \tag{A.3}$$

Since kinetic energy is a quadratic function of the momentum, we can carry out them analytically, reducing the Hamiltonian to the potential energy only without involving the momentum. Equation (A.4) will be used for the calculation of ensemble average.

$$\langle A \rangle = \frac{\int dr^N \exp\left[-\beta U(r^N)\right] A(\mathbf{r}^N)}{\int dr^N \exp\left[-\beta U(r^N)\right]}. \tag{A.4}$$

The significance of the partition function is that it has very important statistical meaning. The probability of finding the system in micro state s is

$$P_s = \frac{1}{Z} e^{-\beta E_s}. \tag{A.5}$$

Therefore, the partition function is a normalizing constant guaranteeing the probabilities in all possible micro states sum up to one:

$$\sum_s P_s = \frac{1}{Z}\sum_s e^{-\beta E_s} = \frac{1}{Z}Z = 1. \tag{A.6}$$

This is where the name partition function comes: it shows how the probabilities are partitioned among all possible different micro states, based on the energy of micro state.

Most of the thermodynamic variables of the system can be related to the partition function as follows.

$$\text{Pressure} = k_B T(\frac{\partial \ln Z}{\partial V})_{T,N}. \tag{A.7}$$

$$\text{Entropy} = k_B \ln Z + k_B T(\frac{\partial \ln Z}{\partial T})_{V,N}. \tag{A.8}$$

$$\text{Internal energy} = k_B T^2(\frac{\partial \ln Z}{\partial T})_{V,N}. \tag{A.9}$$

Appendix B

Simulation Supplementary

Simulation Data

Table B.1: Results of repeated application of block transformation

Blocking time	n	$c_0/n-1$	$\sigma[c_0/n-1]$
0	8 388 608	1.208988E-06	5.903262E-10
1	4 194 304	1.709180E-06	1.180247E-09
2	2 097 152	2.415883E-06	2.359262E-09
4	524 288	4.818336E-06	9.410822E-09
6	131 072	9.559574E-06	3.734223E-08
8	32 768	1.885825E-05	1.473323E-07
10	8 192	3.728971E-05	5.826873E-07
12	2 048	7.388723E-05	2.309540E-06
14	512	1.458701E-04	9.125797E-06
16	256	2.754037E-04	3.456073E-05
18	64	4.686011E-04	0.990248E-04
20	16	4.690203E-04	2.107019E-04
22	4	4.682541E-04	4.243705E-04

Table B.2: Experimental data for phase coexistence of CH_4/C_2H_6 at 180K

P/MPa	0.313	0.471	0.567	0.953	1.567	2.128	2.482	2.828
x	0.0745	0.1241	0.1567	0.2825	0.4928	0.6898	0.8025	0.902
y	0.7493	0.8335	0.8633	0.9213	0.9551	0.9712	0.9792	0.9874

Density Evolution

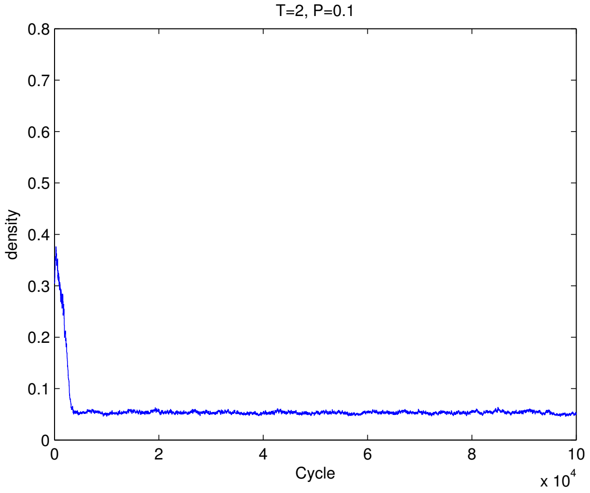

Figure B.1: Density evolution for $T^* = 2, P^* = 0.1$

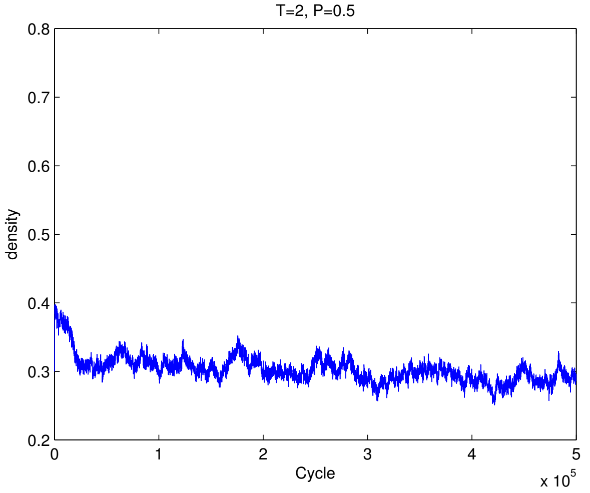

Figure B.2: Density evolution for $T^* = 2, P^* = 0.5$

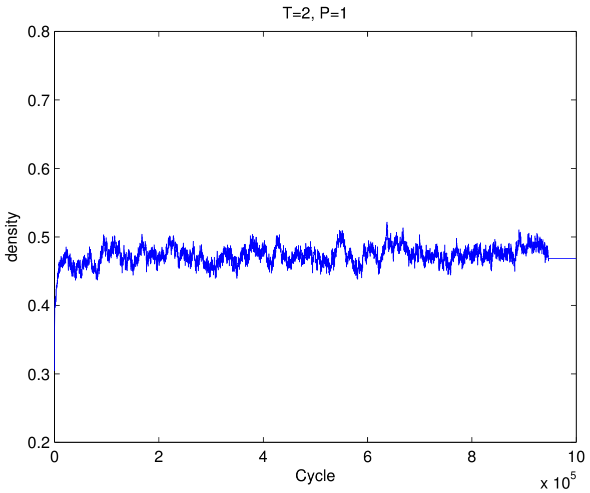

Figure B.3: Density evolution for $T^* = 2, P^* = 1$

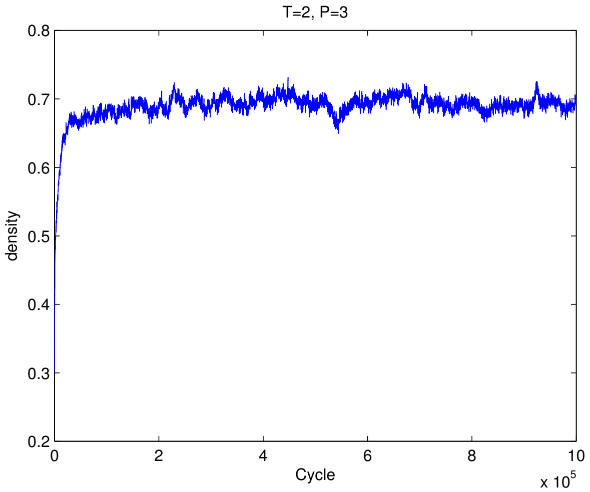

Figure B.4: Density evolution for $T^* = 2, P^* = 3$

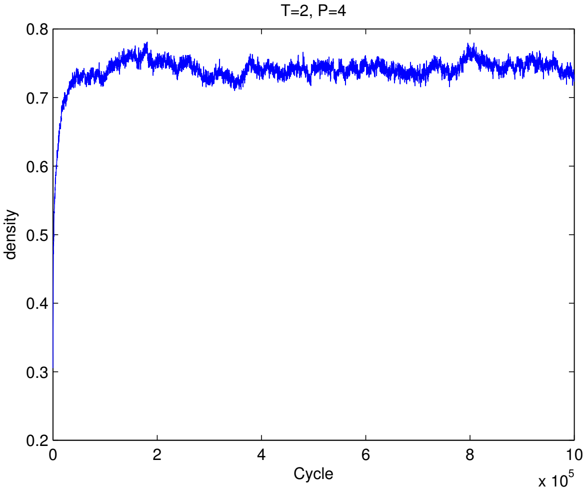

Figure B.5: Density evolution for $T^* = 2, P^* = 4$

Pseudo Code

Attached is the pseudo code for the mcmove and mcvol subroutines.

```
! Perform particle displacement
    Subroutine mcmove
        ! Select a particle randomly
        old = int(Ranf()×N)+1
        call energy(X(old), en_old)
        ! Translational particle displacement
        ! X is the three dimensional coordinates vector
        xnew = X(old) +(Ranf()-0.5)$\Delta s$
        call energy(xnew, en_new)
        ! Acceptance rule
        if (Ranf() < exp(-$\beta$(en_new-en_old)))
            ! Put the particle back to box if it is out
            ! Update the coordinates of particles
            X(old) = xnew
        end if
    end
! Perform particle displacement
    Subroutine mcvol
        call total_energy(enold)
        v_old = box$^3$
        ! Perform random walk in ln(V)
        ln_volume = log(v_old) + (Ranf()-0.5)$\Delta V$
        v_new= exp(ln_volume)
        box_new = v_new$^{\frac{1}{3}}$
        ! Scale the center of mass
        do i=1:N
            X(i) = X(i) $*$ box_new/box_old
        end do
        call total_energy(ennew)
        arg = -$\beta *$(en_new-en_old + P(v_new-v_old)-(N+1)*log(v_new/v_old)/$\beta$)
        if (Ranf() $<$ exp(-arg) )
           ! accept
        else
           ! Restore to the old configuration
        end if
    end
```