Molecular Dynamcis Simulation:
Phase Coexistence Curve and Properties of Lennard-Jones Fluid

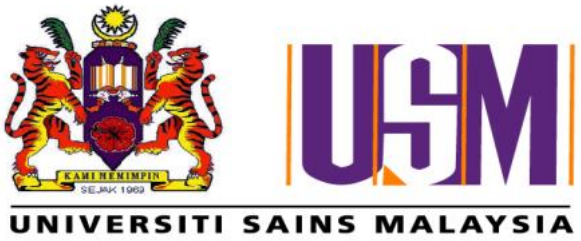
by

GOH EONG SHENG

SUPERVISOR

DR YOON TIEM LEONG

Final Year Project report submitted in fulfillment of the requirements for the Bachelor of Science (Honours) (Physics)



School of Physics

Universiti Sains Malaysia

11800 Pulau Pinang

2012/2013

Acknowledgement

This project would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

I would like to express my deepest appreciation to my supervisor, Dr Yoon Tiem Leong, who has the attitude and substance of a respectful physicist. He continually and convincingly conveyed a spirit of adventure in regard to research and writing of this project. Dr Yoon has been a source of my inspiration, whose sincerity and encouragement I will never forget.

I also conveyed my thanks to Universiti Sains Malaysia (USM) and especially USM School of Physics, for providing me an opportunity to conduct my research. I acknowledge that I have made full use of the computational power provided by the computer lab for School of Physics, without which I would not be able to obtain my results.

Utmost gratitude will have to be given to all the staffs of the School of Physics, for keeping the School operating so well even in the semester break. I would like to especially thank Mr. Anas and the other staffs, who are in charge of the computer lab. They have undeniably fulfilled their responsibility and opened the computer lab every day for me and my partners to use the clusters.

I would like to express my gratefulness to my partner, Ms. Soon Yee Yeen. She has always gave me encouragement during my hardship. Her persistent researches on our projects have served as a form of encouragement and boost my confidence towards completing my projects.

Last but not the least, my family and friends, especially my senior Mr Ng Wei Chun, who shares his experience and expertise with me, Mr Goh Pu Khiaw, who always accompanies me in my venture and constantly reminding me of the ultimate truth of the world, and Mr Lim Wei Qiang, whose relentless support and support during my time in the computer lab.

Abstract

Molecular simulation is a viable approach to study the reaction and properties of a system by linking the macroscopic properties of matter with their molecular details and interactions of particles. Computational approach acts as an immediate field bridging the theoretical and experiment approach, allowing testing of a theory to done on a hypothetical system directly. This enables experiments that are otherwise difficult to carry out in practice, such as that involving very high temperatures and pressures, be investigated theoretically. In this work phase transition of a Lennard-Jones fluid system following Lennard-Jones potential is studied through the use of temperature quench molecular dynamics (TQMD) method. TQMD is a method of locating fluid phase equilibria by means of a canonical ensemble (i.e., system at constant temperature) simulation. It involves quenching of an initially homogeneous one-phase fluid to a much lower temperature at which it is thermodynamically and mechanically unstable. The quenching will result in spinodal decomposition, leading to the formation of coexisting phase domains after a short transient. The domains will quickly acquire equilibrium-like properties, allowing the coexisting domains to be post-analyzed using local density or other suitable order parameters. In the TQMD approach, contrary to usual expectations, one does not need to wait for a planar interface to be formed during a full global equilibration. Only local equilibration is needed. We have tested TQMD method on a pure cut and shifted Lennard-Jones fluid, followed by a post-analysis method of histogramming of local density. The results are compared to the popular Gibbs Ensemble Monte Carlo (GEMC) method. Two sets of data using different total simulation time of 120000 time steps and 330000 time steps respectively are used to verify that local equilibration is indeed sufficient to attain the equilibrium-like values. The critical temperature and critical density are found to be, in reduced unit, $T_c^* = 1.2896 \pm 0.0069$ and $\rho_c^* = 0.313224 \pm 0.0013$ respectively. The data are then compared with that of noble gases (neon, argon, krypton, xenon). We found good agreement between the experimentally measured critical values of real noble gasses with our calculation.

We have used Mathematica for the visualisation of the output. Specifically, Mathematica is used to investigate the relation between various state variables. Radial distribution function

(RDF) is used to view the structural information of the system, i.e., whether it is in solid, liquid or gaseous state. The variation of pressure and compression factor with respect to volume for real gases at various temperatures and volumes are evaluated. It is shown that the system behaves like an ideal gas at high temperature and high volume (low pressure), which follows the known variation of a real gas. We conclude that the TQMD is particularly suitable to the determination of phase coexistence of an unfamiliar system. TQMD is a viable alternative to the existing phase transition related method especially GEMC, which is the most popular method.

Abstrak

Simulasi molekul adalah satu pendekatan yang mampu mengkaji reaksi dan sifat-sifat sistem dengan menghubungkan sifat makroskopik sistem dengan butir-butir molekul dan interaksi zarah mereka. Pendekatan komputasi bertindak sebagai medan yang menghubungkan pendekatan teori dan eksperimen, membolehkan ujian teori dilakukan ke atas system hipotesis secara langsung. Ini membolehkan eksperimen yang sebaliknya sukar diamalkan, seperti yang melibatkan suhu dan tekanan yang sangat tinggi, disiasat dari segi teori. Dalam kertas ini, fasa peralihan sistem cecair Lennard-Jones yang mengikuti potensi Lennard-Jones dikaji dengan kaedah "temperature quench molecular dynamics" (TQMD). TQMD ialah satu kaedah mencari keseimbangan fasa cecair melalui simulasi ensemble berkanun (iaitu, sistem pada suhu tetap). Ia melibatkan pelindapkejutan daripada cecair seragam satu fasa kepada suhu yang lebih rendah di mana ia tidak stabil dari segi termodinamik dan mekanikal. Pelindapkejutan akan menyebabkan penguraian spinodal, membawa kepada pembentukan fasa domain bersama selepas satu fasa pendek. Domain akan cepat memperoleh sifat keseimbangan, membolehkan domain bersama dianalisis dengan menggunakan ketumpatan tempatan atau parameter yang sesuai. Dalam pendekatan TQMD, bertentangan dengan jangkaan biasa, seseorang tidak perlu menunggu sehingga muka satah dibentuk semasa proses imbang global. Hanya proses imbang tempatan diperlukan. Kami telah menguji kaedah TQMD pada cecair Lennard-Jones tulen yang dipotong and dipindah, diikuti dengan kaedah histogram ketumpatan tempatan. Keputusan dibandingkan dengan kaedah "Gibbs Ensemble Monte Carlo" (GEMC) yang popular. Dua set data yang menggunakan masa simulasi yang berbeza ialah sebanyak 120000 langkah masa dan 330000 langkah masa masing-masing digunakan untuk mengesahkan bahawa imbang tempatan memang mencukupi untuk mencapai nilai-nilai keseimbangan. Suhu kritikal dan ketumpatan kritikal adalah, dalam unit kurangan, $T_c^* = 1.2896 \pm 0.0069$ dan $\rho_c^* = 0.313224 \pm 0.0013$ masing-masing. Data ini kemudiannya dibandingkan dengan gas mulia (neon, argon, kripton, xenon). Kami mendapati konsistensi yang baik antara nilai kritikal gas mulia sebenar dengan pengiraan kami.

Kami telah menggunakan Mathematica untuk visualisasi output. Khususnya, Mathematica digunakan untuk menyiasat hubungan antara pelbagai pembolehubah keadaan. Fungsi taburan jejari (RDF) digunakan untuk memerhati maklumat struktur sistem, iaitu, sama ada dalam keadaan pepejal, cecair atau gas. Perubahan tekanan dan faktor pemampatan dengan isi padu gas benar dinilai pada pelbagai suhu dan isi padu. Ia menunjukkan bahawa sistem bertindak seperti gas ideal pada suhu dan isi padu yang tinggi (tekanan rendah), yang mengikuti perubahan yang diketahui bagi gas benar. Kami membuat kesimpulan bahawa TQMD ialah kaedah yang sangat sesuai untuk menentukan fasa satu sistem yang tidak dikenali. TQMD adalah alternatif yang mampu bersaing dengan kaedah yang sedia ada terutamanya GEMC yang merupakan kaedah yang paling popular.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Objective of report and Problem Statement

The goal of this work is to investigate the phenomenon of phase transition and properties of Lennard Jones fluids using temperature quench molecular dynamics (TQMD) technique. Although Lennard Jones fluid has been extensively investigated over the past few decades, it still retains its significance as a popular computational model due to simplicity and versatility.

A quick browse through of the published journal will reveal that most the previous research is conducted using Gibbs Ensemble Monte Carlo (GEMC) method, which has become the de facto simulation method for generating phase diagrams for fluid systems. However the GEMC method is not without limitations according to Panagiotopoulos (1997) [19], which will be further discussed in the literature review section along with the other methods.

The other major molecular simulation technique is the molecular dynamics method. However, the conventional molecular dynamics technique has several critical disadvantages, which caused it to be superseded by GEMC method, being costly from computational point of view and time consuming, according to F. Martínez-Veracoechea & E.A. Müller (2005) [10].

The TQMD method is hence proposed as a potential alternative to study phase transition properties of a system. This technique has been evaluated in Gelb and Müller (2002) [11] and to F. Martínez-Veracoechea & E.A. Müller (2005) [10]. TQMD involves locate phase coexistence points using molecular dynamics (MD) simulations by quenching the system into a two-phase region. It effectively overcomes the difficulties faced by the other two techniques. It can be applied without difficulties to very complex molecules and mixtures, and mostly importantly can be implemented on large parallel computers.

This report evaluates the application of TQMD method on pure or single component Lennard Jones fluid. The implementation process is explained in details in the methodology section. Further investigation of properties of Lennard Jones fluid on various state points of the phase diagram is done using Mathematica software [27], by observing the radial distribution function and the variation in density and compression factor with respect to temperature.

1.2 Background

The advent of computer technology with more powerful processor during recent decades has made it feasible to simulate the dynamics of molecular systems on a computer. The method of molecular dynamics (MD) solves Newton's equations of motion for a molecular system, which results in trajectories for all atoms in the system. From these atomic trajectories a variety of properties can be calculated.

The main aim of computer simulations of molecular systems is to compute macroscopic behaviour from microscopic interactions of particles in a given system. Hence a computer simulation has many benefits to offer: (1) to predict experimental observables, (2) To validate models of systems which predict observables and (3) to refine models and understanding of systems [1].

An atomic-level modelling of a system is done due to [1]:

1) Analytical solution of statistical thermodynamic equations is impossible.
2) Numerous parameters for interatomic interactions signify large numbers of degrees of freedom.
3) Able to follow Newton’s equations of motion or perform statistical sampling (Monte Carlo) which satisfies statistical thermodynamics.
4) From sampled conformations, calculate observables for comparison to experiment.

Molecular dynamics simulation is only a member of the big family of computational modelling. A brief chart of atomistic computational approaches is given in Figure 1.1. It is evident that molecular dynamics method is only applicable to simulation of a classical system, together with another stochastic approach called Monte Carlo. The main difference between molecular dynamics and Monte Carlo method are that their different approach towards simulating a system, being that molecular dynamics follows a deterministic process whereas Monte Carlo constitutes a random trajectory for the particles. The main idea of molecular dynamics is to determine the path of a particle by solving the Newton’s equations of motion. Monte Carlo method focuses on the respective transition probability of a particle to take a certain path. Additional materials on the Monte Carlo can be found in the article by Panagiotopoulos (1997) [19].

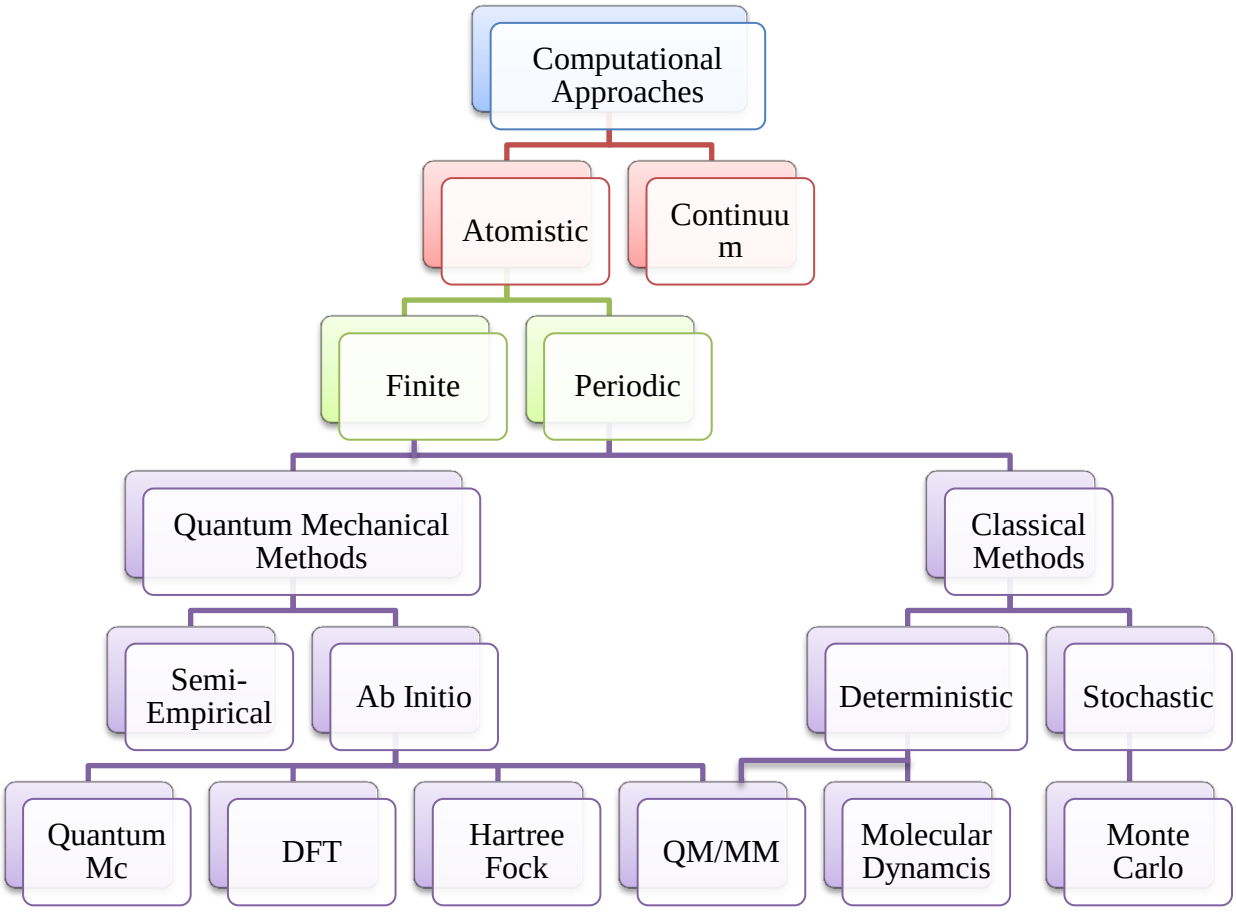


Figure 1.1
Hierarchy chart of computational approaches [21]

There exist two basic problems in the field of molecular modelling and simulation. One is to efficiently search the vast configuration space spanned by all possible molecular conformations for the global low energy regions, populated by a molecular system in thermal equilibrium. The other problem is the derivation of a sufficiently accurate interaction energy function or force field for the molecular system of interest. Hence, an important part of the art of computer simulation is to choose the unavoidable assumptions, approximations and simplifications of the molecular model and computational procedure such that their contributions to the overall inaccuracy are of comparable size, without affecting significantly the property of interest [24].

There is a variety of molecular models and force fields, differing in the accuracy by which different physical quantities are modelled. The choice of a particular force field will depend on the property and level of accuracy one is interested in. In short there are 3 factors that should be considered when studying a molecular system by computer simulation (see Figure 1.2) [24]:

1) The properties of the molecular system one is interested in should be listed and the configuration space (or time scale) to be searched for relevant configurations should be estimated.
2) The required accuracy of the properties should be specified.
3) The available computer time should be estimated.

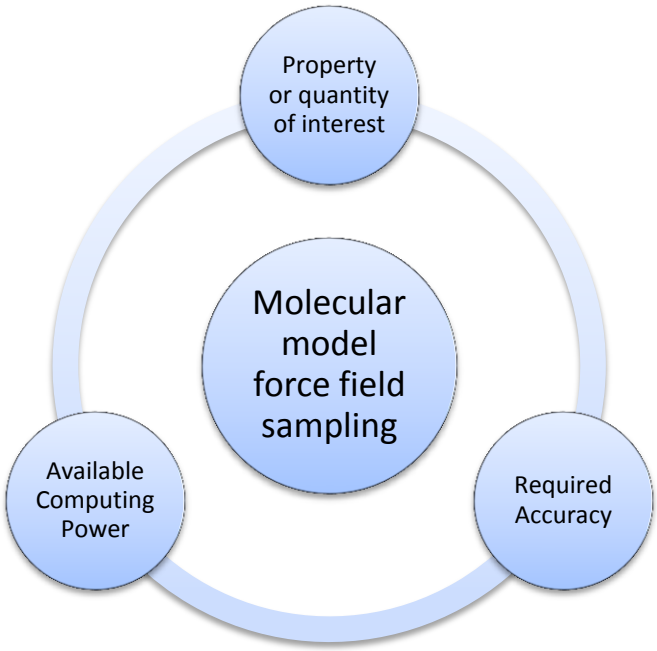


Figure 1.2
Factors affecting choice of molecular model, force field and sample size [24]

There will be a trade-off between accuracy of the force field on the one hand and the searching power or time scale that can be attained on the other. After considering all these factors, Lennard-Jones potential is selected for our work for reasons that will be explained in the following sections.

The rapid increase in computing power over the last few decades has drastically diversifying the possibility of simulating real system using computer. However, human technology seems

to be still at a loss compared to Mother Nature. According to Gunsteren and Berendsen (1990) [24], even a supercomputer is still many orders of magnitude slower than nature: a state of the art simulation is about 10^{15} times slower than nature, at least at their time of publication. The sampling configuration of nature, which is of the order of Avogadro numbers, is too much for our current technology to handle. Yet they predicted that with the current growth rate, the speed of simulation of a supercomputer will have caught up with that of nature in about 80 years.

1.3 Acknowledgement of the Previous Work

Several prominent and influential works in this field had been done as early as in 1969 by Hansen and Verlet in their paper "Phase transitions of the Lennard-Jones System" [16]. The work done by Verlet (1967) is considered as one of the earliest computer simulation on classical fluids [25].

Theoretical aspect of this field has been studied through the construction of equation of state in the influential work of Nicolas et al.(1979) [14]. The method of Nicolas et al. has been revised and improved by Johnson et al.(1993) [15], which since become the main theoretical tool to verify simulation data obtained.

Molecular simulation of fluid phase equilibria has studied and published several times, including Smit (1996) [5], Kai Gu et al.(2010) [13], Matsumoto (1998) [17]. The basic reviews of molecular simulation method, including molecular dynamics (MD) and Monte Carlo (MC), are done by Bopp et. al (2008) [20]. On the other hand, Panagiotopoulos (2000) explored the Monte Carlo methods available for phase equilibria of fluids [19].

As stated previously, the main scope of this work is based on the work of Gelb and Müller (2002), "Location of phase equilibria by temperature-quench molecular dynamics simulations" [11] and Veracoechea and Müller (2005), "Temperature-quench Molecular Dynamics Simulations for Fluid Phase Equilibria" [10]. The details of all these articles will be discussed in the literature review section.

1.4 Scope and Content of this work

This work will be focusing on location of phase equilibria of single component Lennard-Jones fluid by TQMD method. Although TQMD method is perfectly capable of stimulating a more complex system such as binary component or mixture system, current work is only limited to pure LJ fluid due to limited resources and time. The current code of TQMD method is sequential, despite molecular dynamics strength lies in its capabilities to be parallelized, as the objective of this work is to show a viable alternative to GEMC method to study phase coexistence, thus verifying the work of Gelb and Müller (2002) [11] and Veracoechea and Müller (2005) [10].

The literature reviews section of this work is described in Section 2, briefly discussing the current development of this field, ranging from typical conventional methods to the more recent methods. Section 3 is the theory section, describing the basic theoretical foundation of molecular dynamics technique. Lennard-Jones potential will be discussed in details and certain nominations and terms in this field will be discussed. This section will essentially cover what one needs to know about basic molecular simulation.

Section 4 will be the important methodology section, involving detailed description of the TQMD method. The theoretical ground of TQMD method will be explored and additional materials such as the procedure and possible errors will be given. The main purpose of this section is to allow the readers to access to assess the believability of our results. It should be possible for a competent physicist to reproduce our results by following our description.

The results of the simulations will be displayed at the Section 5, the Results and Discussion section. The data will take the form of graph and table, and the associated errors will be indicated. Complete sets of simulation data will be include in a disc attached with this work. The discussion in this section will be summarising our results. Major patterns, relationships and generalization of the results will be discussed. Degree of agreement to the previous works and significance of the results are explained in details.

A concluding remark is included in Section 6 the Conclusion and Recommendation section. Recommendations about possible future extension of this research are suggested. The Reference in Section 7 lists all works and articles cited by this work. Computer program and coding will be included in the Appendix and also recorded in a disc.

2. Literature Review

2.1 Classical simulation of Lennard-Jones system

Hansen and Verlet (1969) [16] had published their work in this field as early as forty years ago. Monte Carlo method is used to determine the phase transitions of a system of particles interacting through Lennard-Jones potential. A method has been devised to force the system to remain always homogeneous. The equation of state of the liquid region was obtained for the reduced temperature T = 1.15 and T = 0.75 by a standard Monte Carlo calculation. A system of 864 particles with periodic boundary condition is used. In the gas region the equation of state can be obtained from the virial expression. It is found that the coexistence curve for argon is flatter in the critical region than the one deduced from machine computation. This is due to the long-range density fluctuation, responsible for the peculiar singularity near critical point, cannot be included in the Monte Carlo calculation. Transition density for the liquid branch is also showing a better agreement between theory and experiment than in the case of gas. The discrepancy is expected as properties of dilute argon are poorly accounted for by the Lennard-Jones potential. The fit of results from an approximate equation of state is better for Lennard-Jones case is than that of argon. The phase transition of Lennard-Jones fluid can be calculated using methods where only homogeneous phase are considered.

Verlet (1967) [25] had done one of the earliest computer simulations on classical fluids to study the thermodynamics properties of Lennard-Jones molecules. The equation of motion of 864 particles following Lennard-Jones potential had been integrated for various values of temperature and density. Various equilibrium thermodynamics quantities are calculated and the agreement with the corresponding properties of argon was good. A system of 864 particles enclosed in a cube of side L, with periodic boundary condition, is allowed to interact through a two-body Lennard-Jones potential. The overall agreement between theory and experiment is surprisingly good; Lennard-Jones is a quite satisfactory interaction as far as the equilibrium properties are concerned. However, it is stated that the agreement is not good if xenon is compared instead for argon.

2.2 Derived equation of state

Nicolas et al.(1979) [14] had explored the theoretical aspect of this field through the construction of equation of state. Molecular dynamics calculations of the pressure and configurational energy of a Lennard-Jones fluid are reported for 108 state conditions in the density range $0.35 \leq \rho^* \leq 1.20$ and temperature range $0.5 \leq T^* \leq 6$. Simulation results of pressure and configurational energy, together with low density values calculated from the virial series and value of second virial coefficients, are used to derive an equation of state for the Lennard-Jones fluid that is valid over a wide range of temperatures and densities. The equation of state used is a modified Benedict-Webb-Rubin (MBWR) equation having 33 constants. In fitting the equation of state the virial series at low densities and computer simulation results at the higher densities. The MBWR equation has only one non-linear parameter for numerical convenience. In fitting the data a relatively small weighting procedure was used. The gas-liquid coexistence curve calculated from the obtained equation of state was compared with the existing literatures at that time and the agreement is good. However it is especially stated that the equation of state should not be used at state conditions outside of the region of fit, otherwise it gives significant errors if used to extrapolate to low temperatures.

The method of Nicolas et al. [14] has been revised and improved by Johnson et al.(1993) [15], which since become the main theoretical tool to verify simulation data obtained. New parameters for the modified Benedict-Webb-Rubin (MBWR) equation of state used by Nicolas et al. [14] are presented. In contrast to previous equations, the new equation is accurate for calculations of vapour-liquid equilibria, accurately correlating pressures and internal energies from the triple point to about 4.5 times the critical temperature over the entire fluid range. The constrains for the critical point, which the values are taken by other literatures, are used in the regression of parameters, together with the second virial coefficients and using weighting of the uncertainties of pressures and internal energies. However, it is found that the equation of state is not capable of fitting both the vapour-liquid region and high temperature region with comparable accuracy. In comparing the predicted vapour liquid equilibrium data, the new parameters are significantly more accurate for $T^*>1$. Although the new equation is more accurate than that of Nicolas et al. [14] for VLE calculations, the accuracy of the new equation is somewhat lower for dense fluids at temperatures greater than twice the critical.

2.3 Molecular simulation applications

Smit (1996) [5] reviews some applications of molecular simulations of phase equilibria. In order to shorten the require cpu-time, it is important to either simplify the models or to develop novel simulation techniques. In particular for phase equilibrium calculations the slow equilibration of complex fluids limits the range of applications of molecular simulations. For dipolar fluids, the simplest model is the dipolar hard sphere. In the Gibbs-ensemble technique, simulations of the vapour and liquid phase are carried out in parallel. Monte Carlo moves ensures the thermodynamics equilibrium of the two boxes, and the coexistence densities can be determined directly from the two systems. Surprisingly phase separation for the dipolar hard-sphere fluid is not detected. This implies that phase separation occurs at conditions that different than expected from the statistical mechanical theories. A model polar fluid with dispersive Lennard-Jones interactions is used instead. At conditions where the coexistence curve is expected formation of chains of dipoles aligning nose to tail are observed instead. The formation of chains inhibits the phase separation, which is the reason vapour-liquid coexistence is not detected in the original simulations of the dipolar hard-sphere fluid. These simulation results show that a minimum amount of dispersive energy is required to observe liquid-vapour coexistence in a dipolar fluid. The author concluded that dipolar hard-sphere fluid is not a good starting point to develop a theory for real polar fluids. In real polar fluids the dispersive interactions are essential to stabilize the liquid phase.

In Kai Gu et al. (2010) [13], the equilibrium structure of the finite, interphase interfacial region that exists between a liquid film and a bulk vapour is resolved by molecular dynamics simulation. Argon systems are considered for a temperature range that extends below the melting point. Physically consistent procedures are developed to define the boundaries between the interphase and the liquid and vapour phases. The procedures involve counting of neighbouring molecules and comparing the results with boundary criteria that permit the boundaries to be precisely established. Definitions of both interphase boundaries are necessary to collect molecular mass flux statistics for computation of interfacial mass transfer in MD simulations. The interphase thicknesses determined from the new boundary criteria are more precise. At points away from the melting point, the results are in better agreement with transition state theory. Near the melting point, transition theory approximations are less valid. These are resulted from the application of new criteria for interphase boundaries to the MD computation of condensation and evaporation coefficients.

In Matsumoto (1998) [17], molecular dynamics simulation has been applied for various fluid systems to investigate microscopic mechanisms of phase change. This covers a review of the works done by Matsumoto's group. Evaporation–condensation dynamics of pure fluids under equilibrium condition are investigated to study the dynamic behaviour of molecules under such condition. By analyses of molecular trajectories, dynamic behaviour of molecules near a liquid surface is found to be classified into four categories: evaporation, condensation, self-reflection and molecular exchange. The 4^{th} type, molecular exchange, becomes quite important for some cases such as associating fluids and fluids at high temperatures. For evaporation–condensation dynamics of pure fluids under non-equilibrium condition, the behaviour is more complicated. Even with the liquid temperature given, there are two more control parameters: the temperature and the density (or the pressure) of the vapour. The situations include hot vapour condensation on cool liquid and evaporation into vacuum. The last part involves gas absorption dynamics: CO_2 gas absorption mechanism on water surface is analysed from the view point of adsorption–desorption dynamics. The ions tend to avoid the surface whereas CO_2 molecules are strongly adsorbed on the surface where little ions exist.

2.4 Scope and methods of molecular simulations

Scope and limits of molecular simulation are discussed in Bopp et. el (2008) [20]. The method of molecular dynamics and Monte Carlo are discussed. The basic concepts of molecular simulation are introduced. This is good introductory article of the field of molecular simulation, along with some previous works of the authors presented. The difference between Monte Carlo (MC) and molecular dynamics (MD) methods lies in the way the sample is generated. MC uses a random walk procedure and accepts or rejects displacement of a particle based on its transition probabilities. MD is a deterministic method, where the main idea is the integration of Newton's equation. One of the works discussed is a model of liquid-liquid interface, in which the miscibility of two types of particles is determined by the radii of particles and the strength of interactions between like and unlike particles. It shows that once the plane interfaces separating two different particles are formed, the system remains stable for the duration of the simulation.

On the other hand, Panagiotopoulos (2000) [19] explored the Monte Carlo methods available for phase equilibria of fluids. The Gibbs ensemble method and histogram-reweighting Monte Carlo techniques are described in detail. The Gibbs ensemble method is based on simulations of two regions coupled via volume change and particle transfer moves so that the conditions for phase coexistence are satisfied in a statistical sense. Histogram-reweighting methods obtain the free energy of a system over a broad range of conditions from a small set of grand canonical Monte Carlo calculations. Other methods described briefly include interfacial simulations, the NPT + test particle method, Gibbs-Duhem integration and pseudoensembles. In order to o increase the efficiency of the simulations, configurational-bias sampling techniques and expanded ensembles can be used.

2.5 Temperature Quench Molecular Dynamics

Work of Gelb and Müller (2002) [11], evaluates the usefulness of TQMD as an alternative to the existing methods. It is stated that this method can be used to locate vapour–liquid, liquid–liquid or solid–fluid equilibria. The method is demonstrated on test systems of single component and binary Lennard–Jones (LJ) fluids. The key concept to this method is the spinodal decomposition of the system when the system is suddenly made to an unstable state. The result of single component LJ fluid follows the expected equation of state of Johnson et al. [13] for the full potential since the cutoff radius is relatively large, verifying the use of the method. It is stated that TQMD is not limited to fluid phase equilibria. If the final quench temperature is below the triple point, solid phases can nucleate during the quenching process.

Veracoechea and Müller (2005) [10] is a more detailed application of method outlined by Gelb and Müller (2002) [11]. They particularly analyse the short-time phase separation behaviour of fluids, as well as present some example applications to the vapour–liquid equilibria of a pure LJ fluid, the liquid–liquid–vapour equilibria of a binary LJ system, and the saturation densities of a long-chain alkane. The results are found to similar to simulations using a much higher number of particles and long simulation times, indicating the advantages of this method. The results obtained by this method are also shown to be of the same precision as those obtained by GEMC or volume expansion molecular dynamics (VEMD).

3. Theory

3.1 Lennard-Jones Potential

Proposed by Sir John Edward Lennard-Jones, the Lennard-Jones Potential is a mathematical approximation that illustrates the energy of interaction between two nonbonding atoms or molecules based off their distance of separation. Due to its computational simplicity, the Lennard-Jones potential is used extensively in computer simulations even though more accurate potentials exist. In fact, it has been extensively studied over the decades. Apart from being an important model in itself, the Lennard-Jones potential frequently forms one of 'building blocks' of many force fields.

The Lennard-Jones potential is given by [8]:

$$u(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

where

- r = distance between particles
- V_{LJ} is the intermolecular potential between two particles or sites
- σ is the finite distance at which the inter-particle potential is zero
- ε is the depth of the potential well.

For large separations, the interaction is due to the Van der Waals force, which is a weak attraction arising from the transient electric dipole moments of the atoms. This potential is denoted by $\left(\frac{\sigma}{r}\right)^{12}$. When the atoms get close together, there is a repulsive force due to the overlap of their electron clouds. This potential is denoted by $\left(\frac{\sigma}{r}\right)^{6}$. Adding this to the Van der Waals component yields Lennard-Jones Potential.

The graph of Lennard Jones potential versus interatomic distance is displayed in Figure 3.1.

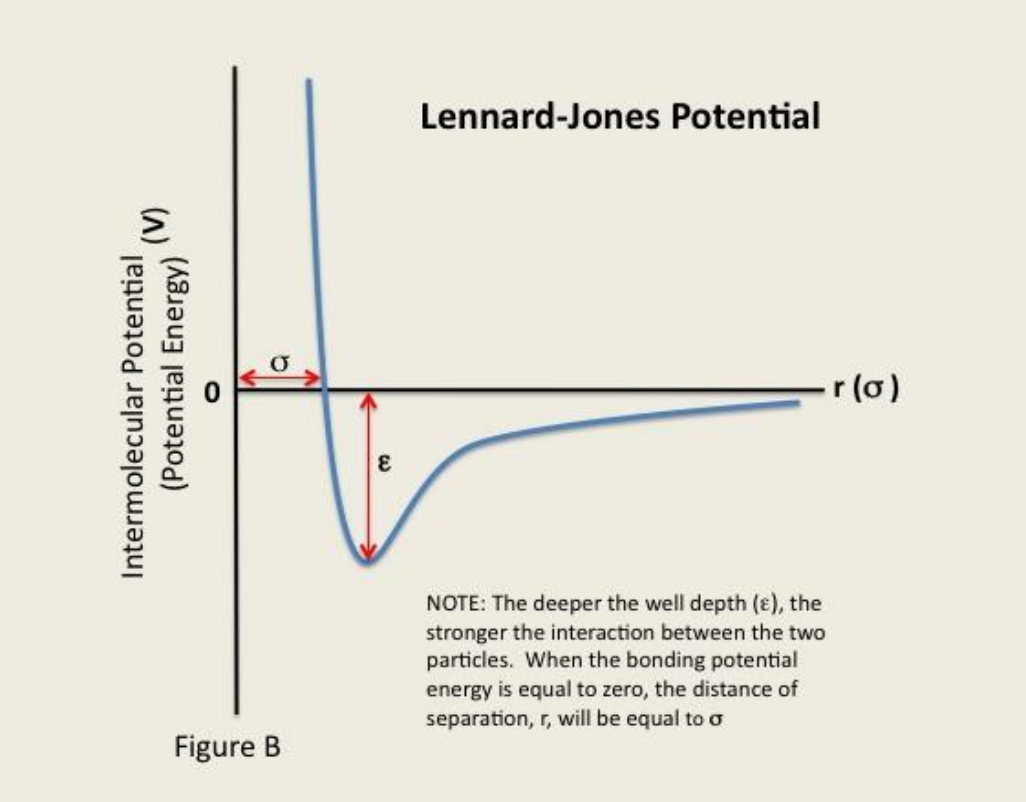


Figure 3.1

A graph of potential versus interatomic distance for the 12-6 Lennard Jones potential [23]

3.2 Reduced Units

In simulations it is often convenient to express quantities such as temperature, density, pressure in reduced units. A convenient unit of energy, length and mass is chosen and then all other quantities are expressed in terms of these basic units. These reduced units are usually denoted with superscript *. The common reduced units used are shown in Table 3.1.

The reduced form for the Lennard-Jones potential is:

$$u^*(r^*) = 4\left[\left(\frac{1}{r^*}\right)^{12} - \left(\frac{1}{r^*}\right)^{6}\right]$$

The most important reason to introduce reduced units is that many combinations of σ, ϵ, and ρ all correspond to the same state in reduced units. This is the law of corresponding states. Another reason is that reduced units make it easier to spot errors, since almost all quantities of interest are of order 1.

Simulation results that are obtained in reduced units can always be translated back into real units. Comparison of the results of a simulation on a Lennard-Jones model with experimental data for argon ($\epsilon/k_B = 119.8$ K, $\sigma = 3.405 * 10^{-10}\ m$, $M = 0.03994\ kg/mol$) can be done using the translation given in Table 3.2.

Table 3.1: Reduced Units for molecular simulation [8]

Reduced Units	In terms of full units
Lengths, L*	L/σ
Energy, U*	U/ϵ
Mass, m*	m/m
Time, t*	t/$\left(\sigma\sqrt{m/\epsilon}\right)$
Temperature, T*	k_BT/ϵ
Pressure, P*	Pσ^3/ϵ
Density, ρ*	$\rho\sigma^3$

Table 3.2: Translation of reduced units to real units for Lennard-Jones argon [8]

Quantity	Reduced units	Real units
temperature	$T^* = 1$	$T = 119.8\ K$
density	$\rho^* = 1.0$	$\rho = 1680\ kg/m^3$
time	$\Delta t^* = 0.005$	$\Delta t = 1.09 * 10^{-14}\ s$
pressure	$P^* = 1$	$P = 41.9\ MPa$

3.3 Periodic Boundary Condition

However, the number of atoms in computer simulation is far too less compared to those of thermodynamics limits. In small systems, the collisions with the walls can be a significant fraction of the total number of collisions, in contrast to real system. In order to simulate bulk phases it is essential to choose boundary conditions that mimic the presence of an infinite bulk surrounding our *N*-particle model system. Hence the periodic conditions are applied. The volume containing the *N* particle is treated as the primitive cell of an infinite periodic lattice of identical cells, as shown in Figure 3.2. A given particle now interacts with all other particles in this infinite periodic system, that is, all other particles in the same periodic cell and all particles (including its own periodic image) in all other cells.

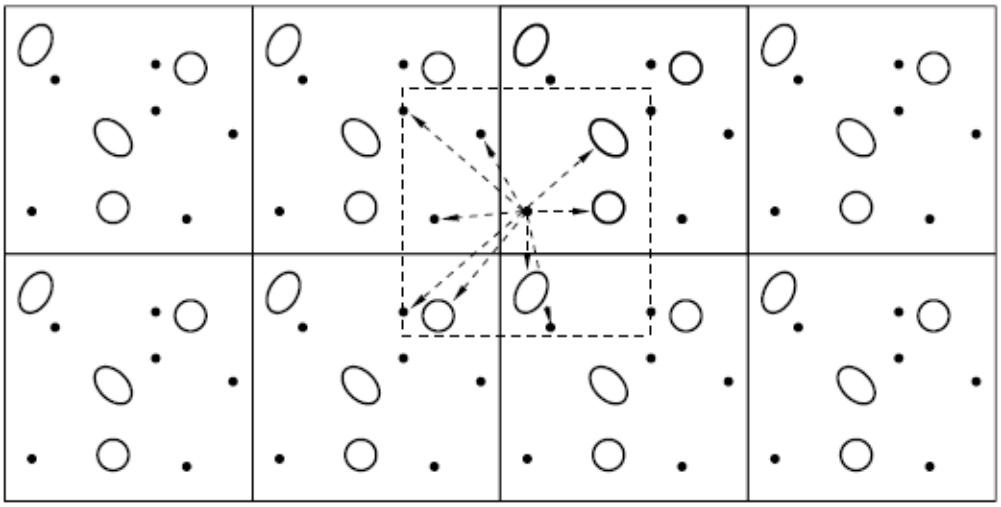


Figure 3.2
Schematic representation of periodic boundary conditions. [8]

The implementation of periodic condition in the program is illustrated in Figure 3.3. Whenever an atom encounters a wall, it is transported instantly to the opposite side of the system. This effectively avoids all the collisions with the walls, stimulating a large real system.

The "Minimum separation rule" takes advantage of the teleportation of periodic boundary condition. This rule acts as a precautionary step when considering relative positions of the particles. The particles should only be allowed to interact only once with a particular particle,

either with the real particle in the box or its mirror image. It directly implies that smaller separation is used to calculate the magnitude and direction of the force.

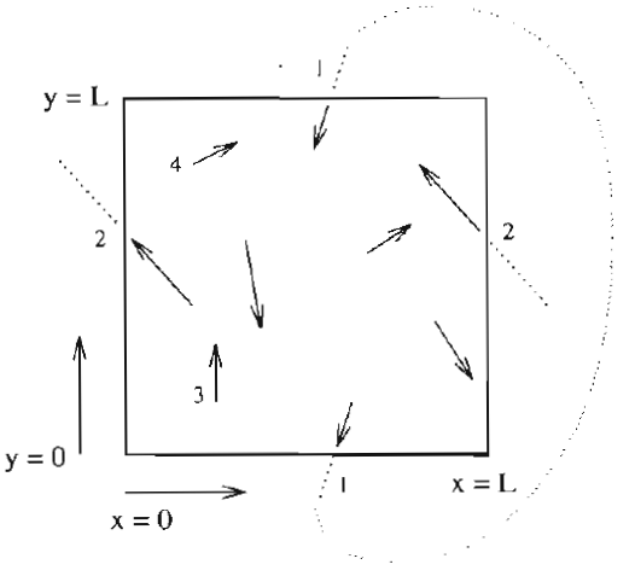


Figure 3.3

Periodic boundary condition for a molecular dynamics simulation using a $L \times L$ box. [12]

The arrows denote atoms and their velocities.

3.4 Molecular Simulation

The fundamental idea underlying all molecular simulations is simple and follows directly from Boltzmann's thinking about the ''thermodynamic ensembles'' [20]:

1) Simulation: Construct a sufficient number of microscopic configurations, or states, compatible with:
    - First, the macroscopic thermodynamic constraints (temperature, density, etc.) of the system under consideration,
    - Second, the intermolecular (or interatomic) interactions in the system.
2) Evaluation: Use statistical tools to compute averages over these configurations.

The two simulation methods alluded to above criteria, molecular dynamics (MD) and Monte Carlo (MC), differ in the way the sample is generated. While MD uses, as Boltzmann envisaged, classical (Newtonian) mechanics, MC rests on a random walk procedure. In both

cases, however, the model describing the interactions between the particles in the system is the critical input to any simulation.

The two methods do not yield the same amount of information about the system: MD, being based on Newton's equation, samples the "phase space" of the system. A phase space contains all positions and all momenta of all particles in the system at a given time *t*. The sample of the ensemble constructed thus contains information about the time evolution of the system. MC samples ''configurational space", concerning only information about the particle positions.

The overall idea of molecular simulation is shown in Figure 3.4.

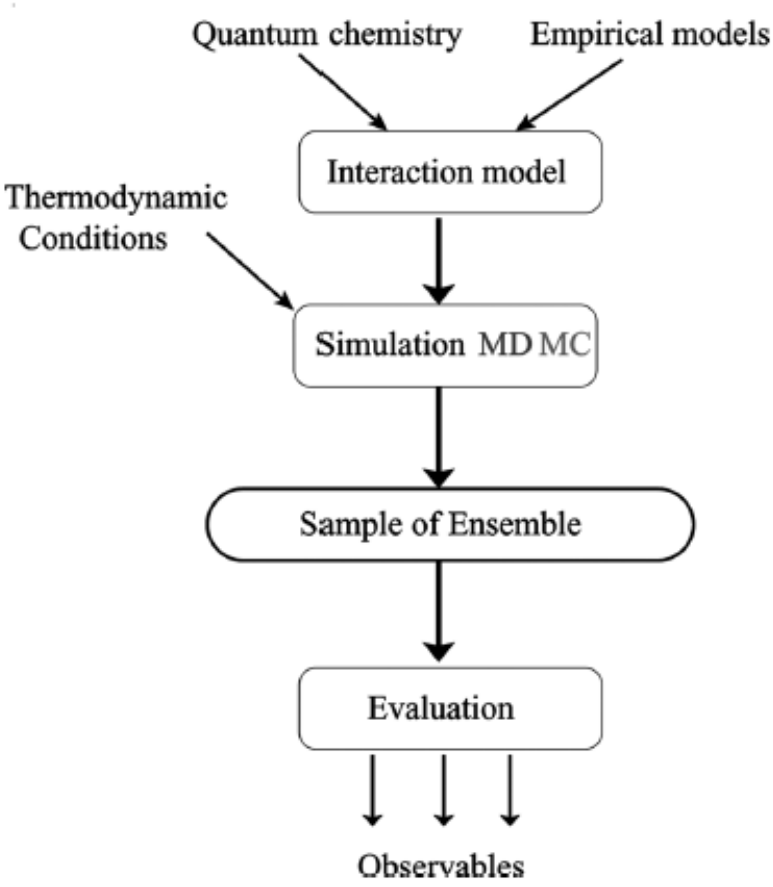


Figure 3.4
Schematic view of molecular simulations [20]

The challenge in constructing an ensemble is to construct a sufficient number of microscopic configurations, or micro-states, compatible with the interactions and the macroscopic constraints. Only systems with a fixed number of particles N in a fixed volume V with

periodic boundary conditions will be considered here. The configurations can be constructed by the following ways [20]:

- In MD by solving numerically Newton's equations of motion for all particles under the influence of their mutual forces, computed from the interaction model, for a very small time step δt, (typically δt = a few $10^{-15} – 10^{-16}$ seconds, depending on the masses and interactions),
- In MC through a random walk procedure (obeying the so-called ''detailed balance condition'') in which new configurations are added to the ensemble according to predefined probabilities (usually a Boltzmann factor $\exp(\Delta V/k_B T)$, where V is the potential energy from the interaction model, T the desired temperature, and k_B Boltzmann's constant).

The ensembles are constructed through iterations of these procedures. In summary, MD samples a microcanonical ensemble whereas MC samples a canonical ensemble. After the simulation steps, observables can be computed from these ensembles.

3.5 Molecular Dynamics [6]

Molecular dynamics is mainly about solving Newtonian Mechanics and using numerical integration. The Newtonian equations of motion can be expressed as:

$$m\ddot{\boldsymbol{r}}_{\boldsymbol{i}} + \nabla_i U = 0$$

where $\ddot{\boldsymbol{r}}_{\boldsymbol{i}}$ is the acceleration of particle i and the force acting on particle i is given by the negative gradient of the total potential, U, with respect to its position:

$$\boldsymbol{f}_i = -\nabla_i U = -\frac{\partial U}{\partial \boldsymbol{r}_i}$$

It is suffice to consider a system with generic pairwise interactions, for which the total potential is given by:

$$U = \sum_{j} \sum_{k>j} u_{jk}(r_{jk})$$

where r_{jk} is the scalar distance between particles *j* and *k*, and u_{jk} is the pair potential specific to pair (j, k).

For a system of *N* particles, the force on any particular particle *i*,

$$\boldsymbol{f}_i = -\sum_{j=1}^{N} \frac{\partial u_{ij}(r_{ij})}{\partial \boldsymbol{r}_i} = \sum_{j=1}^{N} \boldsymbol{f}_{ij}$$

where $\boldsymbol{f}_{ij}$ is the force exerted on particle *i* by virtue of the fact that it interacts with particle *j*. The derivative can be broken up,

$$\boldsymbol{f}_{ij} = -\frac{\boldsymbol{r}_{ij}}{r_{ij}} \frac{\partial U_{LJ}(r_{ij})}{\partial r_{ij}}$$

As $\boldsymbol{r}_{ij} = -\boldsymbol{r}_{ji}$, the above equation illustrates that:

$$\boldsymbol{f}_{ij} = -\boldsymbol{f}_{ji}$$

This leads to a results that

$$\boldsymbol{F} = \sum_{i} \boldsymbol{f}_i = 0$$

That is, the total force on the collection of particles is zero. The practical advantage of this result is that we only need to calculate the force of a pair of particles once. Some refer to this as "`Newton's Third Law".

The other key aspect of a simple MD program is a means of numerical integration of the equations of motion of each particle. The first algorithm considered in D. Frenkel and B. Smit. (2002) [8] is the simple Verlet algorithm, which is an explicit integration scheme. Consider a Taylor-expanded version of one coordinate of the position of a particular particle,

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}(\Delta t)^2 + \frac{(\Delta t)^3}{3!}\dddot{r} + O[(\Delta t)^4]$$

By letting $\Delta t \to -\Delta t$,

$$r(t-\Delta t) = r(t) - v(t)\Delta t + \frac{f(t)}{2m}(\Delta t)^2 - \frac{(\Delta t)^3}{3!}\dddot{r} + O[(\Delta t)^4]$$

Adding both equations together,

$$r(t+\Delta t) \approx 2r(t) - r(t-\Delta t) + \frac{f(t)}{2m}(\Delta t)^2$$

This is the Verlet algorithm, introduced in Verlet (1967) [25] as discussed in literature review. If a small Δt is chosen, the position of particles at time $t + \Delta t$ can be predicted given its position at time t and the force acting on it at time t. The new position coordinate has an error of order $(\Delta t)^4$. Δt is the so called "time step" in a molecular dynamics simulation.

A system obeying Newtonian mechanics conserves total energy. For a dynamical system obeying Newtonian mechanics, the configurations generated by integration are members of the *microcanonical ensemble*; that is, the ensemble of configurations for which *NVE* is constant, constrained to a subvolume Ω in phase space.

When the Verlet algorithm is used to integrate Newtonian equations of motion, the total energy of the system is conserved to within a finite error, as long as Δt is small enough. Although velocities are not necessary in Verlet algorithm, they can be easily generated provided that one stores previous, current, and next-time-step positions in implementing the algorithm:

$$v(t) = \frac{r(t+\Delta t) - r(t-\Delta t)}{2\Delta t} + O[(\Delta t)^2]$$

There is some variants of the Verlet algorithm. Among them is the most popular integrator, the Velocity Verlet algorithm [8]. The current work in this report has used the velocity Verlet algorithm. Velocity Verlet algorithm requires updates of both positions and velocities:

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}(\Delta t)^2$$

$$v(t+\Delta t) = v(t) + \frac{f(t+\Delta t) + f(t)}{2m}\Delta t$$

The update of velocities uses an arithmetic average of the force at time t and $t + \Delta t$. This result in a slightly more stable integrator compared to the standard Verlet algorithm, in that one may use slightly larger time-steps to achieve the same level of energy conservation.

It is possible to manipulate the Verlet algorithm that two parallel force arrays are not necessary. The velocity update can be split to either side of the force computation, forming a so-called "leapfrog"' algorithm [8]:

i. Update positions

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}(\Delta t)^2$$

ii. Half-update velocities

$$v\left(t + \frac{\Delta t}{2}\right) = v(t) + \frac{f(t)}{2m}\Delta t$$

iii. Compute forces

$$r(t + \Delta t) \rightarrow f(t + \Delta t)$$

iv. Half update velocities

$$v(t + \Delta t) = v\left(t + \frac{\Delta t}{2}\right) + \frac{f(t + \Delta t)}{2m}\Delta t$$

This is the integrator that will be employed in this work.

Following the theorem of equipartition of energy, a working definition of instantaneous temperature, *T*:

$$\frac{3}{2}Nk_BT = \frac{1}{2}\sum_{i=1}^{N} m_i|\boldsymbol{v}_i|^2$$

For a microcanonical system the actual temperature is time average.

The pressure can be computed from

$$P = \rho k_B T + \frac{vir}{V}$$

where *V* is the system volume and *vir* is the virial:

$$vir = \frac{1}{3}\sum_{i>j} \boldsymbol{f}(r_{ij}) \cdot \boldsymbol{r}_{ij}$$

Summarizing all the above information, the important procedure of any MD program is [12]:

1. Read in the parameters that specify the conditions of the run (e.g., initial temperature, number of particles, density, and time step).
2. Initialize the system (select initial positions and velocities).
3. Compute the forces on all particles.
4. Integrate Newton's equations of motion. This step and the previous one make up the core of the simulation. They are repeated until the time evolution of the system is computed for the desired length of time.
5. After completion of the central loop, compute and print the averages of measured quantities, and stop.

3.5.1 Truncation of interactions

As mentioned by D. Frenkel and B. Smit. (2002) [8], the total potential (as a sum of pair potentials) in principle diverges in an infinite periodic system. This can be circumvented by introducing a finite interaction range to the pair potential. The error that results when interactions with particles at larger distances are ignored can be made arbitrarily small by choosing r_c to be sufficiently large. Due to periodic boundary condition, only the interaction of a given particle i with the nearest periodic image of any other particles j will be considered.

The major point is that the cutoff must be spherically symmetric [6]; that is, we can't simply cut off interactions beyond a box length in each direction, because this results in a directional bias in the interaction range of the potential. A hard cutoff radius, r_c, is hence required, and should be less than half a box length. If the intermolecular potential is not rigorously zero for $r \geq r_c$, truncation of the intermolecular interactions at r_c will result in a systematic error in the total potential energy.

Hence if one wishes to mimic a potential with infinite range, the correction terms for energy and pressure must be used. D. Frenkel and B. Smit. (2002) [8] show that, for Lennard-Jones pair potential:

i. Potential tail correction

$$u^{tail} = \frac{8}{3}\pi\rho\epsilon\sigma^3\left[\frac{1}{3}\left(\frac{\sigma}{r_c}\right)^9 - \left(\frac{\sigma}{r_c}\right)^3\right]$$

ii. Pressure tail correction

$$\Delta P^{tail} = \frac{16}{3}\pi\rho^2\epsilon\sigma^3\left[\frac{2}{3}\left(\frac{\sigma}{r_c}\right)^9 - \left(\frac{\sigma}{r_c}\right)^3\right]$$

There are a few methods to truncate the potential. Two of them will be discussed here:

i. Simple Truncation

The simplest method to truncate potentials is to ignore all interaction beyond r_c, the potential that is simulated is

$$u^{trunc}(r) = \begin{cases} u^{lj}(r), & r \le r_c \\ 0, & r > r_c \end{cases}$$

This method may result in an appreciable error in the estimate of the potential energy of the true Lennard-Jones potential. As the potential changes discontinuously at r_c, a truncated potential is not particularly suitable for a Molecular Dynamics simulation, but can be used for Monte Carlo simulations.

ii. Truncated and Shifted

This is the method employed in our work. It is a common procedure to be used in molecular dynamics simulations. The potential is truncated and shifted, such that the potential vanishes at the cut-off radius:

$$u^{tr-sh}(r) = \begin{cases} u^{lj}(r) - u^{lj}(r_c), & r \le r_c \\ 0, & r > r_c \end{cases}$$

There are no discontinuities in the intermolecular potential. The advantage of using such a truncated and shifted potential is that the intermolecular forces are always finite. This is essential as impulsive forces cannot be handled in those Molecular Dynamics algorithms that are based on a Taylor expansion of the particle positions to integrate the equations of motion.

3.5.2 Lennard-Jones potential in MD

Using reduced units and using truncated and shifted potential method, Lennard-Jones potential to be used is of the form:

$$U_{LJ} = 4\left(\frac{1}{r^{12}} - \frac{1}{r^{6}}\right) - U_{cut}$$

where U_{cut} is Lennard-Jones potential at cut-off.

$$U_{cut} = 4\left(\frac{1}{r_c^{12}} - \frac{1}{r_c^{6}}\right)$$

The force exerted on particle *i* by virtue of its Lennard-Jones interaction with particle *j* (in reduced units),

$$\boldsymbol{f}_{ij}(r_{ij}) = \frac{48\,\boldsymbol{r}_{ij}}{r_{ij}^2}\left[\left(\frac{1}{r_{ij}}\right)^{12} - \frac{1}{2}\left(\frac{1}{r_{ij}}\right)^{6}\right]$$

The virial is defined by:

$$vir = \frac{1}{3}\sum_{i>j} 48\left[\left(\frac{1}{r_{ij}}\right)^{12} - \frac{1}{2}\left(\frac{1}{r_{ij}}\right)^{6}\right]$$

4. Methodology

4.1 Overview

The parameter for a molecular dynamics simulation is chosen. For this work, the number of particle is set to 32000, overall density of particle is 0.328, where the volume of the containing cubic box is controlled by the former parameters and the cut-off radius is 5.0. The time step used is 0.004. Simulations are done at the chosen quenched temperature of 0.7, 0.8, 0.9, 1.0, 1.1, 1.2. Two sets of the simulations using the above parameters are done, one with total simulation steps of 120000 with equilibration after 70000 steps, another one with total simulation steps of 330000 with equilibration after 130000 steps. In order to increase the efficiency of force calculation algorithm, the neighbour cell algorithm is used instead of the usual pair interaction method [7].

The 32000 particle in a system is set to face-centred lattice. The algorithm used to generate a FCC lattice is inspired by that used by Thijssen in his example code for his book [22]. The velocity of each particle is then assigned according to the Boltzmann distribution function after setting the initial temperature.

The simulation is carried out as a *NVT* ensemble, using Berendsen thermostat to control the temperature. The system is first at a temperature higher than the critical temperature, $T = 4.0$ in our case, and then suddenly quenched to the above given temperature in a single time step. The particles will be separated into 2 distinct phases due unstable configuration of the system. The particles at the interface of both phases are then detected using interface detection algorithm and isolated out. The bulk liquid and gas phases are then obtained and their respective densities are calculated.

After simulations are carried out for the given set of quenched temperature, critical point of the phase diagram is found using the law of rectilinear diameter [9]. Agreement of the predicted critical point to the existing literature will be studied and discussed. The results of both sets of simulations with different simulation steps are also compared.

After TQMD simulations are finished, the Mathematica software is used to stimulate the Lennard-Jones particles under various state point of the phase diagram. Only 2000 particles will be used this time, as the objective is only to observe the variation in certain observables as we traverse through the phase diagram. Two set of simulation results are produced, one for NVT ensemble using Berendsen thermostat and another for NPT ensemble using Berendsen barostat [6].

4.2 Temperature Quench Molecular Dynamics (TQMD)

TQMD is a method for locating fluid phase equilibria by means of a single canonical molecular dynamics simulation [9]. By equilibrating a single phase in a relatively large simulation cell under *NVT* conditions and then quenching the system into a two-phase region, it spontaneously separates into two coexisting phases. A suitable analysis of the coexisting domains in terms of local densities, compositions, or some other order parameter gives the phase equilibrium properties. The method can be used to locate vapour–liquid, liquid–liquid or solid–fluid equilibria.

4.2.1 Method

A single component liquid–vapour system is considered, but the method is entirely general [9]. One starts out with a system composed of a given number of particles *N*, volume *V* and temperature *T*, large enough to guarantee a one-phase system. Keeping both *N* and *V* constant, the temperature, which is controlled by means of a "thermostat", is lowered abruptly. The situation is illustrated in Figure 4.1.

The initial state is a one phase system (point a). The quench target temperature (point b) must be such that the resulting state point lies within the spinodal envelope, e.g. at conditions that are both mechanically and thermodynamically unstable.

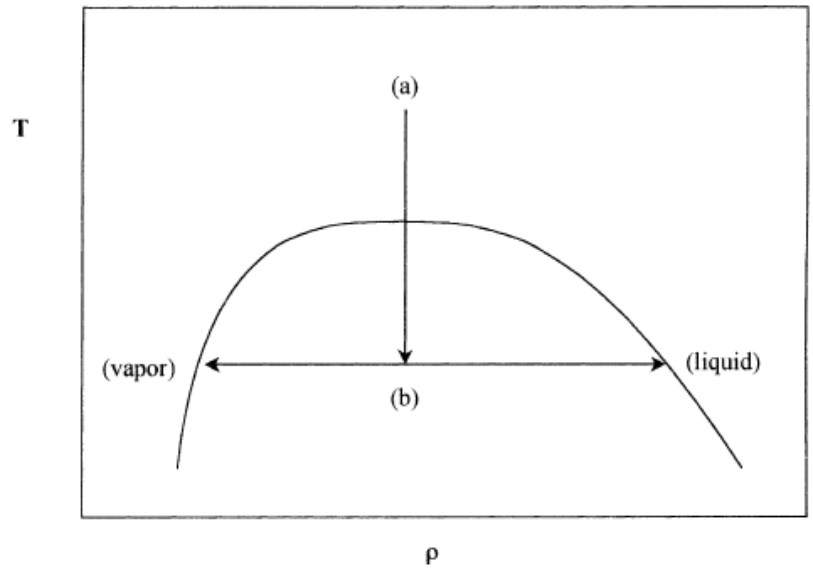


Figure 4.1
Temperature *T* vs. density *ρ* diagram for a pure fluid. [11]

After a short transient, domains of coexisting phases form, which quickly acquire equilibrium-like properties. The connectivity and morphology of these phases will depend on their volume fractions.

The system is far from equilibrium after quenching, thereby maximizing the driving force for diffusive transport. At short times the surface area between the two phases is very large, and the combination of these two factors ensures that local densities and concentrations stabilize at their equilibrium values very quickly. At later times, the reduction of surface tension between the two phases is the driving force for the next stage of separation of phases, which is thus rather slow. According to Gelb and Müller (2002) [11], careful equilibration at this temperature is not necessary, as no data is gathered at the initial point. The system equilibrates to domains separated by flat interfaces if enough time elapses.

4.2.2 Interface Detection

The obvious way for us to estimate the local equilibrium densities in a multiphase system is to wait until the system shows two distinct domains divided by flat interfaces, where the free energy of the system is at its lowest. According to Veracoechea and Müller (2005) [10], by choosing the simulation box in such a way that one of its axes is longer than the other two, at long simulation times the planar interface will form normal to the longer axis. The density profile along such axis can be fitted to a smooth stepwise function, thus allowing for the calculation of the bulk densities and the profiling of the interface.

This conventional method is of course time consuming, which is the downfall of the usual molecular dynamics simulation. The evolution of the system into global equilibrium can even pose a challenge even for today modern computer systems. This is where the advantage of TQMD comes in. The technique of TQMD enables the equilibrium property analysis may be performed much before the system attains global equilibrium, thus decreasing the necessary computation time by more than half in most of the cases.

Figure 4.2 shows the final configuration of our simulation where the simulation is stopped at a point in which certain domains are formed, even if they are not consolidated (using 120000 steps only). One may divide the system into small subcells, and for each of these, determine the local density. The collection of this information in the form of frequency against a given density range (or composition or order parameter) gives a histogram which profiles the overall system. An example is shown in Figure 4.3, the result of the simulation that will be described in "Result" section. The histogram shows two obvious peaks, corresponding to the vapour and liquid average densities.

It is important to notice that the choice of sub-cell size to perform the histogramming is not trivial [10]. These cells must be large enough that they can give a reasonable estimate of the density of a single phase; if the cells are too small, the density histogram will be "quantized" due to the small integer number of particles that can fit in each sub-cell. However, the use of large sub-domains will contribute to have a larger number of boxes that include significant portions of two (or more) phases, causing a smear out of the histogram. The optimal sub-cell size, despite an arbitrary quantity, is a compromise between these competing requirements.

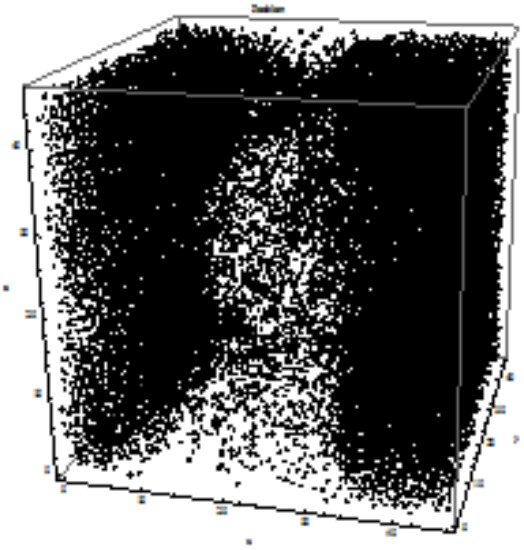


Figure 4.2
Final configuration of TQMD results using $N = 32000$, $T = 1.1$, $r_c = 5.0$

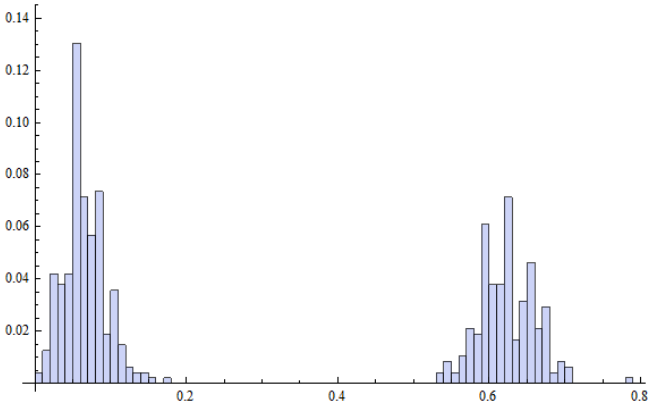


Figure 4.3
Frequency of occurrence, f, as a function of the subcell density r* for the configuration in Figure 4.2

Some sub-cells will tend to contain significant portions of two (or more) phases. A proposal for detecting and avoiding this situation based on a microscopic analysis of the fluid configuration is given, based on that in Veracoechea and Müller (2005) [10]. Average coordination number is chosen as the proper order parameter in this work. An upper (CNU) and lower (CNL) bound coordination number for which a molecule is considered as a member of each phase is defined.

The coordination number is defined arbitrarily as the number of neighbours a molecule will have within a fixed radius, which is 2σ in this work. Particles located in interface between two phases have coordination numbers reflecting the interfacial region; e.g. in the case of vapour–liquid equilibria, they have neighbours lesser than particles in the liquid phase, and greater than particles in the vapour phase. The result is that sub-cells that contain more than 15% "interfacial" particles are excluded from the histogram count. Only a rough estimate of the density or concentration differences between the two phases is needed, which is easily obtained through computer graphics visualizations of the quenched system.

4.2.3 Determination of Equilibrium Properties

It is tempted to choose the maximum shown in the histograms to obtain the corresponding phase densities. Veracoechea and Müller (2005) [10] has mentioned that such method is quantitatively poor due to the quantization of the histogram, and thus the accuracy of the estimation will be of the order of magnitude of the bin size of the histograms.

Following the discussion by the mentioned paper, if one chooses to take the maximum frequency of occurrence to correspond to the mean density, the results are erroneous. A good remedy to this problem to either use a maximum likelihood analysis or a weighted average of the histograms will give the correct results. In fact, the results in this work are found using the latter method.

5. Results and Discussion

5.1 Temperature Quench Molecular Dynamics Results

As state previously, simulations were carried out to determine the phase diagram of a pure Lennard–Jones fluid with a CS potential at a radius of $r_c = 5\sigma$. Results will be given in the usual LJ reduced units. Two sets of results are obtained, one with 120000 time steps with equilibration after 70000 steps and another with 330000 time steps with equilibration after 230000 steps. The first set of results is given in Table 5.1.

The results are quantitatively compared to those obtained from GEMC done by Veracoechea and Müller (2005) [10] (this paper will be called as Muller in the text below). The GEMC runs on 4000 particle systems, discarding the initial 40×10^6 configurations and averaging over 100×10^6 configurations. The agreement of both sets of data is considered good and acceptable. However, it is noticed the statistical error of TQMD data is large compared with those calculated by the same TQMD method by Muller. It is speculated that the main culprit behind this is the use of differing integrating algorithm and thermostats. The thermostat used in this work is the comparatively inferior Berendsen thermostat whereas the more superior Nose´-Hoover thermostat is used to control temperature by Muller [26].

Berendsen thermostat employs the scaling of velocities to obtain an exponential relaxation of the temperature to the chosen value. Temperature of the system is coupled to an external heat bath. This method gives an exponential decay of the system towards the desired temperature. Although this scheme is widely used due to its efficiency, Berendsen thermostat does not generate a correct canonical ensemble. This is due to the thermostat suppresses fluctuations of the kinetic energy of the system and therefore cannot produce trajectories consistent with the canonical ensemble.

On the other hand, Nose´-Hoover thermostat is based on extended Lagrangian formalism, which leads to a deterministic trajectory.There are no random forces or velocities to deal with. The main idea is an additional degree of freedom coupled to the physical system acts as heat bath. It permits fluctuations in the momentum temperature. Dynamics of all degrees of

Table 5.1: Saturated vapour density ρ_v^*, liquid density ρ_l^*, as a function of temperature T* for a pure LJ cut and shifted potential for 120000 steps as obtained from TQMD.

	TQMD		GEMC	
T^*	ρ_v^*	ρ_l^*	ρ_v^*	ρ_l^*
0.7	0.00259387(4)	0.00433121(30)		
0.8	0.00662778(7)	0.788268(31)	0.0071(5)	0.79(1)
0.9	0.0178247(13)	0.742599(32)	0.016(2)	0.74(2)
1.0	0.0380098(19)	0.688202(32)	0.034(2)	0.69(1)
1.1	0.0664918(28)	0.622972(38)	0.063(6)	0.625(10)
1.2	0.118373(73)	0.543659(81)	0.117(7)	0.54(1)
1.2896(7)	0.313224(1)*	0.313224(1)*		

GEMC data is taken from Veracoechea and Müller (2005) [10]
0.123(4) corresponds to 0.123 ± 0.004.
*** error calculated using Mathematica using errors as weights in model fitting**

freedom are deterministic and time-reversible. Its biggest advantage is that it correctly samples NVT ensemble for both momentum and configurations.

The integration algorithm is also of interest in determining the accuracy and precision of our results. The integration algorithm used by Muller is 5th-order Gear predictor–corrector algorithm, whereas our work uses the common velocity Verlet algorithm. Predictor-corrector algorithms is the most popular class of higher-order algorithms used in Molecular Dynamics simulations. Higher-order algorithm is an algorithm that employs information about higher order derivatives of the particle coordinates. Such an algorithm makes it possible to use a longer time step without loss of (short-term) accuracy or, alternatively, to achieve higher accuracy for a given time step. However higher-order algorithms require more storage and are, and usually, neither reversible nor area preserving. It is noted that for most Molecular Dynamics applications, Verlet-like algorithms are perfectly adequate. As shown by our results, velocity Verlet algorithm is perfectly capable to produce the correct result, although the errors associated are undeniably much larger than that of predictor-corrector algorithms.

The lowest temperature point (T^* = 0.7) was not reported for GEMC, due to the poor statistics obtained caused by the failure of the particle insertion step to accurately sample the high density of the corresponding liquid. The critical temperature for this work with 120000 time steps is $T_c^* = 1.2896 \pm 0.0069$ and critical density is $\rho_c^* = 0.313224 \pm 0.0013$. They

are estimated through rectilinear diameter of the coexistence curve and also scaled temperature–density relation using an Ising exponent of 0.32, as applied to TQMD data.

However while agreeably, the system size far larger than needed for this particular application, which is of order of magnitude 100000 particles, is needed for studies of multicomponent mixtures, asymmetric and/or multiphase fluids, as stated by Muller. According to Muller, in this work the difference in number of particles only affects the stability of the approach towards the expected equilibrium values. The smaller system size is expected to show greater fluctuations than larger system size. The use of smaller system size in our work is justified by the fact that both the systems behave similarly in terms of the number of time steps required to obtain a suitable density estimate. Muller states that after roughly 100,000 time steps the density analysis will give the same resulting value. This confirms the fact that the cluster size needed for the accurate determination of equilibrium properties is small.

In order to illustrate the advantage of using TQMD method, two sets of simulation data using 120000 time steps and 330000 time steps respectively are compared in Table 5.2. Both sets of results are in good agreement with each other within statistical uncertainties. The critical temperature and density calculated for both sets of data have a discrepancy only of order of 0.001, as shown in Table 5.2. This corresponds to the differing visual perception of the attainment of equilibrium. The systems undergoing 120000 time steps are supposedly not yet reach thermal equilibrium. Yet the value of coexisting density calculated is the same as that

Table 5.2: Comparison of TQMD result of simulation using 120000 time steps and 330000 steps

	TQMD (120000 steps)		**TQMD (330000 steps)**	
T^*	ρ_v^*	ρ_l^*	ρ_v^*	ρ_l^*
0.7	0.00259387(4)	0.00433121(30)	0.00261529(4)	0.836016(25)
0.8	0.00662778(7)	0.788268(31)	0.00691875(7)	0.786721(35)
0.9	0.0178247(13)	0.742599(32)	0.0171754(12)	0.743123(31)
1.0	0.0380098(19)	0.688202(32)	0.0349213(19)	0.686836(34)
1.1	0.0664918(28)	0.622972(38)	0.0644553(27)	0.626188(36)
1.2	0.118373(73)	0.543659(81)	0.115772(70)	0.549937(80)
	$T_c^* = 1.2896(7)$	$\rho_c^* = 0.313224(1)$	$T_c^* = 1.28969(12)$	ρ_c^* =0.313735(2)

0.123(4) corresponds to 0.123 ± 0.004.

of system undergoing 330000 time steps, which is considered reaching the thermal equilibrium. The key idea here is the concept of global and local equilibration.

In order to illustrate this, the diagram of final configurations of simulation using 120000 time steps is shown. It is seen from Figure 5.2 that at 120000 time steps, the system is far from being considered at global equilibrium. However, the densities of the corresponding phases have already achieved their equilibrium values, as shown by the data in Table 5.2. Hence contrary to the general assumption, phase coexistence data may be gathered from these locally equilibrated sub-domains, not being necessary to continue the simulation until global equilibrium is reached, resulting in a dramatic savings of computational effort.

Figure 5.3 shows the final configurations of simulation for 330000 time steps. Some are considered as reaching the thermal equilibrium. An indicator of thermal equilibrium is the existence of the flat interface separating the different bulk phases, as seen clearly in $T = 0.7$ especially. The flat interface is a consequence of the attempt by the system to reduce its free energy. It seems that this configuration is of the least free energy, in other words, the most stable form. However it seems flat interface has not yet formed in configurations of $T = 0.8$ and $T = 0.9$, indicating that equilibrium has yet to be reached. This shows that it takes a long time for the system to reach global equilibrium, even after 330000 time steps. The strength of TQMD method manifests as it can reply on local equilibrium only to predict the phase coexistence without the need for the system to attain global equilibrium.

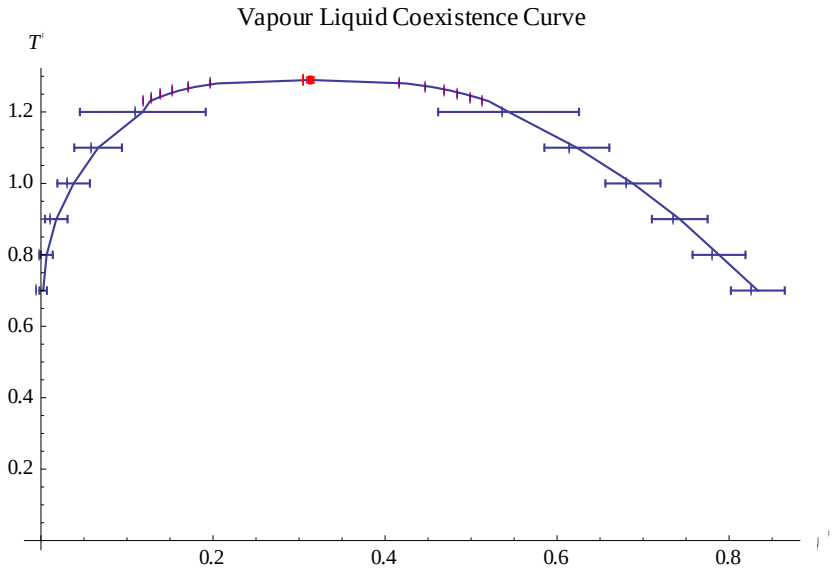


Figure 5.1
Vapour Liquid Coexistence Curve
(32000 particles, 120000 time steps, cut-off radius 5σ)

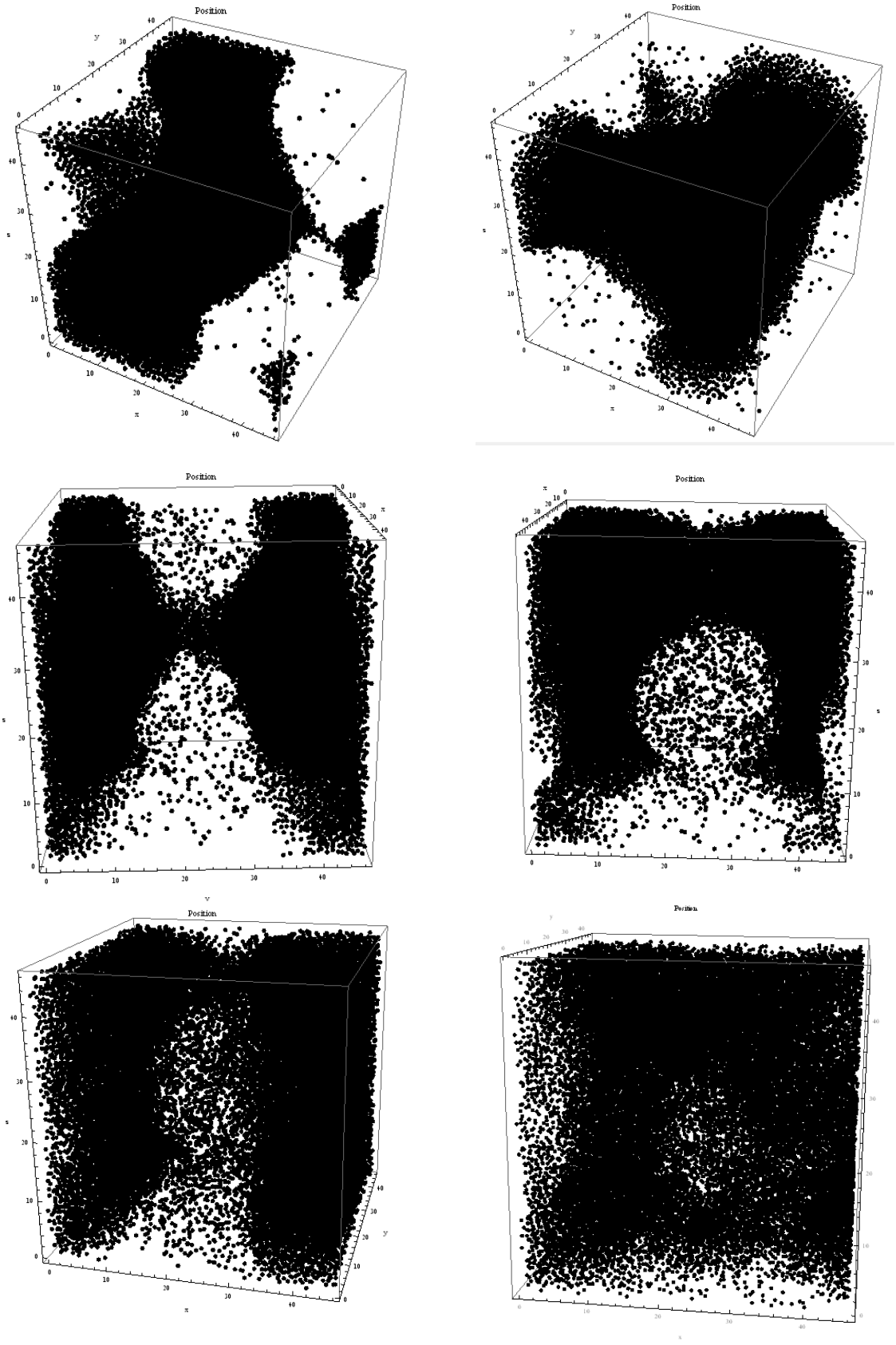


Figure 5.2
Final configuration of TQMD using 120000 time steps. From top to bottom, left to right, the temperatures of systems are respectively 0.7, 0.8, 0.9, 1.0, 1.1, 1.2.

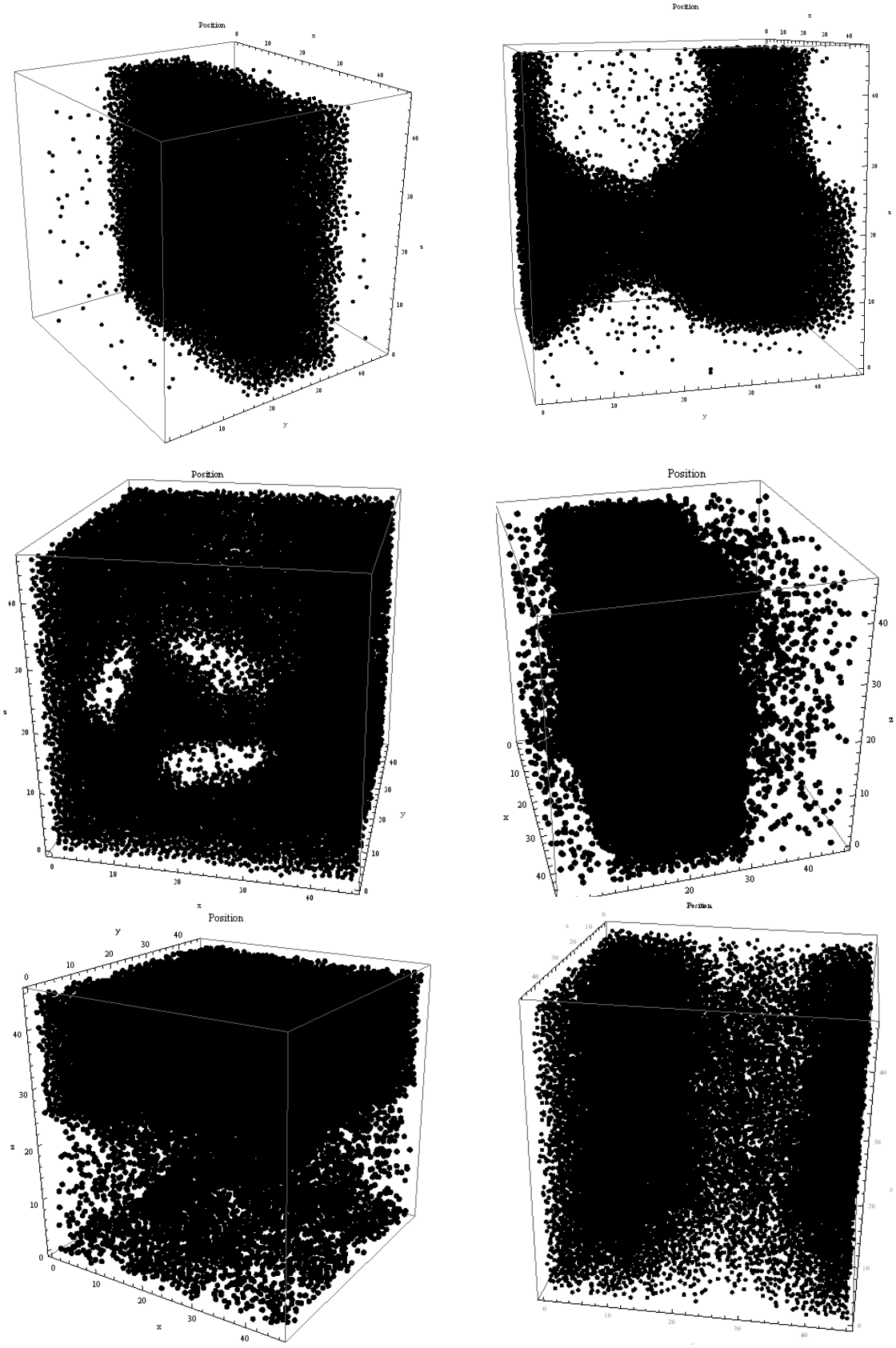


Figure 5.3
Final configuration of TQMD using 330000 time steps. From top to bottom, left to right, the temperatures of systems are respectively 0.7, 0.8, 0.9, 1.0, 1.1, 1.2.

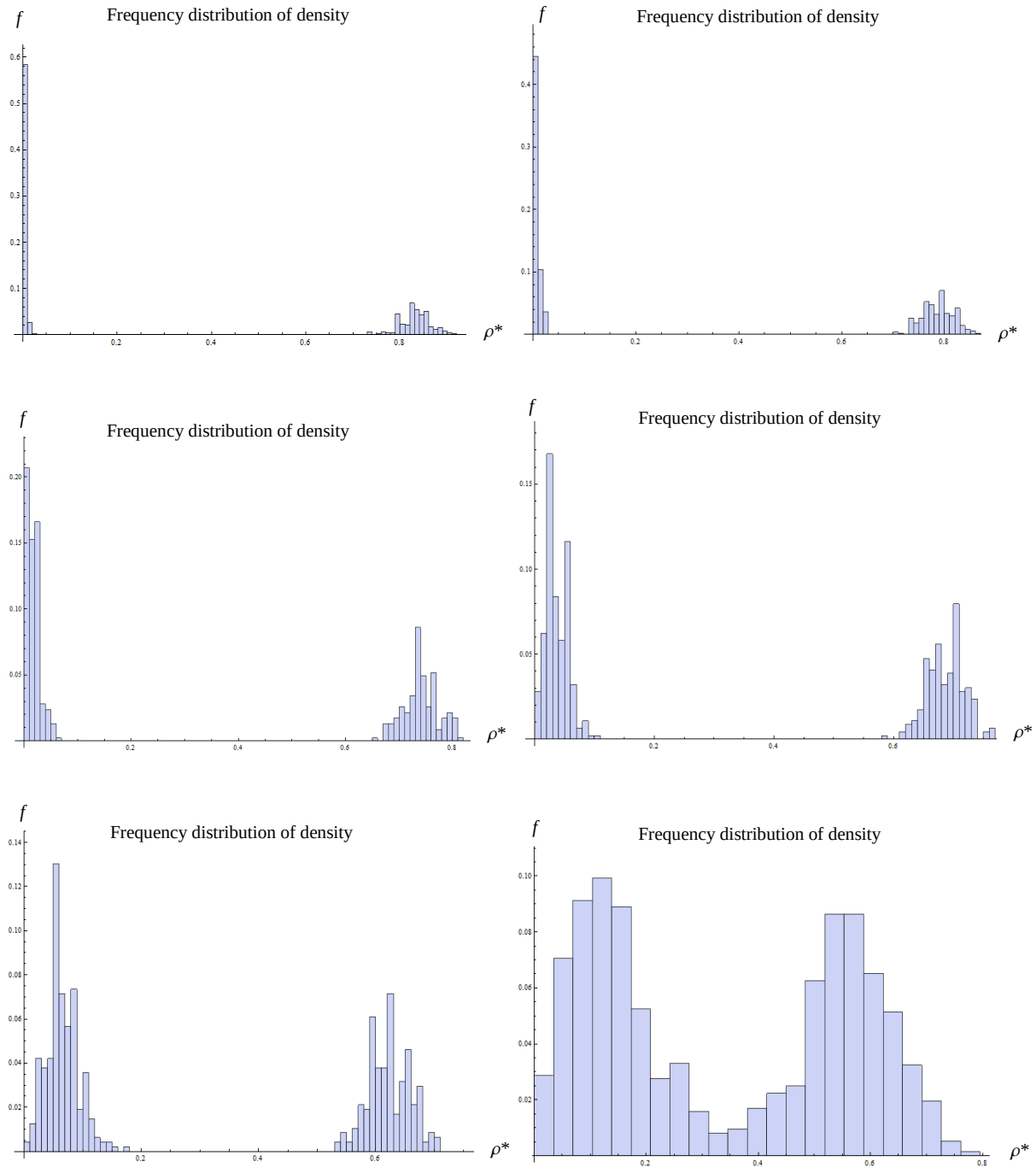


Figure 5.4
Frequency distribution of density at various temperatures, using 32000 particles, 120000 time steps. From top to bottom and left to right, the temperatures are respectively 0.7, 0.8, 0.9, 1.0, 1.1, 1.2.

For sake of completeness, Figure 5.4 comprises of frequency distribution of density of system at various temperature. As discussed previously, the whole volume of system is divided into many subcubics and their respective densities are recorded. After implementing the interface detecting algorithm, subcubics containing more than 15% interface particles are discarded. The end result is hence the existence of two histogram peaks, each indicating the bulk liquid and gas phase, the region between them are excluded using interface detection algorithm.

Using the above information, the vapour liquid coexistence curve is plotted in Figure 5.1. The data used is from simulation using 32000 particles and 120000 time steps. The critical points (temperature and density) are predicted using the law of rectilinear diameter and scaled density temperature relation with an Ising exponent of 0.32 [10]. The agreement of critical temperature to that found by Muller is good.

The critical temperature and density found are then used to predict the critical point of noble gases in Table 5.3. Lennard-Jones potential is suitable to approximate the force between noble gas particles, due to the inert, monoatomic and Van der Waals interaction force properties of noble gases. It is noticed there is a certain discrepancy between the literature values of critical points of noble gases and our calculated results. Percentage of discrepancies of critical temperature ranges from about 1.38% in case of xenon to about 6.29% in case of neon. On other hand, the critical densities of neon and argon agree well with their theoretical values, but there are some discrepancies for critical densities of krypton and xenon. It is surmised that the agreement of Lennard-Jones potential deviates from that of noble gases for

Table 5.3: Predicted values of critical temperature and density of noble gases using critical point of TQMD (120000 time steps)

Noble gas	Potential parameters		Literature values		Experimental values	
	σ(Å)	$\epsilon/k_B(K)$	$T_c(K)$	$\rho_c(\frac{g}{cm^3})$	$T_c(K)$	$\rho_c(\frac{g}{cm^3})$
Neon	2.79	36.68	44.5	0.484	47.303(25)	0.483(2)
Argon	3.38	120.0	150.85	0.536	154.752(84)	0.538(2)
Krypton	3.60	171.0	209.35	0.908	220.52(120)	0.934(3)
Xenon	4.10	221.0	289.74	1.100	285.00(155)	0.991(3)

Data of potential parameters of noble gases are taken from Balasubramanya et. al. (2006) [2]
For T_c, 47.3(25) corresponds to 47.3 ± 0.25.
For ρ_c, 0.483(2) corresponds to 0.483 ± 0.002.

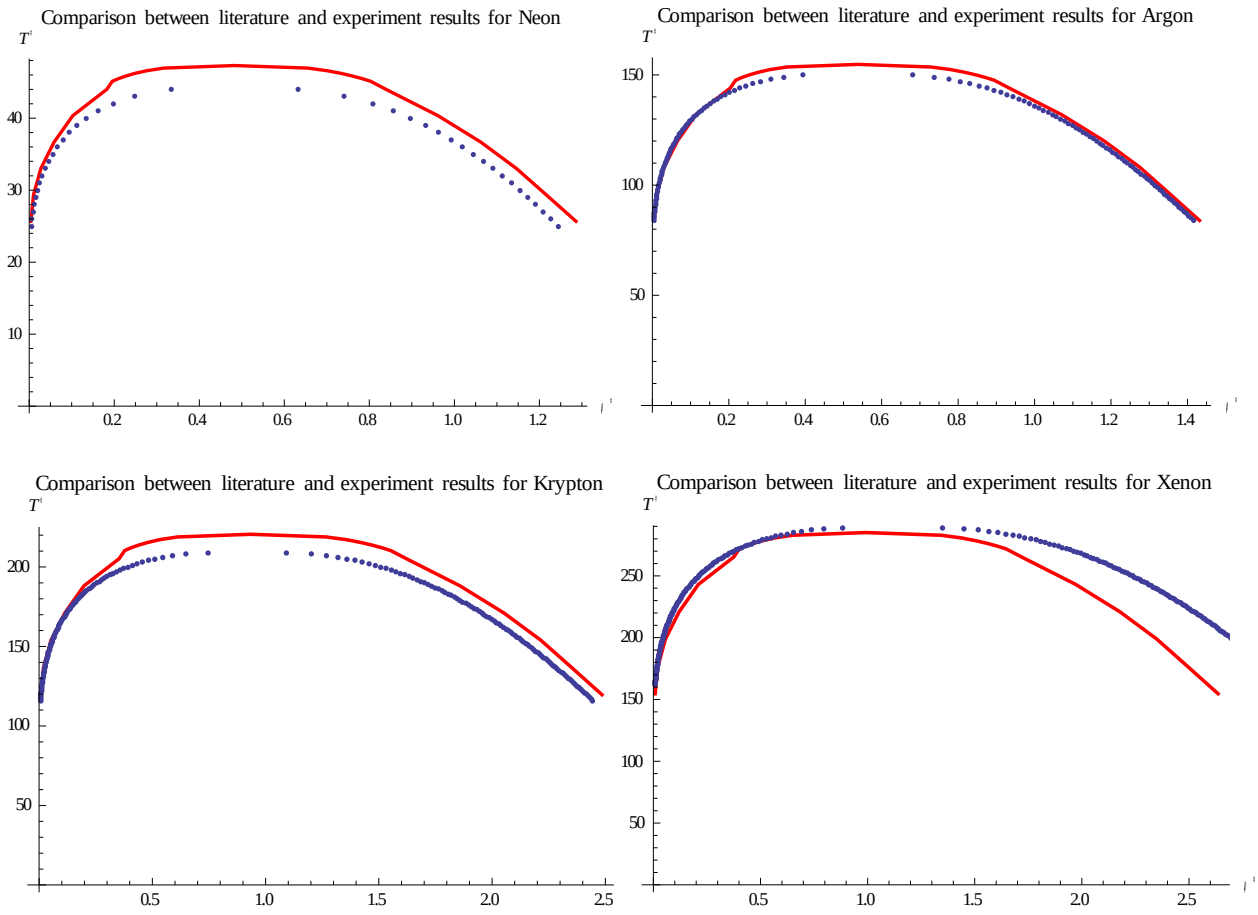


Figure 5.5
Comparison between the vapour liquid coexistence curve from literature and data from this work. The red line represents the result from this work, joining all data together. The blue dots represent the saturation data from NIST Chemistry WebBook. [18]

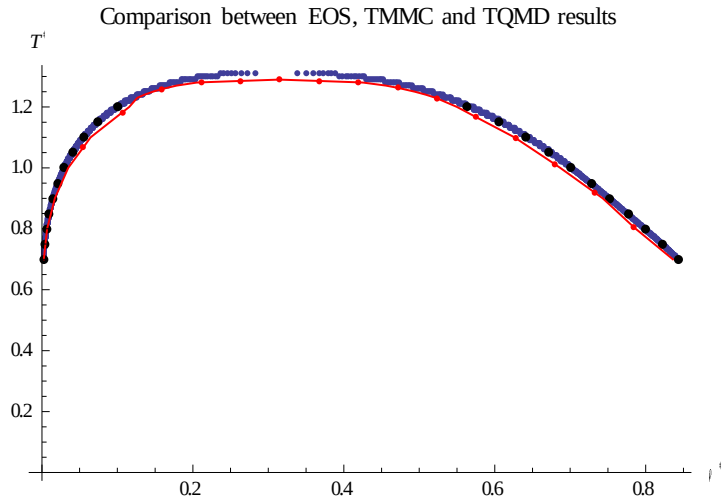


Figure 5.6
Comparison between data from Johnson's equation of state [3] (Blue dots), grand-canonical transition-matrix Monte Carlo and histogram re-weighting (Black dots) [4] and TQMD results (Red dots)

high atomic mass. It can be seen that critical density of xenon deviates much more than its literature value, by $0.109\ g/cm^3$, compared to krypton which deviates by $0.026\ g/cm^3$ only.

For sake of completeness, the comparison between results from this work and data from NIST Chemistry WebBook [18] is made. The reduced unit results from this work have been multiplied with the corresponding position and temperature parameters, stated in Table 5.3, to enable comparisons by the real unit. It is found that the agreement between argon and our result is the best among the 4 graphs, although deviations occur in region near the critical point. It may be due to inaccuracies of the law of rectilinear diameter used to estimate the data near the critical point. Neon and Krypton both deviate from our results to a certain extent, but the same consistency of data is maintained in both cases. However, Xenon shows some obvious deviation in the high liquid density region. We surmise that the Lennard-Jones potential may not be suffice to describe the real potential between the heavier xenon atoms. The higher mass of xenon may have introduced some unseen interactions between the atoms.

It should be reminded that the Lennard-Jones potential is only a theoretical approach to approximate the interaction between two uncharged molecules or atoms. Potential of noble gas is suitably described by Lennard-Jones potential, as they are monoatomic, inert, uncharged and attracting each other by Van der Waals force only. However, their potentials are not exactly described by Lennard-Jones potential. There are often more factors to be considered in the interaction between atoms than the assumed simplicity of Lennard Jones potential. Hence it is expected that the values calculated from Lennard-Jones potential will not fit their respective real values exactly.

Lastly, the TQMD results using 120000 steps are fitted against the data obtained from Johnson's equation of state [3] and grand-canonical transition-matrix Monte Carlo and histogram re-weighting [4]. Although our result is quite close to the literature data, there are some slight discrepancies between them, especially in the region near the critical point. However, the overall agreement of our results to the literatures data is considered to be good enough, and the method of TQMD is proved to be a viable method to study the location of phase coexistence of a system.

5.2 Investigation of phase properties of Lennard-Jones fluid

In order to observe the variation in some state variables and parameters as we traversing the phase diagram, the Mathematica software [27] is used to simulate the system. 3 sets of simulations are done to investigate: 1) relation between temperature and density, 2) relation between pressure and temperature, and 3) relation between pressure and volume. The number of particle used and total simulation steps will be much smaller than those of TQMD. This is due to the objective at this stage is only to observe the variation in variables across the phase diagram, not to calculate the precise values of phase coexistence variables.

5.2.1 Relation between temperature and density

The simulation is done in a *NVT* ensemble, with simulation details: number of particles = 256, total simulation steps = 20000 with equilibration after 10000 steps, cut-off radius = 2.5. For lower density $\rho = 0.05$ total simulation steps of 40000 is used whereas the total simulation steps is 30000 for $\rho = 0.10$ and $\rho = 0.20$.

The method is as follow: the density of system is fixed while the temperature is increased in steps gradually. The pressure, potential energy and compression factor at every temperature interval together with the respective radial distribution functions are recorded. The graphs of pressure versus temperature, potential energy versus temperature and compression factor versus temperature are plotted. It is deduced that the gradient of the graphs should be have some noticeable change as the phase coexistence curve is crossed. It is undeniable that the detection of phase coexistence point by changes in gradient of parameter alone is not a robust way compared to TQMD method. However this is not our objective as stated previously, so this lesser method should be sufficient for our case.

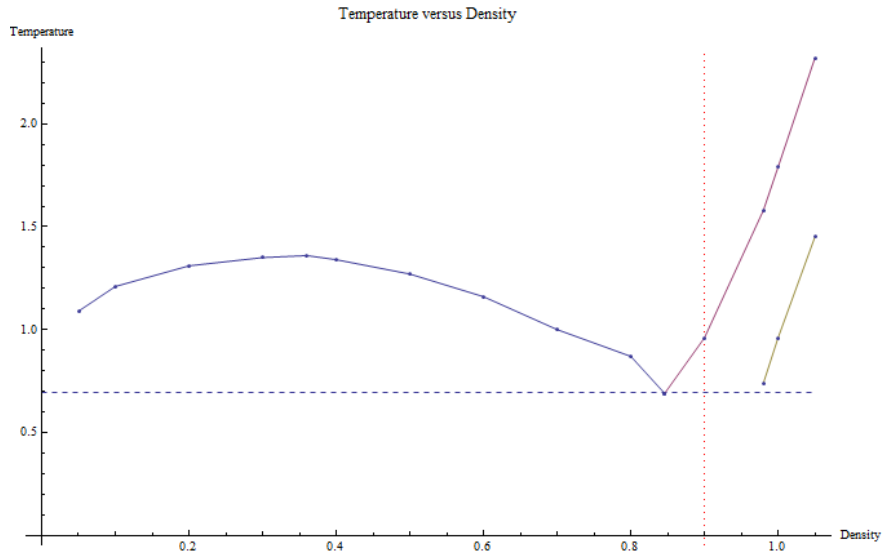


Figure 5.7
Phase diagram of temperature versus density.
The horizontal dashed line indicates the line of triple point $T^* = 0.694$, taken from Mastny and de Pablo (2007) [9].
The dotted vertical line at $\rho^* = 0.9$ will be discussed below.
Error bar is not shown as data is obtained solely through observations.

The phase diagram of temperature versus density is plotted from our simulation results. As an example, the line at $\rho^* = 0.9$ traverses from the triple line to the solid-fluid region, and then crossing the phase boundary into fluid region. The variation in parameters and graph of radial distribution function of the respective phases is shown in Figure 5.8. There is a noticeable change in gradient of the graphs of pressure, potential energy and compression factor versus temperature, indicating a phase change occurring. There seems to a discontinuous change in derivative of the graphs when the phase boundary line is crossed. On the other hand, the radial distribution function (RDF) gives structural information of particles in a system. The RDF at $T^* = 0.9$ shows a RDF shape typical of that of a solid, indicating proportion of solid is large at this stage, whereas the RDFs at $T^* = 1.1$ and $T^* = 2.5$ indicates a liquid structure. The variation in parameters along other line is included in the disc attached with this report.

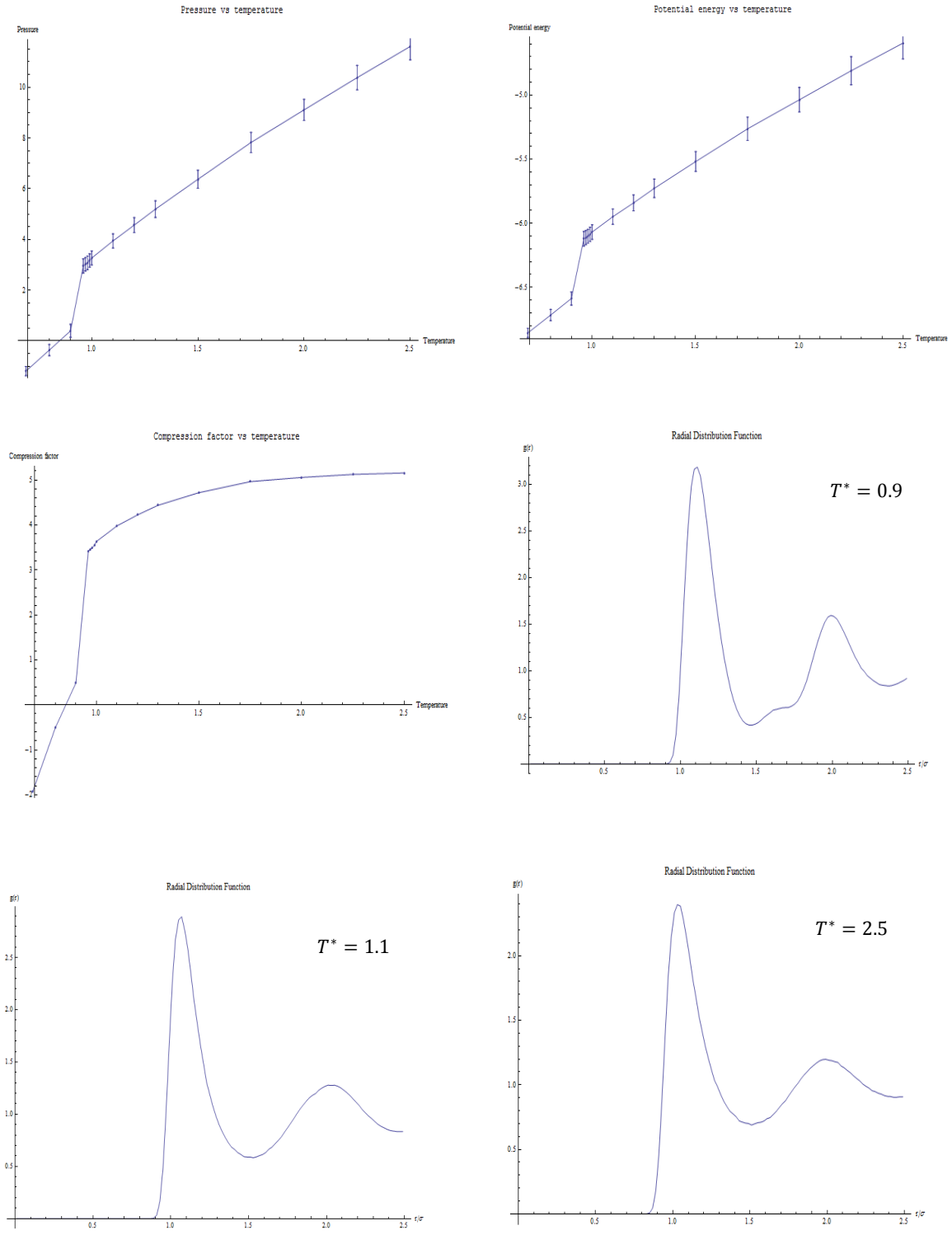


Figure 5.8

Variation in parameter along the vertical dotted line in phase diagram Figure 5.7.

From top to bottom and left to right respectively are: 1) Pressure vs temperature, 2) Potential energy vs temperature, 3) Compression factor vs temperature, 4) RDF at $T^* = 0.9$, 5) RDF at $T^* = 1.1$ and 6) RDF at $T^* = 2.5$.

5.2.2 *Relation between pressure and temperature*

The simulation is done in a *NPT* ensemble, using Berendsen thermostat to control temperature and Berendsen barostat to control pressure. The simulation details remain the same as the previous simulation. The generated phase diagram is as shown in Figure 5.9.

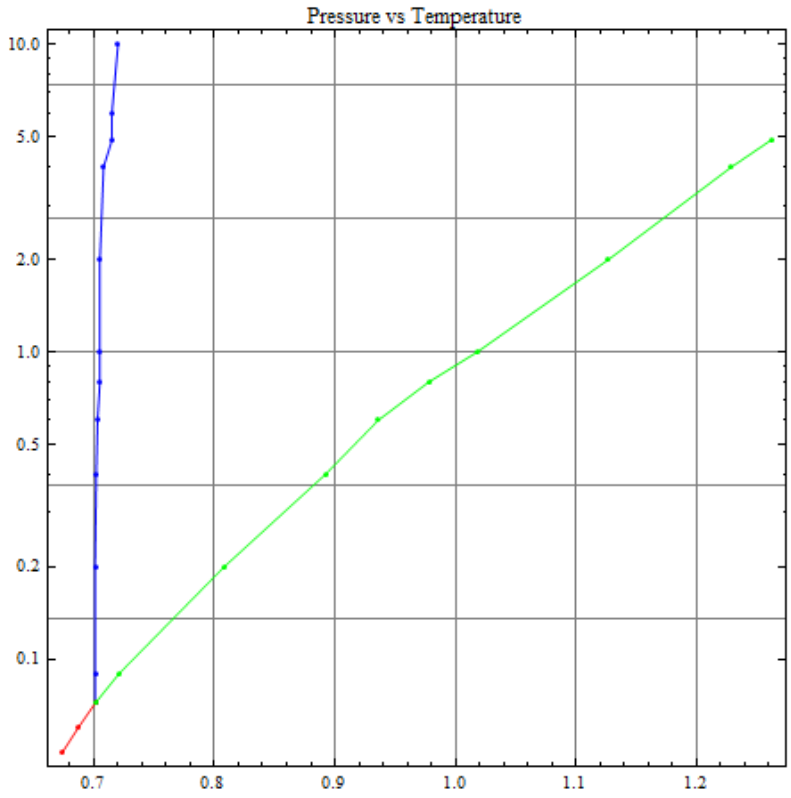


Figure 5.9
Generated phase diagram of pressure versus temperature.

Again, the generated phase diagram and the corresponding graphs and RDFs are all in included in the disc attached, so the diagrams will not be included here. However, although the phase diagram is successfully generated by detection of the changes in gradient as before, there are some peculiar trends of RDFs as one slowly increases the temperature at a fixed temperature. It is noticed that although the phase boundary is crossed, the RDF has not changed into the corresponding pattern of the new phase until the temperature is high. The reason of this problem maybe either the *NPT* ensemble is not suitable to be simulated in this way, or more superior thermostat and barostat are needed. Berendsen thermostat and barostat, though fast and effective in regulating parameters, are not a good method compared to Nose-Hoover approach, as the ensemble generated does not follow the correct distribution of a correct system.

5.2.3 Relation between pressure and volume

The last simulation is done to investigate the relationship between pressure and volume. The temperature is fixed using the Berendsen thermostat and the variation in parameters with respect to volume is recorded. The end results are displayed below: 1) pressure vs volume in Figure 5.10 and 2) compression factor vs volume in Figure 5.11. It is noted the error bars of the isotherms in both figures are omitted as it would be too messy and the objective is only obtain a comparison between various isotherms. However, the error bars do get included in the individual graphs of the isotherms, included in the disc attached, if one wishes to get information about it.

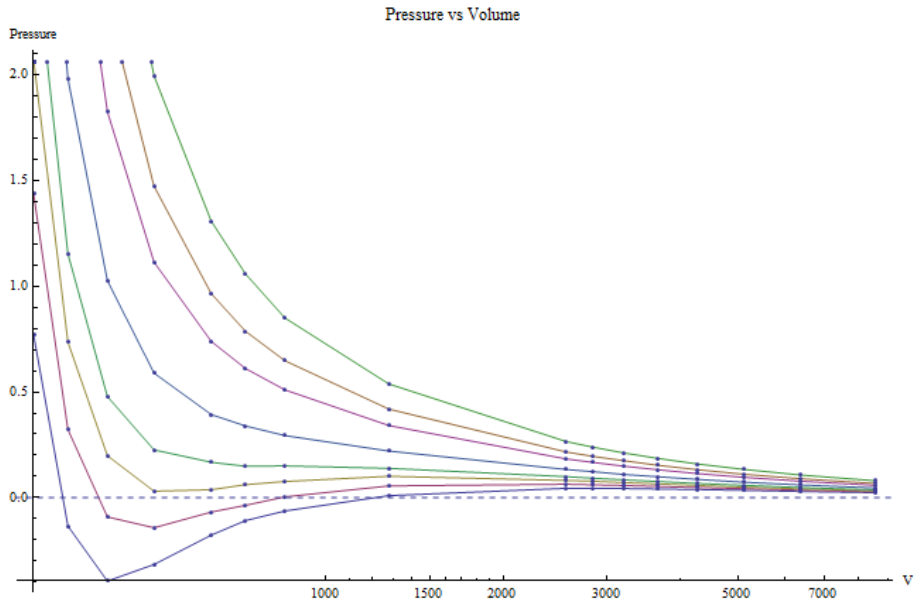


Figure 5.10
Relation between pressure and volume for various isotherms.
From the lowest to top line, the isotherms are respectively $0.7\,T_c$, $0.8\,T_c$, $0.9\,T_c$, $1.0\,T_c$, $1.2\,T_c$, $1.5\,T_c$, $1.7\,T_c$, and $2.0\,T_c$

Figure 5.10 shows the behaviors of pressure to the change in volume at constant temperature. These patterns are indeed expected as it is known a real gas behaves as an ideal gas at high temperature. The lowest line, corresponds to $0.7\,T_c$, shows the known behavior of real gas as the containing volume increases. The pressure tends to be negative in small volume, indicating that interatomic attractive force is dominant at this region, whereas interatomic force is considered negligible for an ideal gas. The threshold temperature for the system to behave like an ideal gas is the expected critical temperature $1.0\,T_c$. In fact ideal gas behavior is shown at the highest temperature $2.0\,T_c$.

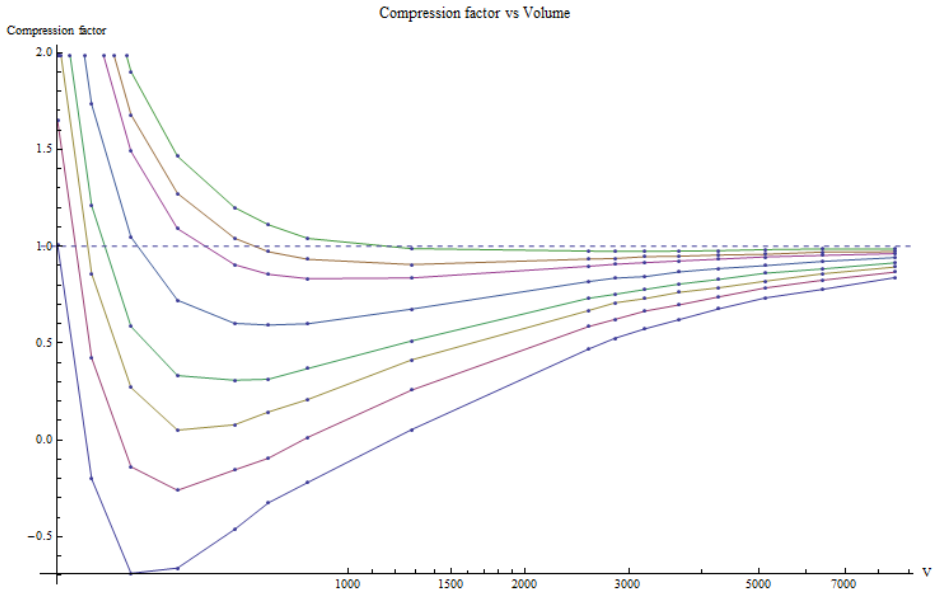


Figure 5.11
Relation between compression factor and volume for various isotherms.
From the lowest to top line, the isotherms are respectively $0.7\,T_c$, $0.8\,T_c$, $0.9\,T_c$, $1.0\,T_c$, $1.2\,T_c$, $1.5\,T_c$, $1.7\,T_c$, and $2.0\,T_c$

Another way to investigate the system is through compression factor, as shown in Figure 5.11. Compression factor (Z) of the system is used to determine whether a system obeys ideal gas law and is given by the formula:

$$Z = \frac{PV}{Nk_BT}$$

When a gas obeys ideal gas law, the compression factor equals 1.

A gas behaves more like an ideal gas when the temperature increases. Figure 5.11 shows that this is indeed the case. The isotherm at lower temperature deviates much from the ideal gas law, as indicated by the deep valley at small volume. The top 3 isotherms begin to obey the ideal gas law, approaching unity quickly as the volume of system is increased.

6. Conclusions and Recommendations

Phase transition and properties of Lennard-Jones fluid have been investigated using TQMD method and Mathematica software. A detailed analysis of TQMD method has been provided and applied to our system. TQMD method can locate phase coexistence points based on canonical molecular dynamics simulation together with a post-simulation analysis method. A system is propelled into a thermally and mechanically unstable state to be rapidly divided into two coexisting phases separated by an interface.

The strength of TDMD method lies in the extremely fast quenches accessible to simulation, providing a large driving force for phase separation to happen. Contrary to the normal perception, local equilibration of densities and compositions occurs quickly. In this work, it is shown that these densities and compositions are representative of the bulk equilibrium values, by comparing configuration at immediate point to that in global equilibrium. Hence simulation times are greatly reduced as the formation of a planar interface is not need, at least in this simple pure Lennard-Jones fluid, compared to the conventional molecular dynamics method.

The results obtained shown to be in good agreement with that of GEMC, although the errors are larger due to reasons explained previously. Although GEMC is faster than TQMD, TQMD can be applied in a straightforward way to more types of phase equilibrium and more complex molecular species and is especially suited for dense fluid phase equilibria, for complex molecules and for multicomponent or polydisperse systems, according to Veracoechea and Müller (2005) [10]. It is noted that the interface created using TQMD method is exactly what GEMC are designed to avoid. However, as more modern computing equipment is produced, the difficulties created by the interface can be avoided.

TQMD approach is especially suitable for the study of phase equilibria in unfamiliar systems (Gelb and Müller, 2002) [11]. The methodology is the use of a sequence of quenches at different densities and quench-temperatures to determine points of phase coexistence efficiently. Quenches which do not result in phase separation used to bound coexistence lines and surfaces.

The Mathematica software [27] is powerful in illustration of a list of data, as in our case of displaying the variation of state variables and pattern of radial distribution functions. Our simulations successfully generate the phase diagram for temperature vs density, although its accuracy is not comparable to the TQMD method. The change of gradient of state variables as the phase boundary line is crossed indicates a change in properties of the system. The same can said about the simulation in *NPT* ensembles and investigation of effect of variation of volume on pressure and compression factor. The tendency of the system to approach the properties of an ideal gas at large volume and high temperature, as shown in the last simulation results, indicates the ability of our code to model the real system.

Future extension to this project can be on the parallelization of TQMD code. Parallelized MD code enables the use of massively parallel computers and takes full advantage of the advent in technology nowadays. The trends in electronic market to produce more and more of lower cost and powerful parallel computer setups further enhances and promote the use of the method. Binary and multiphase system can also be simulated using TQMD method due to universality of the method to apply to various systems.

7. References

[1] Anthony Felts (2010): Introduction: History and Reasons for Molecular Simulations, Simulations in Computational Biophysics, Rutgers the State University of New Jersey, Spring 2010

[2] Balasubramanya, M. K., Roth, M. W., Tilton, P. D., & Suchy, B. A. (2006). Molecular dynamics simulations of noble gas release from endohedral fullerene clusters. arXiv preprint cond-mat/0607535.

[3] Benchmark results for Lennard-Jones fluid, SAT-EOS: Liquid-vapor coexistence properties
http://www.cstl.nist.gov/srs/LJ_PURE/sateos.htm

[4] Benchmark results for Lennard-Jones fluid, SAT-TMMC: Liquid-vapor coexistence properties
http://www.cstl.nist.gov/srs/LJ_PURE/sattmmc.htm

[5] Berend Smit (1996): Molecular simulations of fluid phase equilibria, Fluid Phase Equilibria 116 (1996) 249-256

[6] Cameron Abrams (2009). Lecture Notes, CHE 800-002: Molecular Simulation Spring 0304, Drexel University, Philadelphia, March 21 - December 1, 2009

[7] D. C. Rapaport. The Art of Molecular Dynamics Simulation. Cambridge University Press, Cambridge, 1995.

[8] D. Frenkel and B. Smit. Understanding Molecular Simulation: From Algorithms to Applications. Academic Press, San Diego, 2 edition, 2002.

[9] Ethan A. Mastny and Juan J. de Pablo "Melting line of the Lennard-Jones system, infinite size, and full potential", Journal of Chemical Physics 127 104504 (2007)

[10] F. Mart ́nez-Veracoechea & E.A. Müller (2005): Temperature-quench Molecular Dynamics Simulations for Fluid Phase Equilibria, Molecular Simulation, 31:1, 33-43

[11] Gelb, L.D. and Müller, E.A. (2002) “Phase equilibria by means of temperature-quench molecular dynamics”, Fluid Phase Equilibr. 203, 1.

[12] Giordano and Nakanishi. Computational Physics. Addison-Wesley, 2nd Edition, 2005

[13] Gu, K., C. B. Watkins, and J. Koplik. "Molecular dynamics simulation of the equilibrium liquid-vapor interphase with solidification." Fluid Phase Equilibria 297.1 (2010): 77-89.

[14] J.J. Nicolas , K.E. Gubbins , W.B. Streett & D.J. Tildesley (1979): Equation of state for the Lennard-Jones fluid, Molecular Physics: An International Journal at the Interface Between Chemistry and Physics, 37:5, 1429-1454

[15] J.K. Johnson, J.A. Zollweg, K.E. Gubbins, Mol. Phys. 78 (1993) 591–618.

[16] Jean-Pierre Hansen and Loup Verlet (1969): Phase Transition of the Lennard-Jones System, Physical Review Volume 184, Number 1

[17] Matsumoto, M. (1998). Molecular dynamics of fluid phase change. Fluid phase equilibria, 144(1), 307-314.

[18] NIST Chemistry WebBook, Thermophysical Properties of Fluid Systems http://webbook.nist.gov/chemistry/fluid/

[19] Panagiotopoulos, A. Z. (2000). Monte Carlo methods for phase equilibria of fluids. Journal of Physics: Condensed Matter, 12(3), R25.

[20] Philippe A. Bopp , Jörn B. Buhn , Holger A. Maier & Manfred J. Hampe (2008): SCOPE AND LIMITS OF MOLECULAR SIMULATIONS, Chemical Engineering Communications, 195:11, 1437-1456

[21] T.P.Straatsma (2004), Molecular Modeling and Simulation, Computational Biology Noontime Seminar Series, Pacific Northwest National Laboratory, July 26, 2004.

[22] Thijssen, J. (2007). Computational physics. Cambridge University Press

[23] UCDAVIS CHEMWIKI (2010) Lennard-Jones Potential http://chemwiki.ucdavis.edu/Physical_Chemistry/Quantum_Mechanics/Atomic_Theory/Intermolecular_Forces/Lennard-Jones_Potential

[24] van Gunsteren, W. F., & Berendsen, H. J. (1990). Computer simulation of molecular dynamics: Methodology, applications, and perspectives in chemistry. Angewandte Chemie International Edition in English, 29(9), 992-1023.

[25] Verlet, L. (1967). Computer" experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. Physical review, 159(1), 98.

[26] Victor Rühle (2007). Berendsen and Nose-Hoover thermostats. August 8, 2007

[27] Wolfram Mathematica 8 documentation

8. Appendix

The complete TQMD code will be listed here. The rest of work such as Mathematica code is stored in the disc attached to the project.

```
# include <iostream>
# include <cmath>
# include <vector>
# include <cstdlib>
# include <complex>
# include <ctime>
# include <fstream>

using namespace std;

# include "normal.h"    // normal distribution function

#define PI 3.14159      // values of Pi
#define N_OFFSET 14     // total number of offsets for cell method

/*************************************************************************/

/* Initialization of system by assigning position and velocity */
void init(double rx[], double ry[], double rz[], double vx[], double vy[],
 double vz[], int nparticle, double targetT, double L)
{
    int i, ix, iy, iz;
    int LinCell;        // number of unit cells along cube edge length
    int seed;           // normal distribution generating function use
    int counter=-1;     // number of particles inserted

    double cmvx=0., cmvy=0., cmvz=0., KE=0.;
    double lattconst, T, fac;

    LinCell = floor(pow(nparticle/4.,1./3.) + 0.5);
    lattconst = L/LinCell;      // lattice constant of unit cell
    for(ix=0;ix<LinCell;ix++)
    {
        for(iy=0;iy<LinCell;iy++)
        {
            // assignning particle to a unit cell in fcc lattice
            for(iz=0;iz<LinCell;iz++)
            {                           // 4 particles in a unit cell
                counter++;
                rx[counter]=(ix+0.25)*lattconst;
                ry[counter]=(iy+0.25)*lattconst;
```

```
            rz[counter]=(iz+0.25)*lattconst;

            counter++;
            rx[counter]=(ix+0.75)*lattconst;
            ry[counter]=(iy+0.75)*lattconst;
            rz[counter]=(iz+0.25)*lattconst;

            counter++;
            rx[counter]=(ix+0.75)*lattconst;
            ry[counter]=(iy+0.25)*lattconst;
            rz[counter]=(iz+0.75)*lattconst;

            counter++;
            rx[counter]=(ix+0.25)*lattconst;
            ry[counter]=(iy+0.75)*lattconst;
            rz[counter]=(iz+0.75)*lattconst;
        }
    }
}

seed = time(NULL);           // initialization for normal.h
for(i=0;i<nparticle;i++)     // assignning velocity to particles
{
    vx[i] = r8_normal_01 ( seed );
    vy[i] = r8_normal_01 ( seed );
    vz[i] = r8_normal_01 ( seed );
}

for(i=0;i<nparticle;i++)     // total velocity along a dimension
{
    cmvx += vx[i];
    cmvy += vy[i];
    cmvz += vz[i];
}

for(i=0;i<nparticle;i++)     // remove velocity of center of system
{
    vx[i] -= cmvx/nparticle;
    vy[i] -= cmvy/nparticle;
    vz[i] -= cmvz/nparticle;
    KE += vx[i]*vx[i]+vy[i]*vy[i]+vz[i]*vz[i];
}

KE*=0.5;                     // kinetic energy
T=KE/nparticle*2./3.;        // temperature
fac=sqrt(targetT/T);         // temperature rescaling factor
KE=0;
for(i=0;i<nparticle;i++)     // velocity rescaling
{
    vx[i] *= fac;
    vy[i] *= fac;
    vz[i] *= fac;
```

```
        KE += vx[i]*vx[i]+vy[i]*vy[i]+vz[i]*vz[i];
    }
    KE*=0.5;                    // new kinetic energy

}

/*****************************************************************************/

// function to wrap the cell due to periodic boundary
void cellwrap(int &var, double &change, int cells, double L)
{
    if(var >= cells){       // when variable exceeds max number of cells
        var = 0;
        change = L;         // position wrap
    }
    else if(var < 0){       // when variable does not belong to any cell
        var = cells - 1;
        change = -L;        // position wrap
    }
}

/*****************************************************************************/


/*

Cell subdivison method is used to reduces the computational effort to the
O(N) level. Simulation region is divided into a lattice of small cells, and
that the cell edges all exceed rc in length. Atoms are assigned to cells on
the basis of their current positions, so that interactions are only
possible between atoms that are either in the same cell or in immediately
adjacent cells. Due to symmetry only half the neighboring cells need be
considered

A linked list is used to store the required data about cell no. and
position of particles. Each linked list requires a separate pointer f to
access the first data item, and the item terminating the list must have a
special pointer value, 1 in our case. f = a points to a-th particle as the
first item in the list, p_a = b points to b-th particle as the second item,
and so on, until a pointer value p_z = 1 is encountered, terminating the
list. In summary, linked lists are used to associate atoms with the cells
in which they reside at any given instant.

Normally each a separate list is required for each cell. However rather
than use separate arrays for the two kinds of pointer, namely that between
atoms in the same cell and that to the first particle in a cell, the first
nparticle elements in array cellList are used for the former and the
remainer for the latter. This method is introduced in The Art of Molecular
Dynamics Simulation by D. C. Rapaport.

*/
```

```
// computation of forces
double compute(double rx[], double ry[], double rz[], double fx[],
 double fy[], double fz[], int *gr, int nparticle, double L, double rc,
 int n, int nEqui,  int grs, double &vir, double ecut, double ecor,
 double dh, int cells,  int cellList[])
{
    // declaration of function
    void cellwrap(int &, double &, int, double);

    // common variable declaration
    int i,j1,j2,bin;          // for loop variables and bin counting
    double dx,dy,dz;          // distance between a pair of particles
    double e=0., hL=L/2.0;    // potential energy and half cube edge length
    double f, r2, rc2, r6i;   // variable for force calculation

    // cell list variable declaration
    double shiftx, shifty, shiftz;      // shift of cell due to PBC
    int ccx, ccy, ccz, c, m1x, m1y, m1z, offset;
    int m1, m2vx, m2vy, m2vz, m2;
    int vOff[14][3]={{0,0,0}, {1,0,0}, {1,1,0}, {0,1,0},{-1,1,0},{0,0,1}, \
    {1,0,1}, {1,1,1}, {0,1,1}, {-1,1,1}, {-1,0,1}, \
    {-1,-1,1}, {0,-1,1}, {1,-1,1}};  // offsets of the 14 neighbor cells

    // Initialization
    rc2=rc*rc;
    for(i=0;i<nparticle;i++)          // set initial force to zero
    {
        fx[i]=fy[i]=fz[i]=0.;
    }
    vir=0.;                           // set virial value to zero

    // Cell list method initialization
    for(i=nparticle;i<(nparticle+cells*cells*cells);i++) cellList[i] = -1;
    for(i=0;i<nparticle;i++)          // assigning each particle to cell
    {
        ccx  = rx[i]/(L/cells);       // cell no. along a given dimension
        ccy  = ry[i]/(L/cells);
        ccz  = rz[i]/(L/cells);
        // cell no. in scalar form
        c = ((ccz * cells + ccy) * cells + ccx) + nparticle;
        cellList[i] = cellList[c];  // form a linked list for the cells
        cellList[c] = i;
    }

    for (m1z = 0; m1z < cells; m1z++)        // looping of cells
    {
        for (m1y = 0; m1y < cells; m1y ++)
        {
            for (m1x = 0; m1x < cells; m1x ++)
            {
                // cell no.
```

```
m1 = ((m1z * cells + m1y) * cells + m1x) + nparticle;
// looping of neighbor cells
for (offset = 0; offset < N_OFFSET; offset ++)
{
    // numbers of the neighboring cell
    m2vx = m1x + vOff[offset][0];
    m2vy = m1y + vOff[offset][1];
    m2vz = m1z + vOff[offset][2];
    shiftx=shifty=shiftz=0.;
    // periodic boundary of cell
    cellwrap(m2vx, shiftx, cells, L);
    cellwrap(m2vy, shifty, cells, L);
    cellwrap(m2vz, shiftz, cells, L);

    // scalar cell no. of the neighboring cell.
    m2 = ((m2vz * cells + m2vy) * cells + m2vx) + nparticle;
    // non-sequential progression of cell elements
    for (j1 = cellList[m1]; j1 >= 0; j1 = cellList[j1])
    {
        for (j2 = cellList[m2]; j2 >= 0; j2 = cellList[j2])
        {
            // avoid double counting
            if (m1 != m2 || j2 < j1)
            {
                // pair distance
                dx = rx[j1] - rx[j2];
                dy = ry[j1] - ry[j2];
                dz = rz[j1] - rz[j2];
                // change of distance due to PBC
                dx -= shiftx;
                dy -= shifty;
                dz -= shiftz;
                // sum of squares
                r2 = dx*dx + dy*dy + dz*dz;
                // enter if located within cut-off radius
                if(r2<rc2)
                {
                    r6i = 1./(r2*r2*r2);
                    // potential energy calculation
                    e += 4*(r6i*r6i - r6i) - ecut;
                    // force magnitude calculation
                    f = 48*(r6i*r6i - 0.5*r6i);
                    // force component along a direction
                    fx[j1] += dx*f/r2;
                    fx[j2] -= dx*f/r2;
                    fy[j1] += dy*f/r2;
                    fy[j2] -= dy*f/r2;
                    fz[j1] += dz*f/r2;
                    fz[j2] -= dz*f/r2;
                    // virial value
                    vir += f;
                }
```

```
                                    // radial distribution data
                                    if(n > nEqui && n%grs == 0 && r2 < rc2)
                                    {
                                        bin=(int)(sqrt(r2)/dh);
                                        gr[bin]+=2;
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
        // returning the corrected value of potential energy
        return e+nparticle*ecor;

}

/*************************************************************************/

// main executing function
int main(int argc, char *argv[]) {
    /* basic variable declartion */

    // number of particle, simulation steps, equibration steps, for rdf use
    const int nparticle= 32000, nSteps= 120000, nEqui= 70000, grs= 10;
    // time to start quench the system
    int ndrop= 20000;
    int ngr, nHis, i, n;    // variable for RDF use
    int * gr;
    double * doublegr;
    clock_t t1,t2;          // for execution time calculation

    // density, time steps, cut-off radius, berendsen coefficient
    double rho = 0.328, dt = 0.004, rc = 5.0, tau = 0.1;
    // initial temperature, quench temperature, temperature variable
    double initT = 4.0, finalT = 0.7, targetT;

    // general needed variable like correction terms of potential energy
    double V, L, rr3, ecor, pcor, ecut, dt2, rc2, dh;
    double T, stemp, temp, P, spres, pres, sPE, vPE;
    double vb, nid;
    // potential and kinetic energy, virial, thermostat ratio
    double PE, KE, vir, lambda;
    // forces of particle
    double fx[nparticle], fy[nparticle], fz[nparticle];
    // positions of particle
    double rx[nparticle], ry[nparticle], rz[nparticle];
    // velocities of particle
    double vx[nparticle], vy[nparticle], vz[nparticle];
```

```
// for calculating execution time
t1=clock();

V = nparticle/rho;  // volume of system
L = pow(V,1./3.);   // cube edge length
rc= min(rc,L/2);    // minimum between cut0off radius and half length
rr3 = 1/pow(rc,3.);
// potential energy correction term
ecor = 8*PI*rho*(rr3*rr3*rr3/9.0 - rr3/3.0);
// pressure correction term
pcor = 16.0/3.0*PI*rho*rho*(2./3.*rr3*rr3*rr3 - rr3);
// shifted potential energy calue
ecut = 4.*(pow(rr3,4.) - pow(rr3,2.));
rc2 = pow(rc,2.);
dt2 = pow(dt,2.);

/* cell list variable declaration */
// number of cell along a dimension
int cells = L/rc;
// array storing cell information
int cellList[cells*cells*cells + nparticle];

// set the initial temperature
targetT = initT;

// initialize the sum of temperature, pressure and potential energy
stemp=0.0;
spres=0.0;
sPE=0.0;

/* variable used to calculate radial distribution function (rdf) */
ngr = 0;
dh = 0.02;
nHis = (int)(rc/dh);
gr = (int*)calloc(nHis,sizeof(int));
doublegr = (double*)calloc(nHis,sizeof(double));

/* intialization of system */
// calling functions
init(rx,ry,rz,vx,vy,vz,nparticle,targetT,L);
PE = compute(rx, ry, rz, fx, fy, fz, gr, nparticle, L, rc, 1,
 nEqui, grs, vir, ecut, ecor, dh, cells, cellList);

cout << "Number of particles: " << nparticle << endl;

/****************** starting the simulation ******************/

for(n=0;n<nSteps;n++)    // looping of simulation steps
{
    if(n%1000==0) cout << n << endl;
    // quenching of system is done
    if(n==ndrop) targetT = finalT;
```

```
/* First integration half-step */
for(i=0;i<nparticle;i++)
{
    // first part of velovity Verlet algorithm
    rx[i]+=vx[i]*dt+0.5*dt2*fx[i];
    ry[i]+=vy[i]*dt+0.5*dt2*fy[i];
    rz[i]+=vz[i]*dt+0.5*dt2*fz[i];
    vx[i]+=0.5*dt*fx[i];
    vy[i]+=0.5*dt*fy[i];
    vz[i]+=0.5*dt*fz[i];
    /* Apply periodic boundary conditions */
    if (rx[i]<0.0) rx[i]+=L; else if (rx[i]>L) rx[i]-=L;
    if (ry[i]<0.0) ry[i]+=L; else if (ry[i]>L) ry[i]-=L;
    if (rz[i]<0.0) rz[i]+=L; else if (rz[i]>L) rz[i]-=L;
}
// Calculate forces
PE = compute(rx, ry, rz, fx, fy, fz, gr, nparticle, L, rc,
 n, nEqui, grs, vir, ecut, ecor, dh, cells, cellList);

// required for normalization of radial distribution function
if(n > nEqui && n%grs == 0) ngr++;

/* Second part of velocity Verlet algorithm */
KE = 0.0;
for(i=0;i<nparticle;i++)
{
    vx[i]+=0.5*dt*fx[i];
    vy[i]+=0.5*dt*fy[i];
    vz[i]+=0.5*dt*fz[i];
    KE+=vx[i]*vx[i]+vy[i]*vy[i]+vz[i]*vz[i];
}
KE*=0.5;

/* Berendsen thermostat */
// Berendsen coefficient
lambda = sqrt(1 + dt/tau*(targetT/(2.0*KE/3.0/nparticle) - 1.0));
KE=0.0;
for(i=0;i<nparticle;i++)    // rescaling velocity
{
    vx[i]*=lambda;
    vy[i]*=lambda;
    vz[i]*=lambda;
    KE+=vx[i]*vx[i]+vy[i]*vy[i]+vz[i]*vz[i];
}
KE*=0.5;

/* determine the current state propeties */
temp = KE/nparticle*2./3.;
if(n>nEqui) stemp+=temp;

pres = rho*KE*2./3./nparticle + vir/3.0/V + pcor;
```

```
    if(n>nEqui) spres+=pres;

    if(n>nEqui) sPE+=PE;

}

/* normalizing radial distribution function */
for(i=0;i<nHis;i++)
{
    vb = ((i+1)*(i+1)*(i+1)-i*i*i)*dh*dh*dh;
    nid = (4./3.)*PI*vb*rho;
    doublegr[i]=(double)(gr[i])/(ngr*nparticle*nid);
}

/*************************** output ******************************/

/* radial distribution function data */
ofstream file1 ("rdf.txt");
for(i=0;i<nHis;i++)
{
    file1 << "{" << dh*(i+0.5) << "," << doublegr[i] << "}" << endl;
}
file1.close();

/* state variables  */
T = stemp/(nSteps-nEqui);
P = spres/(nSteps-nEqui);
vPE = sPE/nparticle/(nSteps-nEqui)+ecor;

ofstream file2 ("variable.txt");
file2 << "The temperature is " << T << endl;
file2 << "The pressure is " << P << endl;
file2 << "The potential energy is " << vPE << endl;
file2.close();

cout << "The temperature is " << T << endl;
cout << "The pressure is " << P << endl;
cout << "The potential energy is " << vPE << endl << endl;

/* position of particles at final configuration */
ofstream file3 ("position.txt");
for(i=0;i<nparticle;i++)
{
    file3 << "{" << rx[i] << "," << ry[i] << "," << rz[i]
     << "}" << endl;
}
file3.close();

// calculating execution time
t2=clock();
cout << (t2-t1)/CLOCKS_PER_SEC;
```

```
    cin.get();
    return 0;
}
```