UNIVERSITY OF CAPE COAST

MOLECULAR DYNAMICS SIMULATION: FROM THEORETICAL CONCEPT TO APPLICATION

BY

CHRISTOPHER NIKINGBONG

Thesis submitted to the Department of Mathematics of the School of Physical Sciences, College of Agriculture and Natural Sciences, University of Cape Coast, in partial fulfilment of the requirements for the award of Master of Philosophy degree in Mathematics

JULY 2019

DECLARATION

Candidate's Declaration

I hereby declare that this thesis is the result of my own original research and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature .... Date ....

Name: Christopher Nikingbong

Supervisors' Declaration

We hereby declare that the preparation and presentation of the thesis were supervised in accordance with the guidelines on supervision of thesis laid down by the University of Cape Coast.

Principal Supervisor's Signature .... Date ....

Name: Dr. Samuel Mindakifoe Naandam

Co-Supervisor's Signature .... Date: ....

Name: Dr. Stephen Edward Moore

ABSTRACT

Molecular dynamics is a computer simulation method that calculates the time dependent behavior of a molecular system. It predicts how the atoms of a given system are moving with respect to time by numerically solving the Newtons equation of motion. In this thesis, description of the foundations of molecular dynamics was presented and a MATLAB code built to implement molecular dynamics. Simulations were run from the implementation and the results including some dynamical properties of molecular dynamics were reported and discussed. A comparative study between two numerical integration techniques, Velocity Verlet and Euler's algorithms, of the equations of motion in terms of the computational time was investigated. It was observed that the Velocity Verlet algorithm is faster and more stable than Eulers algorithm. Finally, a real life application of molecular dynamics techniques for protein structure prediction was presented using the AMBER package; a well-known package for molecular dynamics simulation.

KEY WORDS

Computational Speed

Equations of Motion

Lennard-Jones Potential

Molecular Dynamics Simulation

Numerical Integration

Protein Folding

ACKNOWLEDGEMENTS

I wish to thank Dr. Samuel Mindakifoe Naandam for his continuous guidance, patience, motivation and invaluable time and ideas throughout my academic career. His guidance, suggestions and encouragement led to the development of this work. I also wish to thank Dr. Stephen Edward Moore for his invaluable suggestions, comments, guidance and support towards completion of this thesis.

I wish to acknowledge the Head of Department of Mathematics, UCC, Prof. Emmanuel Kwame Essel for his support and advice throughout this programme. I will also like to acknowledge the lecturers and all the staff of the department.

I would like to specially thank Gideon Kwadzo Gogovi for introducing me to the research field of molecular dynamics. I have gained much knowledge and insightful thoughts on scientific research, and I am thankful for all his guidance, support, trust and encouragement. I appreciate him for spending his valuable time to support me come out with this work.

I will also like to thank Paul Chataa, Gazari John, James Kangben, Ahmed Dawal and all my friends who supported me in one way or the other towards my pursuit for MPhil degree.

Finally, I would like to express my very profound gratitude to my dad (Poidim Nikingbong), my mum (Monoca M. Duut), my sisters (Mariama Bajian, Fausty Bajian, Biidi Yakubu and Biili Poidim) and my brother (Timothy Yada Kizito) for their love, prayers and financial support. I appreciate my entire family and friends for all their contributions to my life.

God bless you all.

DEDICATION

To my family

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

The things that we can see and feel around us are made up of tiny particles called atoms which interact through an interatomic forces. We can think of the human body, the door of a room, the tree in a garden, the water we drink and all those things that make up the universe. If we can tell how the atoms of a these systems interact and move over time, then we can understand, describe and predict the behaviour of these systems. One way to achieve this is the use of the molecular dynamics simulation technique. It is a computational technique that is used to describe the structure and properties of a given system and predict its behaviour by studying the physical movements of atoms and molecules over a period of time. This technique has become a very powerful scientific tool that is used in different fields and hence gained a lot of interest in recent days. In this thesis, we presented the theoretical background of molecular dynamics and implemented this concept in MATLAB. We used our MATLAB implementation to perform a comparative study between the Velocity Verlet and Euler's integration algorithms in terms of their computational speed. We then applied the concept of MD to protein folding.

Background to the Study

Computer simulations was first performed in the early 1950s after the first powerful electronic computers were allowed to be used for non-military research. Metropolis et al. (1953) performed the first molecular simulation for calculation of the equations of the state of rigid sphere in 2-dimensions using the Monte Carlo simulations. This was later followed by Alder & Wainwright (1957) who performed the first ever molecular dynamics simulation. Since then, computer simulation has been applied to study some properties of many physical systems and have become a very powerful research tool as a result of innovations in computer technology and improvement of algorithms.

Computer simulation is considered as a useful and a very powerful research tool for the modelling of many systems in material sciences, physics, biology, chemistry, engineering, economics, psychology and the social sciences. Computer simulation involve the use of computer program to simulate the outcome of a mathematical model of a given physical system to explore the mathematical prediction of the physical process. One use of experiments is to test for the validity of theories. In some cases, the experiment does not agree with the theory and sometimes it may agree but a wrong comparison between the experiment and the theory might have been made or the experimental results and the theoretical calculations may both be wrong (Allan & Slobodan, 2016). Computer simulations can be used to bridge this difference between experiments and theories or to validate the theoretical calculations and the experimental results. There are some physical concepts that is applicable to important biological phenomena like quantum, classical and statistical mechanics which lead to some equations that does not have an analytic solution and in such cases computer simulations can give the “exact” results or provide enough inputs or data about the microscopic properties of such complex systems (Meller, 2001; Kumar & Maiti, 2011). Computer simulations are very useful in studying the physical properties of systems whose experiments are expensive, impossible or too dangerous to be carried out.

The two most common simulation techniques in modelling of molecular dynamics are the Monte Carlo (MC) and molecular dynamics (MD) simulations. The Monte Carlo (MC) simulation technique is a stochastic approach that make use of the probability concept to mimic the behaviour of complex systems. It involves the use of computer to generate random numbers and estimate the thermodynamic properties of a given system of interest. On the other hand, molecular dynamics simulation which will be discussed in detail in this research involves the prediction of how atoms and molecules will be moving according to time, by solving the second equation of motion numerically (Allen & Tildesley, 1989).

Molecular Dynamics Simulation

Molecular dynamics simulation was first developed by Alder & Wainwright (1957) and which was followed by Rahman (1964) in late 1950s and the early part of 1960s respectively. Molecular dynamics simulations was mostly used in statistical mechanics but as a result of computer and technological advancement, it is now applied in many areas such as engineering, physical, biological and material sciences and for modelling of biomolecules such as nucleic acids or proteins. Molecular dynamics gives a significant microscopic insight into many physical problems and also provide an alternative route in cases where experiments are difficult or dangerous to be performed (Kumar & Maiti, 2011). Although molecular dynamics simulation basically involve computing the equilibrium and transport properties of a classical many-body systems (Frenkel & Smit, 1996), researchers have developed methods to solve the dynamical evolution of non-equilibrium processes. These methods first appeared in the early 1970s as an alternative to equilibrium simulations for computing transport coefficients (Heile, 1992).

Molecular dynamics is used as a computational "experiment" to study the physical movements of atoms and molecules which are allowed to evolve and then observations made based on its evolution. Molecular dynamics generates the dynamical trajectories of atoms and molecules by numerically solving the classical equations of motion (Newton's Equation) for a system of interacting particles, where forces between the particles and their potential energy are defined by molecular mechanic force fields. There are three basic ingredients that are necessary in performing molecular dynamics simulations; we must first define the law describing the mutual interactions between the atoms in the system, then numerically integrate the equations of motion for the atoms in the system using an appropriate algorithm and finally, in order to solve the equations of motion, we must define the appropriate initial conditions (initial positions and velocities for all atoms in the system) in the integration scheme (Hernandez, 2008).

Molecular dynamics can be applied to wide range of problems in different research fields. In statistical mechanics and physics, molecular dynamics is used to study correlated many-body motion, theory of liquids, liquid-to-glass transition, phase transitions, structure and properties of small clusters. It is also used in material science to study microscopic mechanisms of fracture, defects in crystals, extreme conditions, melting and glass properties. It can also be applied to chemistry and biochemistry to study molecular structures, reactions, drug design, vibrational relaxation and energy transfer, structure of membranes and dynamics of large biomolecules.

This concept provides an insight into biological phenomena such as the role of flexibility in ligand binding, catalytic cleavage of the peptide bond by chymotrypsin or the light-induced charge transfer in the photosynthetic reaction. It is being used to determine protein structures from Nuclear Magnetic Resonance (NMR), to define protein X-ray crystal structures faster from poorer starting models and to calculate the free energy changes resulting from mutations in proteins (Karplus & Petsko, 1990).

In recent days, molecular dynamics is also being used as a computational tool for drug discovery. One important example is the discovery and development of several potent and selective inhibitors of human immunodeficiency virus (HIV) protease as drugs against HIV infection (Meller, 2001).

Free-Energy Perturbation (FEP) using molecular dynamics technique is successfully employed to predict the affinity of a novel HIV-1 peptide inhibitor (Ferguson et al., 1991) and have now become an excellent testing ground for the whole FEP methodology (Vivo et al., 2016).

Molecular Dynamics Ensembles

In the molecular dynamics simulation method, the energy, volume, and number of particles of the system are constant for a particular simulation. In some situations, it is desirable to perform simulations of the material for specific values of temperature and pressure or under conditions of fluctuating energy and volume of the material.

An ensemble is a collection of possible systems which have different microscopic states but have identical macroscopic or thermodynamic state. That is, it is simply an assembly of microstates to at least one extensive constrain. Some of the ensembles used in molecular dynamics are Microcanonical Ensemble (NVE), Canonical Ensemble (NVT) and Isothermal-Isobaric (NPT) ensemble.

Microcanonical Ensemble(NVE)

Microcanonical ensemble is a thermodynamic state which is characterized by fixed number of particles (N), fixed system volume (V) and a fixed energy (E). Microcanonical ensemble is an isolated system with defined energy. Thus, the system exchanges neither energy nor particles with its environment, so that the energy of the system remain constant throughout the process.

Canonical Ensemble (NVT)

Canonical ensemble is the assembly of systems whose thermodynamic state is characterized by fixed number of particles (N), fixed system volume (V) and a fixed temperature (T). In the NVT, the energy of the microstates can fluctuate and the energy of the endothermic and exothermic process is exchanged with the thermostat. There are different ways of controlling the temperature. This can be done by rescaling the velocities or by a heat bath. This ensemble is useful for treating the actual experimental system which generally has a fixed number of particles, fixed volume and fixed temperature.

Isothermal-Isobaric (NPT) ensemble

Isothermal-isobaric (NPT) ensemble involves maintaining a constant number of particles (N), constant pressure (P) and constant temperature (T). It describes system in contact with thermostat at temperature (T) and a bariostat at pressure (P). The total number of particles (N) is kept constant while the total energy (E) and volume (V) fluctuate at thermal equilibrium.

Molecular Dynamics Model Potentials

In order to use molecular dynamics, the rules or laws that govern the interaction of atoms in the system have to be defined. Potentials refer to these rules or laws of interactions between particles. The potential function $U(r_1, r_2, \ldots, r_N)$ describes how energy of a system of N atoms depends on the coordinates of the atoms $r_1, r_2, \ldots, r_N$. In selecting potentials for a system, one should consider the accuracy, transferability and computational speed of the potential. The results of simulations will be realistic only if the potential energy function is able to reproduce the properties of interest as closely as possible. The potential should also be able to study many systems under different conditions. The potential should also have a simple functional form to boost the computational speed.

The Lennard-Jones potential (Lennard-Jones, 1924) is the simplest and most common pair potential used in molecular dynamics. It is mostly used to model noble gases and also to describe non-bounded type interactions in more complex molecular systems. It is often used to model general effects rather than properties of specific materials. The Tersoff potential (Tersoff, 1986, 1988) is a three-body potential which was originally used to simulate covalent materials such as carbon, silicon and germanium and has since been used to model many other substances. The Tersoff potential is based on the concept of bond strength on local environment, that is, an atom with many neighbors forms weaker bond than an atom with few neighbors.

Some other potentials include the Stillinger-Weber potential which is widely used to model silicon, the Embedded Atom Method (EAM) which is design to simulate metals such as Cu, Ag and Pt, the Tight-Binding Second Moment Approximation (TBSMA) potential and the Morse potential.

In this research, the Lennard-Jones model potential was used to simulate a system of atoms. The system size was varied to accommodate different amount of atoms at different thermodynamic conditions.

Statement of the Problem

Revolutionary advancements in computer technology and algorithmic improvements have made computer simulation a useful and very powerful research tool for mathematical modelling of many natural systems in physics, engineering, economics, psychology and social sciences. Every biological or chemical process has a physical representation but this sometimes lead to complex systems which need to be studied by computer simulation methods. Molecular dynamics (MD) simulation is one of these methods and it is used to gain insight into the macroscopic world of life on the molecular level (microscopic level) by evaluating the movements of particles caused by their interactions. Molecular dynamics offers scientists the opportunity to observe processes at high resolution both in space and time.

Researchers have used molecular dynamics simulations to observe and experiment with models of molecules in order to understand chemical and biological processes and predict macroscopic properties by detailed knowledge of atomic movements caused by their interactions. The method can be used to study wide range of problems; determination of protein structures from Nuclear Magnetic Resonance(NMR) (Karplus & Petsko, 1990; Patodia et al., 2014), computational tool for drug discovery (Meller, 2001; Ferguson et al., 1991; Vivo et al., 2016), DNA simulation (Beveridge et al., 1995; Wan et al., 2013) etc.

Although molecular dynamics simulations have been in existence for a long time, its scope and applications are inexhaustible. Most of the current studies that are done with molecular dynamics uses existing packages such as AMBER, CHARMM, GROMACS, LAMMPS, etc. These packages are like black boxes to people who are new to molecular dynamics since they assume that users should already know the concept of molecular dynamics before using them (packages) and hence do not explain the basic processes to beginners in the field. This research sought to bridge this gap between the concept and the use of these packages for a beginner. In other words, the research sought to present the concept, build a MATLAB code from scratch, run a typical molecular dynamics simulation and using one of the packages for an application study.

The Velocity Verlet algorithm, the integration method used in this molecular dynamics study, have been considered ”better” than most other numerical integration methods in molecular dynamics in terms of accuracy, stability, computational time and energy conservation. In this thesis, a comparative study of the Velocity Verlet and Euler’s algorithms was conducted to verify this claim in terms of their computational speed using the MATLAB code built.

Significance of the Study

Presenting the concept of molecular dynamics simulation and building a MATLAB code will help beginners to understand the process involved in molecular dynamics simulation and further be built on and applied to study various physical, biological and other phenomenons.

The comparative study of the Velocity Verlet and Euler method in terms of their computational speed will enable researchers to make a better choice of algorithms especially if they are interested in the speed of the algorithm.

Research Objectives

The main objective of this research was to bridge the gap between a beginner of molecular dynamics concept and packages usage.

Specifically;

(i) we developed a MATLAB implementation of molecular dynamics following the theory.

(ii) we run some simulations from the implementation, calculated and discussed some physical properties from a typical MD simulation.

(iii) we performed a comparative study of Velocity Verlet and Euler's algorithm in terms of their computational speed.

(iv) we used AMBER package to perform MD simulation on small protein (peptide) of 20 amino acids sequence to determine its folded structure.

Delimitation

In this thesis, we used dimensionless units (reduced units) and makes the assumption that ϵ (the energy constant) and σ (the distance where the potential is zero) are both equal to 1.

Definitions of Terms

Some of the terms or concepts used in this thesis are introduced in this section.

Definition 1 (Mass of a substance)

Mass as used in this research refer to the mass of an atom which is a count of the total number of protons and neutrons in the nucleus of an atom.

Definition 2 (Boltzmann constant (K_B))

The Boltzmann constant (K_B) is a physical constant that defines the relation between absolute temperature and kinetic energy contained in each molecule of an ideal gas. It provides a measure of the amount of energy (heat) corresponding to the random thermal motions of particles making up a substance.

Definition 3 (Temperature)

Temperature T is a fundamental concept in physics used to measure the hotness or coldness of macroscopic objects. It represents the intensity of thermal motion of molecules in microscopic theory. Temperature measures the average kinetic energy of the particles in a system and thus strongly influences molecular dynamics simulation in which the velocities of atoms are continuously adjusted according to the various temperature-controlled algorithms. It can be defined in terms kinetic energy KE through the relation

$$T = \frac{2}{3}\frac{KE}{NK_B} \tag{1.1}$$

where N and K_B are the number of particles and Boltzmann constant respectively.

Definition 4 (Conservation of energy)

The law of conservation of energy explains that energy can neither be created nor destroyed; rather, it can be transformed from one form to another. This means, total energy of a system will remain constant. Total energy obtained from the simulation system is the sum of the total potential energy (U) and kinetic energy (KE) of the particles. Thus,

$$TE = KE + U. \tag{1.2}$$

The potential energy (U) is the energy that an object possessed because of its position relative to some other objects in the system. The potential energy of the systems is thus defined by the interatomic interaction potential. The kinetic energy (KE) of an object or particle is the energy that it possesses due to its motion. It is directly proportional to the mass of an object and to the square of its velocity. Thus, an object with mass m and velocity v have a kinetic energy given as

$$KE = \frac{1}{2}mv^2. \tag{1.3}$$

Definition 5 (Force of interaction)

A force is a vector quantity which describes a push or a pull upon an object, resulting from the object's interaction with another object. A force can maintain or alter the motion of an object or distort it, that is; it can cause an object with mass to change its velocity (accelerate). Velocity of an object describes the rate of change of its position with respect to time while acceleration is the rate of change of velocity with respect to time. That is

$$\begin{aligned} \mathbf{v}(t) &= \frac{dr}{dt}, && (1.4) \\ \mathbf{a}(t) &= \frac{d^2r}{dt^2}. && (1.5) \end{aligned}$$

where $\mathbf{v}(t)$ and $\mathbf{a}(t)$ are respectively the velocity and acceleration with respect to time t.

Definition 6 (Newton's Laws of Motion)

The First Law of Motion states that a body at rest will remain at rest and a body moving at a constant speed in a straight line will remain moving at a constant speed in a straight line unless it is acted upon by an external force. Thus, an object cannot start, stop, or change direction without an external force acting on it. Newton's First Law of Motion is also known as the Law of Inertia.

The Newton's Second Law states that the time rate of change of the momentum of a body is equal in both magnitude and direction to the force imposed on it. Thus, force acting on an object is equal to the mass of that object times its acceleration. This law allows quantitative calculations of dynamics, thus, a quantitative description of the changes that a force can produce on the motion of a body. This is written in mathematical form as

$$\mathbf{F} = m\mathbf{a}, \tag{1.6}$$

where $\mathbf{F}$ is force, m is mass, and $\mathbf{a}$ is acceleration.

Newton's third law states that for every action, there is an equal and opposite reaction. That is, when two bodies interact, they apply forces to one another that are equal in magnitude and opposite in direction. When objects i and j interact with each other, they exert forces upon each other. If for example an object i exerts a force F_{ij} on object j, then object j also exerts a force F_{ji} on object i, and the two forces are equal in magnitude and opposite in direction. Thus,

$$F_{ij} = -F_{ji}. \tag{1.7}$$

These two forces F_{ij} and F_{ji} are called action and reaction forces respectively.

Definition 7 (Atoms and molecules)

An atom is the smallest possible unit of matter that retains all of the chemical properties of an element. Atoms are composed of particles called the subatomic particles. The three subatomic particles of atoms are protons, neutrons and electrons. Protons have a positive electric charge while electrons are negatively charged and neutrons are neutral. The protons and neutrons are contained in a dense core called the nucleus which is at the centre of the atom and the electrons are located in series of outer shells or orbits. A group of two or more atoms of the same element or of different elements which are chemically bonded together forms a molecule.

Organisation of the Study

In this chapter, the background of the study is provided and some concepts in molecular dynamics simulation introduced. The problem the research addresses and the significance of this study is then described. This is followed by objectives of the research, definition of some terms and how this thesis is organised. The rest of this research work is organised as follows:

In chapter two, we discuss and review related literature on the background of molecular dynamics simulation. We look at the literature on intermolecular potentials, numerical integration methods and some applications and other aspects of molecular dynamics simulation.

In chapter three, we present the theoretical background of molecular dynamics. The molecular dynamics algorithms, the model potential (the Lennard-Jones potential) and the numerical integration of the equations of motion is discussed in this chapter.

The molecular dynamics simulation is performed and the results discussed in chapter four. In this chapter, we present the results from a MATLAB implementation of typical MD simulation where we calculate and discuss some dynamical properties from the simulation. We use the MATLAB implementation to perform a comparative study of the computational speed between Velocity Verlet and Euler's algorithms. In this same chapter, we present an application of molecular dynamics simulations to protein folding using AMBER package. We finally summarise and draw conclusions from the results of this study and give recommendations for future studies in chapter five.

Chapter Summary

We gave the background of the study and an introduction of some concepts in molecular dynamics simulation in this chapter. We discussed the problem we intend to study and the significance of the studies. We went ahead to state the objectives of our research and give the definition of some terms. How this research is structured is also presented in this chapter.

CHAPTER TWO

LITERATURE REVIEW

Introduction

In this chapter, we discuss and review related research on a number of aspects of molecular dynamics simulation. We first consider the background and general principles of molecular dynamics simulation. After which we consider research on applications of molecular dynamics and other related works.

Molecular dynamics is a computer simulation method that calculates the time dependent behavior of a molecular system. The first simulation using molecular dynamics started in the late 1950s when Alder & Wainwright (1957) used this method to study the interaction of hard spheres in which they gained useful information in relation to the behaviour of liquids. They studied a solid-fluid transition in a system which consist of hard spheres. This was followed by the first ever simulation to have been performed with a realistic potential to study liquid argon by Rahman (1964). He considered a system containing 864 particles and made the assumption that the attractive and repulsive interaction of the particles follow the Lennard-Jones potential. Many recent research using molecular dynamics simulation technique still employ the methods and analysis of the simulations reported by this study. Based on the developments from Rahman (1964), the first molecular dynamics simulation of a real system was performed by Rahman & Stillinger (1971). They simulated a liquid water which was made up of 216 rigid molecules by employing the techniques of molecular dynamics simulations. After these developments, many researchers have explored molecular dynamics simulation and used it to gain understanding of many problems that result from physics, chemistry, engineering, material sciences, bioinformatics, social sciences and many more.

Molecular dynamics simulation simply involve solving the second equation of motion numerically. The main goal of molecular dynamics simulation is to simulate given atoms and predict how the atoms are moving with respect to time or what the displacement of all the atoms will be at different time. To achieve this, we need to define the initial positions of the atoms and the initial velocities of those atoms. The second and a very key ingredient that will be needed is the interatomic potential. This defines the potential energy between each pair of the atoms. The third ingredient required is the numerical integration algorithm that will solve the equations of motion to predict the next positions and velocities of the atoms (Hernandez, 2008).

Interatomic Potentials

Molecular dynamics is based on interatomic potentials. Defining the right interatomic potential is the most crucial part of molecular dynamics simulation and modelling. It essentially determines whether the numerical integration and algorithm design will be simple or complex and hence the accuracy of the results from the simulations. There is a direct relationship between the reliability of the simulation results and the correctness of the interaction potential function used to define how the particles will interact. Hence, the proper choice of the interaction potential function is absolutely necessary in molecular dynamics simulation. There are many classical potentials that can be used to define the interatomic interactions of a given system but we will review the common model potentials and some of their extensions.

The embedded-atom method(EAM) is of one of these potentials and it is widely used to describe the interactions among metallic molecules (Daw & Baskes, 1983, 1984). It is particularly appropriate for describing the interatomic interactions of many pure metals such as silver (Ag), platinum (Pt), copper (Cu) and gold (Au). It can also be used to approximate alloys.

This function is given as

$$E_i = F_\mu \left(\sum_{i \neq j} \rho_\Theta (r_{ij}) \right) + \frac{1}{2} \sum_{i \neq j} \phi_{\mu\Theta} (r_{ij}) \tag{2.1}$$

where F is an embedding function that represents the energy required to place atom i of type μ into the electron cloud, r_{ij} represent the distance between atoms i and j, the term $\phi_{\mu\Theta}$ represent a pair-wise potential function and ρ_Θ is the contribution to the electron charge density from atom j of type Θ at the location of atom i.

The Tersoff model potential (Tersoff, 1986, 1988) is a three-body component function that takes into consideration the covalent bond between atoms. It was initially proposed and used to simulate covalent bond materials like germanium, silicon and carbon but is now being used to to study many different materials. This interatomic potential on the idea of bond strength of atoms on local environment in which case there will be a weaker bond formed by an atom having many neighbors compared to an atom having fewer neighbors. The function of Tersoff potential of a given systems is given as

$$E = \sum_i U_1(r_i) + \sum_{i<j} U_2(r_i, r_j) + \sum_{i<j<k} U_3(r_i, r_j, r_k) + \dots \tag{2.2}$$

where U_1 is the one body potential, U_2 is the two body, U_3 is the three body potential, etc and this depends on the positions of each atom. The first term of the function is defines external forces and this means it will not be included if we are considering only the interatomic forces. The pair potential (U_2) alone may be good for describing closed packed structures such as liquid noble gases like Krypton(Kr), Argon(Ar)and Xenon(Xe) but not good to be used for systems having strong covalent bond with more open structure. There is no good pair potential that will stabilised the diamond structure and hence the third term of the the expression (the three body component U_3) have to be included to take of it. Combining the two body component U_2 and the three body component U_3 can allow enough accuracy in describing the given real physical system (Biswas & Hamann, 1987; Stillinger & Weber, 1985). Modifications have been made by researchers to this potential function to study various materials; Los et al. (2017) extended the this potential function to describe boron nitride (BN-ExTeP) and used it to perform a larger scale atomistic simulation, it was modified by Sha et al. (2013) to study pure and hydrogenated diamond-like carbons and a modification of the cut-off function for this potential was also presented by Rajasekaran et al. (2016) to overcome overestimation and also used to study the realistic mechanical behaviour of one sheet of graphene.

The Lennard-Jones potential was first presented in the year 1924 by the British physicist Sir Edward Lennard-Jones. This interatomic model potential is used for the description of the interactions of two neutral particles using a comparatively simple mathematical model. When two neutral atoms or molecules interact, they feel both repulsive and attractive forces which is dependent on their relative proximity and polarizability. This potential models an attraction component by a contribution that is proportional to r^{-6} and the repulsion component by a contribution that is proportional to r^{-12} (Lennard-Jones, 1924). The most commonly used L-J interatomic potential U between a given two atoms or moles is defined as

$$U(r) \quad = \quad 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], \tag{2.3}$$

where ε represent the energy constant or the depth of the potential well and which determines the strength of attraction between the two particles. The variable σ represent the distance when the interaction potential between the two atoms or molecules goes to zero and r represent the distance of separation between the centres of the given two atoms. The L-J potential is widely used because it is simple and can easily be computed. It is mostly used to model noble gases (such as neon, argon, krypton and xenon) and metallic liquids (Jensen et al., 1973). It has played a central role in development of molecular dynamics methods and has widely been used to investigate some important concepts such as phase transitions, the effect of surfaces, small clusters etc. In particular, it has been used to study homogeneous nucleation of liquid by Swope et al. (1982), the dynamical properties of diffusion in liquids by Dzugutov (2001), to study some fundamental processes like melting by Jensen et al. (1973) and also to study the structure and dynamical properties of heavy metals ion like Cd^{2+}, Ni^{2+} and Zn^{2+} in an aqueous solution (Dezfoli et al., 2015).

The Lennard-Jones potential have in many cases been modified to accommodate the complexities of interactions to study certain real materials. Oh (2013) proposed a three form parameter Lennard-Jones interaction potential function to calculate some transport properties (such as diffusion coefficient, thermal conductivity and viscosity), and the thermodynamical property (the virial coefficient) of noble gases (xenon (Xe), krypton (Kr), argon (Ar), neon (Ne) and helium (He)) and their mixtures at lower density. The reduced temperature T^* was modified by an introduction of a reduced temperature-correction parameter τ to it. This is given as

$$T^* = \frac{K_B(T-\tau)}{\varepsilon} \tag{2.4}$$

where K_B represent the Boltzmann constant and ε is the depth of the potential well as defined in the Lennard-Jones interaction potential. The parameter sets generated from this study have more physical importance than parameters obtained from a second virial coefficient method. This model produces better results as compared to the normal two-parameter L-J model potential. Also, Semironi & Azimian (2010) used a modified Lennard-Jones model potential which was proposed by Stoddard & Ford (1973) to carry out a molecular dynamics simulation to study molecules of liquid argon and platinum. They considered two groups of molecules. The first group was used for the liquid and its vapor and the other group was used for the solid surface. The results from their simulations when compared with the experimental data showed a good prediction.

Integration Algorithms

The Verlet integration algorithm which is commonly attributed to the French physicist Loup Verlet though it was first published in 1791 by Joseph Delambre (Levesque & Verlet, 1993). It is a numerical algorithm that is being used for the integration of the Newton's equations of motion. This method is mostly used for the generation of trajectories of particles in molecular dynamics simulation. Verlet (1967) in the 1960s discusses this algorithm and its application to molecular dynamics simulation. In his paper, he integrated the equations of motion of a given system containing 864 particles. He used the Lennard-Jones potential to describe the interactions of the particles. This was done for different values of density and temperature to a given fluid state. The Verlet method performs the integration of the equation of motions using the algorithm;

$$r(t+\Delta t) \quad = \quad 2r(t) - r(t-\Delta t) + \mathbf{a}(t)\Delta t^2 + \mathcal{O}(\Delta t^4), \tag{2.5}$$

where $\mathbf{a(t)}$ is the acceleration, $r(t)$ is the position and Δt is the time increment. The verlet algorithm gives better stability and energy conservation as compared to the much simpler Euler method. This algorithm is considered produce simulation results with good accuracy. The velocities of this algorithm are not generated directly and hence the speed and position of the current time step will not be available at the same time. The velocity $\mathbf{v}(\mathbf{t})$ can only be obtained after the time step $r(t + \Delta t)$ is known and that will be a time step after. This is major problem of this algorithm. Although velocities will not be required to compute the trajectories, they are important especially when computing system properties such as kinetic energy and temperature. This algorithm also does not self-start.

In an attempt to rectify some of the problems linked with the original Verlet algorithm, the Leap-frog algorithm was proposed. The Leap-Frog and the Verlet integration algorithms have been proven to be equivalent (Gunsteren & Berendsen, 1988). In the Leap-Frog algorithm, the position r_i is obtained at integer time steps while the velocity v_i is obtained at half integer time steps. This means that the velocities will leap-over the positions and the positions also leap-over the velocities and this gives the name Leap-Frog. The positions and velocities of the Leap-frog algorithm are generated using the equations;

$$r(t + \Delta t) = r(t) + \mathbf{v}(t + \frac{1}{2}\Delta t)\Delta t \tag{2.6}$$

$$\mathbf{v}(t + \frac{1}{2}\Delta t) = \mathbf{v}(t - \frac{1}{2}\Delta t) + \mathbf{a}(t)\Delta t \tag{2.7}$$

and the velocities are then obtained at time t using

$$\mathbf{v}(t) = \frac{\mathbf{v}(t + \frac{1}{2}\Delta t) - \mathbf{v}(t - \frac{1}{2}\Delta t)}{2}. \tag{2.8}$$

The Leap-frog method has the advantage of explicitly calculating the velocities but this is still not done at the same time as the positions are calculated. It is stable, however, this algorithm still has the problem of self-starting.

As mentioned above, Leap-Frog method does not produce the velocities and positions at the same time. Another modified form of the Verlet algorithm has been proposed by Swope et al. (1982) as the Velocity Verlet algorithm. This method is similar to the Leap-Frog algorithm except that it calculates the positions, acceleration and velocities at the same time without losing its accuracy. It thus give a better trade-off between the number of times the algorithm calculate forces and the accuracy of the integration algorithm. It calculates the velocities and positions at integer time steps but still calculates the intermediate velocities at a half integer time steps. The equations for predicting the next positions and velocities of a given system using the Velocity Verlet method are given as,

$$r(t+\Delta t) = r(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2 + \mathcal{O}(\Delta t^3) \tag{2.9}$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \frac{\mathbf{a}(t) + \mathbf{a}(t+\Delta t)}{2}\Delta t + \mathcal{O}(\Delta t^3). \tag{2.10}$$

where $\mathbf{a}(t)$ is the acceleration, $\mathbf{v}(\mathbf{t})$ is the velocity, $r(t)$ is the position and Δt is the time increment. The Velocity-Verlet algorithm is algebraically equivalent to the Verlet scheme because Equation 2.5 can be obtained from Equation 2.9 by eliminating the velocities in the equation updating the positions. Velocity-Verlet has perhaps become most commonly used molecular dynamics algorithm because it is simple and yet very stable.

Applications

As a result of improvement in the design of numerical algorithms and advancement of computers in terms of memory, speed, etc, researchers have applied the molecular dynamics technique to study the dynamics of many real systems in many fields such as engineering, physical, biological and material sciences.

One area of the applications of MD simulations to macromolecular systems is protein. Protein performs very important function in every aspect of cellular life. The use of experimental methods for the study of protein dynamics can fall short and hence alternative methods are needed. Molecular dynamics simulations technique is one of the alternative methods that is being used to study the dynamics of protein. MD simulations is an important tool for the exploration of the structure and dynamics of protein. They have been used to study enzymatic catalysis and protein folding by Karplus & Kuriyan (2005). They described some results of the Src family and F_1 ATPase molecular motor of signaling proteins. Protein-Ligand interaction was also investigated by Swegat et al. (2003) using molecular dynamics simulation. They considered insulin-phenol complex formation and its dissociation. Their simulation results agreed with experimental results. Patodia et al. (2014) also studied protein molecules and residual dipolar coupling using molecular dynamics simulation alongside nuclear magnetic resonance (NMR) to validate the simulation results.

Another important area that we can apply molecular dynamics is the study of drug discovery and design. The protein function is very essential in the discovery and design of drugs and since molecular dynamics simulation can be used to understand the protein function, it thus, can be used as a powerful tool in drug discovery. Vivo et al. (2016) discussed molecular dynamics simulation as a tool to pharmaceutical research with emphasis to drug discovery. They discussed the optimisation of drug residence times and target affinity in improving drug efficacy. The role of molecular dynamics simulations to drug discovery was also examined by Durrant & McCammon (2011). They examined the role played by atomistic simulations of macromolecular receptors such as protein in the discovery of drugs. Also, Seco et al. (2009) used molecular dynamics simulation to predict low affinity binding, protein-protein and druggable sites which is usually done using nuclear magnetic resonance (NMR) and X-ray crystallography experiments.

Molecular dynamics simulation has become a very important technique used to obtain flexibility and structural data on DNA. The method has a good predictive power which has helped in dealing with systems of high complexity that has to do with studying realistic systems of DNA. Atomistic molecular dynamics simulation is also an important method for some parameter generations in the mesoscopic models of DNA flexibility (Perez et al., 2012). Chakraborty et al. (2013) investigated the microscopic structure and flexibility strand and the surrounding molecules of a DNA. They carried out a molecular dynamics simulation of ss-DNA dodecamer, 5′-CGCGAATTCGCG-3′ in an aqueous medium at a room temperature. Beveridge et al. (1995) also performed analysis of a molecular dynamics simulation of a free DNA in solution and DNA in complex with the protein and compared the results with data from X-ray crystallography experiment to check the validity of a theoretical model.

Chapter Summary

In this chapter, the background of molecular dynamics simulation has been discussed and related literature reviewed. We looked at literature on some of the common intermolecular potentials. We particularly discussed the Embedded-Atoms Method(EAM), Tersoff model potential, the Lennard-Jones model potential and their possible extensions. We went ahead to review literature on some of the numerical integration methods used in molecular dynamics simulations and provided some applications of molecular dynamics to protein, drug design, DNA etc.

Molecular dynamics simulation simply involve solving the second equation of motion numerically. The aim of performing molecular dynamics simulation is to be able to predict how given atoms are moving with respect to time. The three main ingredients that dictate the quality of molecular dynamics simulation is having the right initial configuration, the right description of the energy between the atoms (right interatomic potential function) and to choose the right numerical integration algorithm.

Although molecular dynamics simulations have been in existence for long time, its scope and applications are inexhaustible. In this thesis, molecular dynamics simulation was implemented in MATLAB. The Lennard Jones interatomic potential was used in this implementation. The Velocity Verlet and Euler's algorithm was used to carry out the integration of the equations of motion and in so doing, a comparison between their computational speed was made. The application of molecular dynamics concept to protein folding was also studied.

CHAPTER THREE

RESEARCH METHODS

Introduction

In this chapter, we give a description of the ingredients necessary for the performance of molecular dynamics simulation and the properties that can be calculated with this method. We discuss the theoretical background leading to molecular dynamics simulation. The molecular dynamics algorithms is explained and the numerical methods used in the simulation derived in this chapter. The model potential (the Lennard-Jones potential) is also presented in this chapter.

Equations of Motion

Molecular dynamics simulation is simply based on the Newton's second law of motion which states that force equals the rate of change of momentum; acceleration of a particle equals the force acting on it divided by its mass.

$$\mathbf{F} = m\mathbf{a} \tag{3.1}$$

where m and $\mathbf{a}$ respectively represent the mass and acceleration of the particle and $\mathbf{F}$ represent the force acting upon the particle. Molecular dynamics basically involves generating the atomic trajectories of a system of N particles(atoms) by numerically solving/integrating Newton's equation of motion for a system of N particles interacting with each other. This is done for a specific interatomic potential with given initial and boundary conditions satisfying certain thermodynamic constraints. Given a system of N interacting particles with position vector $\vec{r_i}$, $i = 1, 2, 3, \ldots, N$, the equation of motion is a second order differential equation given as

$$m\frac{d^2\vec{r_i}}{dt^2} = F_i(\vec{r_1}, \vec{r_2}, \ldots, \vec{r_N}) \tag{3.2}$$

where $\mathbf{a}_i = \dfrac{d^2\vec{r_i}}{dt^2}$ and F_i represent the force acting upon particle i.

Calculation of Force

The force on particle i can be derived from the potential energy function $U(r_1, r_2, r_3, \ldots, r_N)$ as the negative gradient of the potential energy.

$$F_i(\vec{r_1}, \vec{r_2}, \vec{r_3}, \ldots, \vec{r_N}) = -\nabla U(\vec{r_1}, \vec{r_2}, \vec{r_3}, \ldots, \vec{r_N}). \tag{3.3}$$

This implies the conservation of energy E, thus,

$$E = KE + U = constant \tag{3.4}$$

where KE is the kinetic energy.

The forces of particle i's interaction with the rest of the other particles is calculated as;

$$F_i = \sum_{j \neq i}^{N} f_{ij}, \tag{3.5}$$

where f_{ij} is the force of interaction of particle i with j and is given as the directional derivative,

$$f_{ij} = -\frac{dU(r_{ij})}{dr_{ij}} \cdot \frac{\vec{r_{ij}}}{r_{ij}}.$$

And by the Newton's third law of motion, $f_{ij} = -f_{ji}$.

Thus,

$$
\begin{aligned}
f_{ix} &= -\frac{\partial U(r_{ij})}{\partial r_{ij}} \cdot \frac{\partial r_{ij}}{\partial x_i},\\
f_{iy} &= -\frac{\partial U(r_{ij})}{\partial r_{ij}} \cdot \frac{\partial r_{ij}}{\partial y_i},\\
f_{iz} &= -\frac{\partial U(r_{ij})}{\partial r_{ij}} \cdot \frac{\partial r_{ij}}{\partial z_i}.
\end{aligned}
$$

From $r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$,

$$
\begin{aligned}
\frac{\partial r_{ij}}{\partial x_i} &= \frac{2(x_i - x_j)}{2\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}\\
&= \frac{x_i - x_j}{r_{ij}},\\
\frac{\partial r_{ij}}{\partial y_i} &= \frac{2(y_i - y_j)}{2\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}\\
&= \frac{y_i - y_j}{r_{ij}},\\
\frac{\partial r_{ij}}{\partial z_i} &= \frac{2(z_i - z_j)}{2\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}\\
&= \frac{z_i - z_j}{r_{ij}}.
\end{aligned}
$$

Now given the initial conditions $\vec{r}_i(0)$, $\dot{\vec{r}}_i(0)$ where $\vec{r}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$, $i = 1, 2, 3, ..., N$;

$$
\begin{aligned}
&\{x_1(0), x_2(0), \ldots, x_N(0)\} \quad \& \quad \{\dot{x}_1(0), \dot{x}_2(0), \ldots, \dot{x}_N(0)\}\\
&\{y_1(0), y_2(0), \ldots, y_N(0)\} \quad \& \quad \{\dot{y}_1(0), \dot{y}_2(0), \ldots, \dot{y}_N(0)\}\\
&\{z_1(0), z_2(0), \ldots, z_N(0)\} \quad \& \quad \{\dot{z}_1(0), \dot{z}_2(0), \ldots, \dot{z}_N(0)\},
\end{aligned}
$$

we know the initial forces

$$F_{r_i}(0) = \sum_{j \neq i}^{N} f_{ij}, \quad \text{where} \quad f_{ij} = -\frac{dU(r_{ij})}{dr_{ij}} \cdot \frac{\vec{r_{ij}}}{r_{ij}},$$

and then, we move each particle/atom due to this forces.

The Molecular Dynamics Algorithms

The forces of many-body system depends on the position of the particle which changes whenever there is an interaction or movement of the particle. This requires updating the positions and velocities with a stepwise numerical integration procedure and then recalculating the forces acting on the particle at each step. To perform a molecular dynamics simulation, we first define the initial configuration of the system, that is, the initial positions and velocities of the atoms. We then define the interaction potential function and calculate the interatomic forces at current time. We then move the atoms by numerically solving the equation of motion for all atoms in the system. The required boundary conditions are applied and we can now calculate and obtain the required output(physical quantities). This goes through an iterative process of predicting or updating the next positions of the atoms as required. A schematic of molecular dynamics algorithm is illustrated in Figure 1.

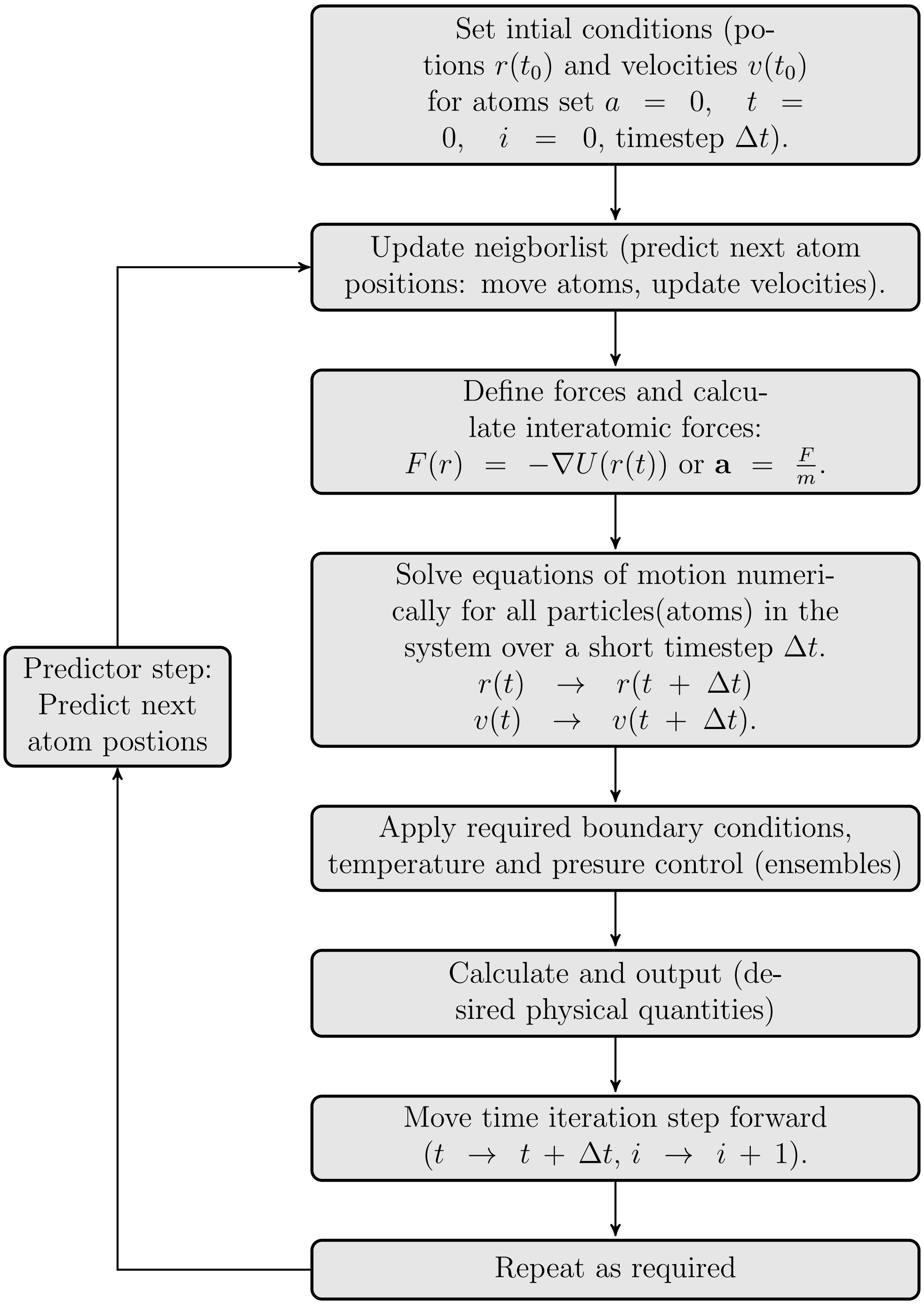

Set intial conditions (positions $r(t_0)$ and velocities $v(t_0)$ for atoms set $a = 0$, $t = 0$, $i = 0$, timestep Δt).
Update neigborlist (predict next atom positions: move atoms, update velocities).
Define forces and calculate interatomic forces: $F(r) = -\nabla U(r(t))$ or $\mathbf{a} = \frac{F}{m}$.
Solve equations of motion numerically for all particles(atoms) in the system over a short timestep Δt. $r(t) \rightarrow r(t + \Delta t)$ $v(t) \rightarrow v(t + \Delta t)$.
Apply required boundary conditions, temperature and presure control (ensembles)
Calculate and output (desired physical quantities)
Move time iteration step forward ($t \rightarrow t + \Delta t$, $i \rightarrow i + 1$).
Repeat as required
Predictor step: Predict next atom postions

Figure 1: Flow Chart Showing the Molecular Dynamics Simulation Algorithm

Numerical Integration of the Equations of Motion

The most time-consuming part of an MD calculation is the evaluation of forces. There are several algorithms for integrating the equations of motion. The choice may depend on the stability (conservation of energy), accuracy in evaluating energies and forces, speed and computational efficiency. Some of these algorithms which will be discussed in this research are the Euler, Verlet, the Verlet Leap-Frog and the Velocity Verlet algorithms. The Velocity Verlet algorithm has been recommended over Euler algorithm by most literature for molecular dynamics simulations. Euler algorithm is a first order accuracy while Velocity Verlet algorithm is of third order accuracy. Velocity Verlet algorithm offer a good stability for moderately large time steps than Euler algorithm (Heile, 1992). Euler's method is also not time reversible and suffers from energy drift (Frenkel & Smit, 1996). We seek in this research to compare the computational speed of Euler's algorithm and Velocity Verlet algorithm. We will achieve this by implementing these algorithms in MATLAB to simulate a system of N-atoms using the Lennard-Jones potential to describe the intermolecular interaction.

Euler Integration Algorithm

Euler algorithm is a numerical method of first order approximation which can be traced to Leonard Euler. It can be derived from the Taylor expansion

$$r(t+\Delta t) \quad = \quad r(t) + \frac{dr(t)}{dt}\Delta t + \frac{1}{2!}\frac{d^2r(t)}{dt^2}\Delta t^2 + \frac{1}{3!}\frac{d^3r(t)}{dt^3}\Delta t^3 + \cdots .$$

Considering only the first two terms and replacing $\frac{dr(t)}{dt} = \mathbf{v}(t)$ and $\frac{d\mathbf{v}(t)}{dt} = \mathbf{a}(t)$, we obtain the Euler method as

$$
\begin{aligned}
r(t+\Delta t) &= r(t)+\mathbf{v}(t)\Delta t, \\
\mathbf{v}(t+\Delta t) &= \mathbf{v}(t)+\mathbf{a}(t)\Delta t \\
&= \mathbf{v}(t)+\frac{F(t)}{m}\Delta t.
\end{aligned}
$$

It can be seen that the truncation error is of order $\mathcal{O}(\Delta t^2)$ and hence this method is accurate to first order. It is easy to implement but less accurate and less stable for some systems compared to other methods. The Euler algorithm is not time-reversible and phase-space preserving.

Verlet Algorithm

The verlet algorithm is the simplest and the most commonly used time integration in molecular dynamics simulation (Verlet, 1967). Solving Equation 3.2 using the Verlet algorithm is based on the Taylor series expansion. It simply involve writing two third-order Taylor expansions for the positions $r(t)$, one is forward in time $(r(t+\Delta t))$ and the other one backward in time $(r(t-\Delta t))$, where Δt is the integration time step. This is given as;

$$
\begin{aligned}
r(t+\Delta t) &= r(t)+\mathbf{v}(t)\Delta t+\frac{1}{2!}\mathbf{a}(t)\Delta t^2+\frac{1}{3!}\mathbf{b}(t)\Delta t^3+\mathcal{O}(\Delta t^4), \\
r(t-\Delta t) &= r(t)-\mathbf{v}(t)\Delta t+\frac{1}{2!}\mathbf{a}(t)\Delta t^2-\frac{1}{3!}\mathbf{b}(t)\Delta t^3+\mathcal{O}(\Delta t^4),
\end{aligned}
$$

where $r(t)$ is the position, $\mathbf{v}(t) = r'(t)$ is the velocity, $\mathbf{a}(t) = r''(t)$ is the acceleration and $\mathbf{b}(t)$ is the third derivative of $r(t)$ with respect to time.

Adding the two expansions leads to the result,

$$
r(t+\Delta t) = 2r(t)-r(t-\Delta t)+\mathbf{a}(t)\Delta t^2+\mathcal{O}(\Delta t^4). \tag{3.6}
$$

Since we are integrating Newton's equation, the acceleration $\mathbf{a}(t)$ can be written from Equation 3.2 as the force divided by the mass, and the force can also be written in terms of positions as in Equation 3.3. Thus,

$$\mathbf{a}(t) = -\frac{1}{m}\nabla U(r(t)). \tag{3.7}$$

As it can be seen in Equation 3.6, the truncated error of the algorithm is in the order of $\mathcal{O}(\Delta t^4)$ (fourth order). Velocities are not directly generated and hence the position and speed of the current time cannot be obtained at the same time in Verlet algorithm. This is the major problem of this algorithm especially when computing system properties such as kinetic energy and temperature since these properties cannot be obtained in the current state. Also, the algorithm is not self-starting since it estimates the next position $r(t + \Delta t)$ from the current position $r(t)$ and the previous position $r(t+\Delta t)$. The velocities are computed from the positions using an approximate formula based on the derivative of the position given as

$$\mathbf{v}(t) = \frac{r(t + \Delta t) - r(t - \Delta t)}{2\Delta t}.$$

To improve the implementation, modifications can be made to the Verlet algorithm by performing integration over half time step and this will lead to the Leap-Frog and Velocity Verlet algorithms.

Leap-Frog Algorithm

The Leap-Frog algorithm is obtained by simply performing algebraic manipulation of the Verlet algorithm. This algorithm produces the position r_i at integer time steps and velocity v_i at half integer time steps; it starts with the position r at time t and velocities at time $t + \frac{1}{2}\Delta t$, where Δt is the time step. Using this as the initial velocity, we can calculate the new position $r(t)$ at time $t + \Delta t$ and thus compute the velocities. In this way the positions and velocities "leap over" each other. The equations for the Leap-Frog algorithm is given as,

$$\begin{aligned} r(t+\Delta t) &= r(t)+\mathbf{v}(t+\frac{1}{2}\Delta t)\Delta t, \\ \mathbf{v}(t+\frac{1}{2}\Delta t) &= \mathbf{v}(t-\frac{1}{2}\Delta t)+\mathbf{a}(t)\Delta t. \end{aligned}$$

The velocity at current time step can be obtained as an average, given as

$$\mathbf{v}(t) = \frac{\mathbf{v}(t+\frac{1}{2}\Delta t)-\mathbf{v}(t-\frac{1}{2}\Delta t)}{2}.$$

This algorithm is stable and handles velocity better but it still has a problem of self-starting.

Velocity Verlet Algorithm

The major problem with the Verlet algorithm as noted above is that the position and speed of the current time cannot be obtained at the same time, so certain system properties such as kinetic energy, temperature and other information cannot be obtained in the current state. This is not good for actual integration process. Also, the Leap-Frog method which was introduced to improve the Verlet algorithm still does not handle the velocities in a completely satisfactory manner. Velocity Verlet algorithm is another modified form of the Verlet algorithm which is similar to the Leap-Frog algorithm except that the acceleration, velocity and position are calculated at the same time without any loss of precision. The Velocity Verlet is more convenient in handling actual integration process and hence widely used in molecular dynamics simulation of practical applications (Swope et al., 1982). It gives both position and velocity at integer time steps, yet the intermediate velocity at half integer time step. The velocity is calculated at both time t and at intermediate times $t+\frac{1}{2}\Delta t$, whereas the forces are only calculated at the full time-steps, t, $t+\Delta t$, $t+2\Delta t$ etc. The positions $r(t)$ and velocities $\mathbf{v}(t)$ are given as

$$\begin{aligned} r(t+\Delta t) &= r(t)+\mathbf{v}(t)\Delta t+\frac{1}{2}\mathbf{a}(t)\Delta t^2, && (3.8)\\ &= r(t)+\mathbf{v}(t)\Delta t+\frac{F(t)}{2m}\Delta t^2 && (3.9)\\ \mathbf{v}(t+\Delta t) &= \mathbf{v}(t)+\frac{\mathbf{a}(t)+\mathbf{a}(t+\Delta t)}{2}\Delta t && (3.10)\\ &= \mathbf{v}(t)+\frac{F(t)+F(t+\Delta t)}{2m}\Delta t. && (3.11) \end{aligned}$$

This algorithm gives a good trade-off between precision in the integration algorithm and the number of times forces are calculated. It requires lower memory and also very good when it comes to energy conservation. It is also a better choice in handling velocities.

The Velocity Verlet algorithm is equivalent to the Verlet algorithm and this can be shown as follows;

We first write Equation 3.8 for time $(t+2\Delta t)$, which will be given as

$$r(t+2\Delta t) = r(t+\Delta t)+\mathbf{v}(t+\Delta t)\Delta t+\frac{1}{2}\mathbf{a}(t+\Delta t)\Delta t^2. \quad (3.12)$$

We then subtract Equation 3.12 from Equation 3.8 to obtain

$$\begin{aligned} r(t+2\Delta t) &= 2r(t+\Delta t)-r(t)+\left[\mathbf{v}(t+\Delta t)-\mathbf{v}(t)\right]\Delta t\\ &\quad +\frac{1}{2}\left[\mathbf{a}(t+\Delta t)-\mathbf{a}(t)\right]\Delta t^2. && (3.13) \end{aligned}$$

Now, substituting Equation 3.10 into Equation 3.13, we obtain

$$r(t+2\Delta t) = 2r(t+\Delta t)-r(t)+\mathbf{a}(t+\Delta t)\Delta t^2 \quad (3.14)$$

which is the next position of the Verlet algorithm in Equation 3.6.

We will use the Euler and Velocity Verlet algorithms for the numerical integration in this research. The integration using the Velocity Verlet algorithm is performed as follows;

Given the initial conditions $r_i(0)$, $\dot{r_i}(0) = v_i(0)$, where $r_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$, $i = 1, 2, 3, ..., N$, then we can predict the new positions and velocities as follows;

At time Δt;

$$\begin{aligned} r_i(\Delta t) &= r_i(0) + v_i(0)\Delta t + \frac{F_{r_i}(0)}{2m_i}\Delta t^2, \\ v_{r_i}(\Delta t) &= v_{r_i}(0) + \frac{F_{r_i}(0) + F_{r_i}(\Delta t)}{2m_i}\Delta t. \end{aligned}$$

At time $2\Delta t$;

$$\begin{aligned} r_i(2\Delta t) &= r_i(\Delta t) + v_i(\Delta t)\Delta t + \frac{F_{r_i}(\Delta t)}{2m_i}\Delta t^2, \\ v_{r_i}(2\Delta t) &= v_{r_i}(\Delta t) + \frac{F_{r_i}(\Delta t) + F_{r_i}(2\Delta t)}{2m_i}\Delta t. \\ \vdots \quad & \vdots \quad \vdots \end{aligned}$$

At time $n\Delta t$;

$$\begin{aligned} r_i(n\Delta t) &= r_i((n-1)\Delta t) + v_i((n-1)\Delta t)\Delta t + \frac{F_{r_i}((n-1)\Delta t)}{2m_i}\Delta t^2, \\ v_{r_i}(n\Delta t) &= v_{r_i}((n-1)\Delta t) + \frac{F_{r_i}((n-1)\Delta t) + F_{r_i}(n\Delta t)}{2m_i}\Delta t. \end{aligned}$$

This will now generate the trajectory in phase space as,

$$\left\{\begin{pmatrix} x_i(t) \\ y_i(t) \\ z_i(t) \end{pmatrix}\right\}\left\{\begin{pmatrix} v_{x_i}(t) \\ v_{y_i}(t) \\ v_{z_i}(t) \end{pmatrix}\right\}, \quad i = 1, 2, 3, \dots N,$$

and we can calculate the average values over the trajectory as well as other properties.

The integration using the Euler's algorithm also follow the same way.

The Lennard-Jones(L-J) Potential

The Lennard-Jones potential is a simple mathematical model that describes the potential energy of interaction between two non-bonded atoms or molecules based on their distance of separation. This potential is an approximation to the true intermolecular energy curves. The two atoms or molecules feel both attractive and repulsive forces. The Lennard-Jones potential models the attraction component by a contribution that is proportional to r^{-6} and the repulsion component by a contribution that is proportional to r^{-12}. The sum of these forces results in the Lennard-Jones potential given as

$$U(r) = \sum_{i=1}^{N-1}\sum_{j>i}^{N} 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right], \tag{3.15}$$

where $U(r)$ is the intermolecular potential between the two atoms or moles, ε is the energy constant or the depth of the potential well and a measure of how strongly the two particles attract each other, σ is the distance at which the intermolecular potential between the two particles is zero and r is the distance of separation between centres of both particles.

The r^{-12} term, which is the repulsive term, models the repulsion between atoms at short distance due to Pauli repulsion: when there is overlapping of the electronic orbitals, the energy of the system rises sharply. This term could preferably be replaced by an exponential or even other functional forms which would also require larger computational effort. Also, the r^{-6} term, which is the attractive term at long-range, models the attraction between atoms at long ranges.This term is originated by van der Waals dispersion forces caused by the dipole-dipole interactions due to fluctuating dipoles. These are weak interactions but it provides the bonding character of closed-shell systems like rare gases such as Kr and Ar. Combining the repulsive term r^{-12} and the attractive term r^{-6} gives a potential well. This is shown in Figure 2.

In general, both the short-range repulsion and long-range attraction are necessary for describing the physical properties of solids and liquids which we know from everyday experience. The short-range repulsion is necessary to give the system a certain size or volume (density), without which the particles will collapse onto each other and the long-range attraction is also necessary for cohesion of the system, without which the particles will not stay together as they must in all condensed states of matter.

Differentiating the L-J potential with respect to r gives an expression for the net intermolecular force between two molecules. This intermolecular force may be attractive or repulsive, depending on the value of r. When r is very small, the molecules repel each other. At $r = \varepsilon$, $U(r) = 0$ and $U'(r) < 0$, at which point the potential is still repulsive and two atoms would repel each other if separated at this distance. The potential $U(r)$ will have a minimum at $r_e = 2^{\frac{1}{2}}\sigma$ and $U_e = -\varepsilon$ and when $r > r_e$, the potential switches from being repulsive to being attractive. As $r \to \infty$, the potential $U(r)$ is attractive and decays. Also, as $r \to 0$, the potential $U(r)$ is repulsive and blows up so quickly. Whereas the functional form of the attractive term has a clear physical justification, the repulsive term has no theoretical justification. It is used because it approximates the Pauli repulsion well and is more convenient due to the relative computing efficiency of calculating r^{-12} as the square of r^{-6} (Lennard-Jones, 1924; Atkins & de Paula, 2006).

For simplicity, take $\sigma = \varepsilon = 1$ and thus Equation 3.15 can be rewritten as

$$U = 4\left[\frac{1}{r^{12}} - \frac{1}{r^{6}}\right]. \tag{3.16}$$

The potential reaches minimum(equilibrium) at $r_e = 2^{\frac{1}{6}}\sigma = 2^{\frac{1}{6}}$, it decays as $r \to \infty$ and explode as $r \to 0$. As shown in Figure 2, the potential $U(r)$ vanishes at $r = \sigma = 1$ and has a depth $\varepsilon = 1$. Thus, the interaction energy rise abruptly when the atoms are close to each other, reaches a minimum at intermediate separation and decays to zero when the atoms are far from each other.

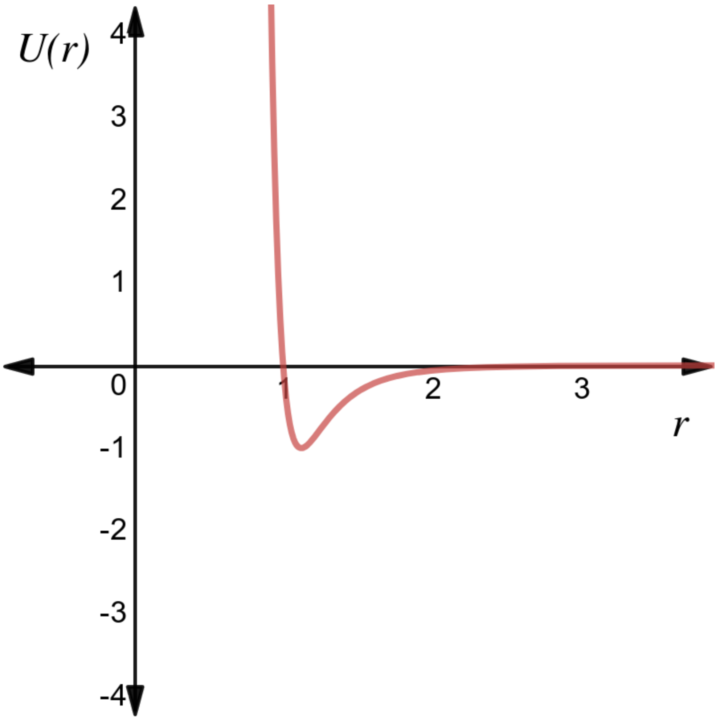


Figure 2: The Lennard-Jones Interatomic Potential $U(r)$ when $\sigma = \varepsilon = 1$. The Potential Reaches Minimum (Equilibrium) at $r_e = 2^{\frac{1}{6}}$, It Decays as $r \to \infty$ and Explode as $r \to 0$

Chapter Summary

The theoretical background leading to molecular dynamics simulation has been presented in this chapter and the molecular dynamics algorithm is discussed. The molecular dynamics algorithms start by defining the initial configuration(the initial positions and velocities of the particles), and then the state of the system can be estimated and the new positions and velocities then computed using Newton's equation of motion. The force is recalculated using the new position and velocity of the particles.

The numerical methods that are used in the simulations are derived and the model potential (the Lennard-Jones potential) is described in this chapter. The net intermolecular force between two atoms obtained from the Lennard-Jones potential may be attractive or repulsive, depending on the distance between the atoms. When the distance between the atoms is very small, the force will be repulsive and the atoms will repel each other. When the atoms are far from each other the force become attractive and the atoms will attract each other.

CHAPTER FOUR

RESULTS AND DISCUSSION

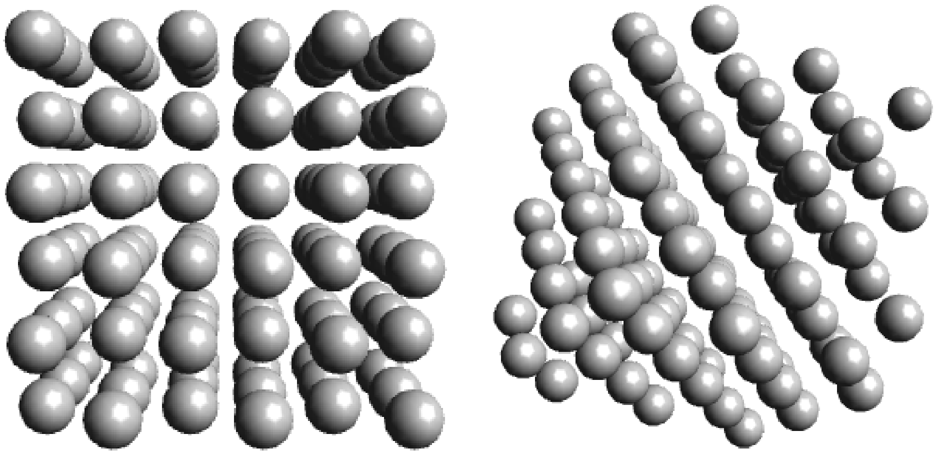
Introduction

In this chapter, we present molecular dynamics simulation on a microcanonical ensemble (NVE) for 27, 64, 125, 216 and 343 atoms using the Velocity Verlet Algorithm discussed in Chapter Three. The energy, volume, and number of particles are constant for a particular simulation, and it is assumed that time averages of properties of the simulated particles are equal to microcanonical ensemble averages of the same properties. We analyse the results obtained from the simulations and then calculate and discuss the dynamical properties from the simulations. All the quantities of the simulation results are dimensionless units (reduced units).

We also compare the computational speed between Euler's method and Velocity Verlet algorithm. We do this by running each algorithm for some systems and finding the average time it takes for each algorithm and the results is compared by plotting it. Finally in this chapter, we present an application of molecular dynamics simulations to protein folding using AMBER package.

Configuration

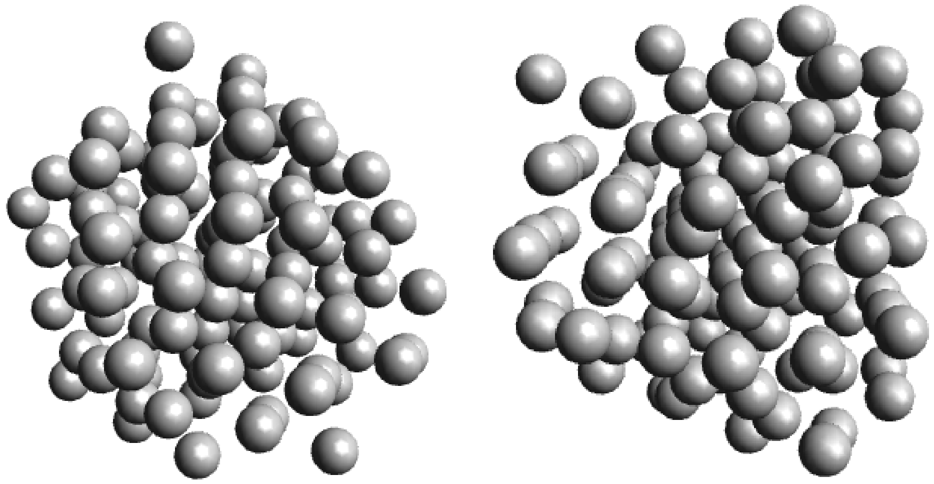
The geometry optimization for atoms are carried out using an iterative process, where atomic coordinates are adjusted until the total energy of individual structure reaches the minimum energy, i.e., when the potential energy reaches a local minimum. The configuration is done using face centered cubic (fcc) structure. The atoms are visualised using AVOGADRO. Figure 3 shows the arrangement of atoms before simulation and Figure 4 shows after the simulation (when total energy of individual structure reaches the minimum energy). The arrangement of atoms at the initial simulation are well arranged and looks nice as shown in Figure 3 but its not at optimum positions. The arrangement in Figure 4 shows the final arrangement of the atoms at equilibrium (optimum) positions, i.e., when the potential energy is at its minimum.



(a) Front View

(b) Side View

Figure 3: The Arrangement of Atoms at Initial Positions (Before the Start of the Simulation)



(a) Front View

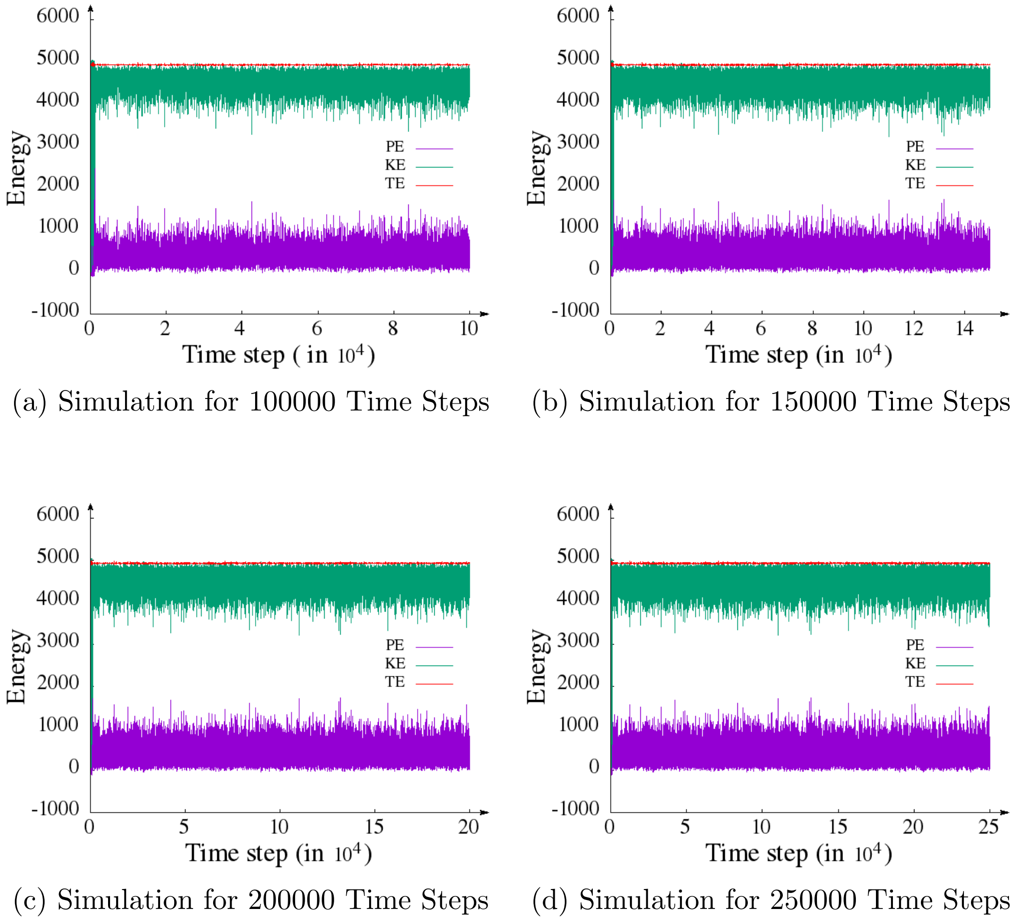
(b) Side View

Figure 4: The Arrangement of Atoms at Final Positions (After the End of the Simulation)

Energy Plots

The Total Energy (TE), Potential Energy (PE) and Kinetic Energy (KE) is plotted against time steps for a simulation of 27, 64, 125 216 and 343 atoms. The equilibrium plots of the energies over time obtained from molecular dynamics simulation for the various simulations are shown in Figures 5 to 9.

For the simulation of 27 atoms, the plots of total energy, potential energy and kinetic energy against time are shown in Figure 5. The energies fluctuates for the first 1500 time steps and then stabilises thereafter. The stability of energies demonstrates that the system relaxed and reached equilibrium. The total energy which is the sum of kinetic and potential energy remain constant throughout after the energy have been stabilised which shows conservation of energy. The potential energy of the system is positive and hence the total energy is above the kinetic energy.



(a) Simulation for 100000 Time Steps

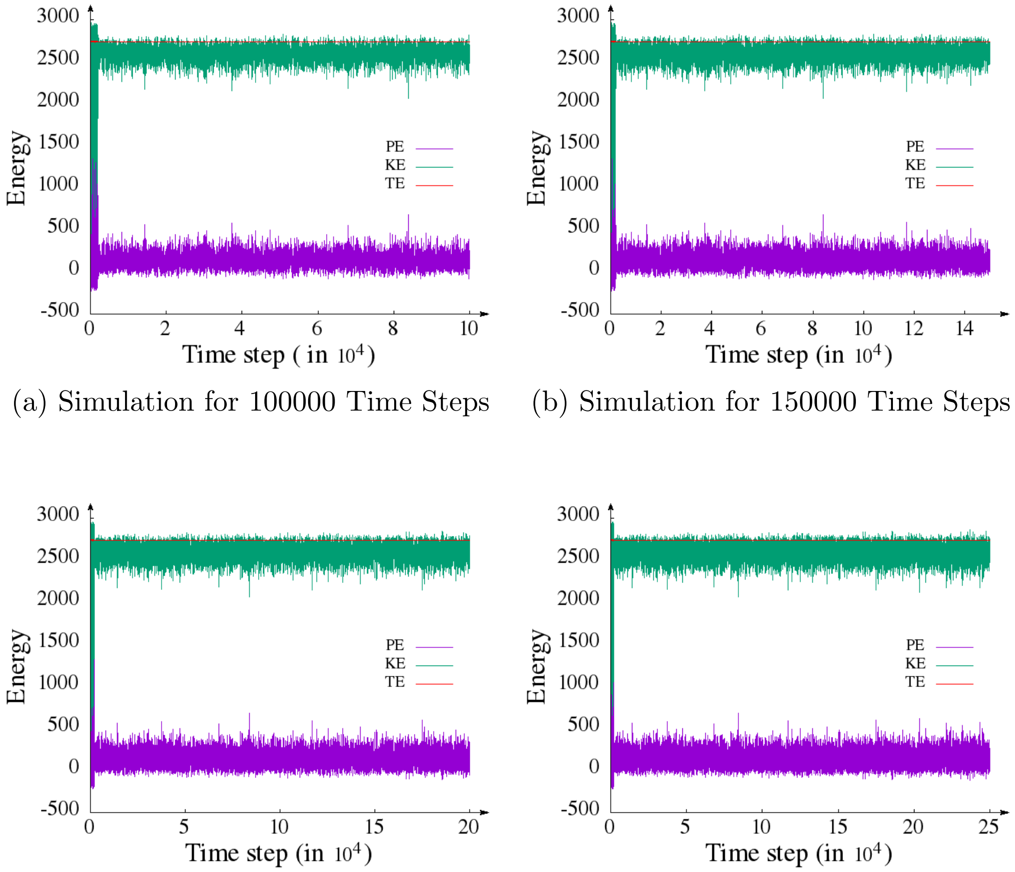
(b) Simulation for 150000 Time Steps

(c) Simulation for 200000 Time Steps

(d) Simulation for 250000 Time Steps

Figure 5: Equilibrium Curves of Total Energy (TE), Potential Energy (PE) and Kinetic Energy (KE) Obtained from Molecular Dynamics Simulation of 27 Atoms at Temperature = 1 at Different Time Steps

Figure 6 shows the plots of total energy, potential energy and kinetic energy as a function of time for the simulation of 64 atoms. The energies are unstable for the first 3000 time steps and then reaches an equilibrium energy after 3000 time steps. The total energy shows conservation throughout the simulation time. The potential energy of the system is also positive and hence the total energy is above kinetic energy as in the case of 27 atoms.



(a) Simulation for 100000 Time Steps

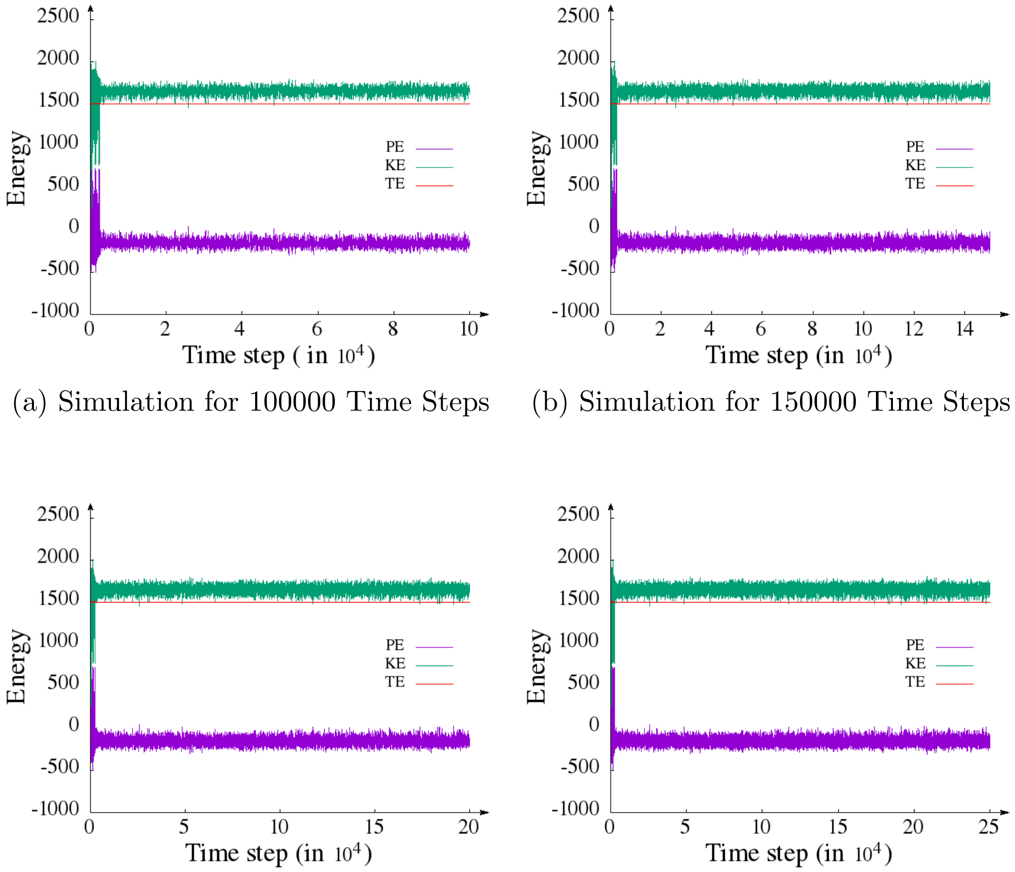
(b) Simulation for 150000 Time Steps

(c) Simulation for 200000 Time Steps

(d) Simulation for 250000 Time Steps

Figure 6: Equilibrium Curves of Total Energy (TE), Potential Energy (PE) and Kinetic Energy (KE) Obtained from Molecular Dynamics Simulation of 64 Atoms at Temperature = 1 at Different Time Steps

The plots of total energy, potential energy and kinetic energy as a function of time for the simulation of 125 atoms is shown Figure 7. The energies fluctuates for the first 3500 time steps and then reaches an equilibrium energy after 3500 time steps. The total energy is conserved when energy reaches equilibrium. The potential energy is negative and total energy is slightly below kinetic energy which shows a better energy as compared to the case of 27 and 64 atoms.



(a) Simulation for 100000 Time Steps

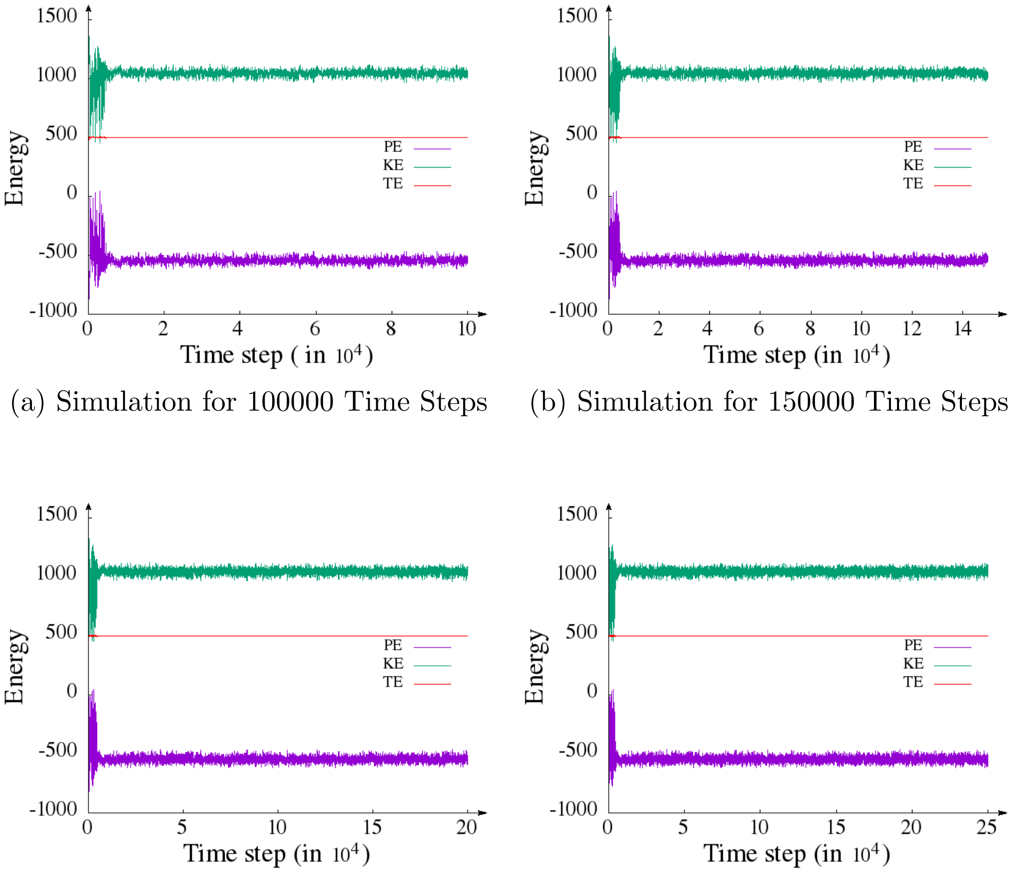
(b) Simulation for 150000 Time Steps

(c) Simulation for 200000 Time Steps

(d) Simulation for 250000 Time Steps

Figure 7: Equilibrium Curves of Total Energy (TE), Potential Energy (PE) and Kinetic Energy (KE) Obtained from Molecular Dynamics Simulation of 125 Atoms at Temperature = 1 at Different Time Steps

For the simulation of 216 the energies fluctuates for the first 5000 time steps and then stabilises (reaches an equilibrium energy) after 5000 time steps as shown in Figure 8. The total energy is conserved. The total energy is between potential energy and the kinetic energy and this shows a good simulation results.



(a) Simulation for 100000 Time Steps

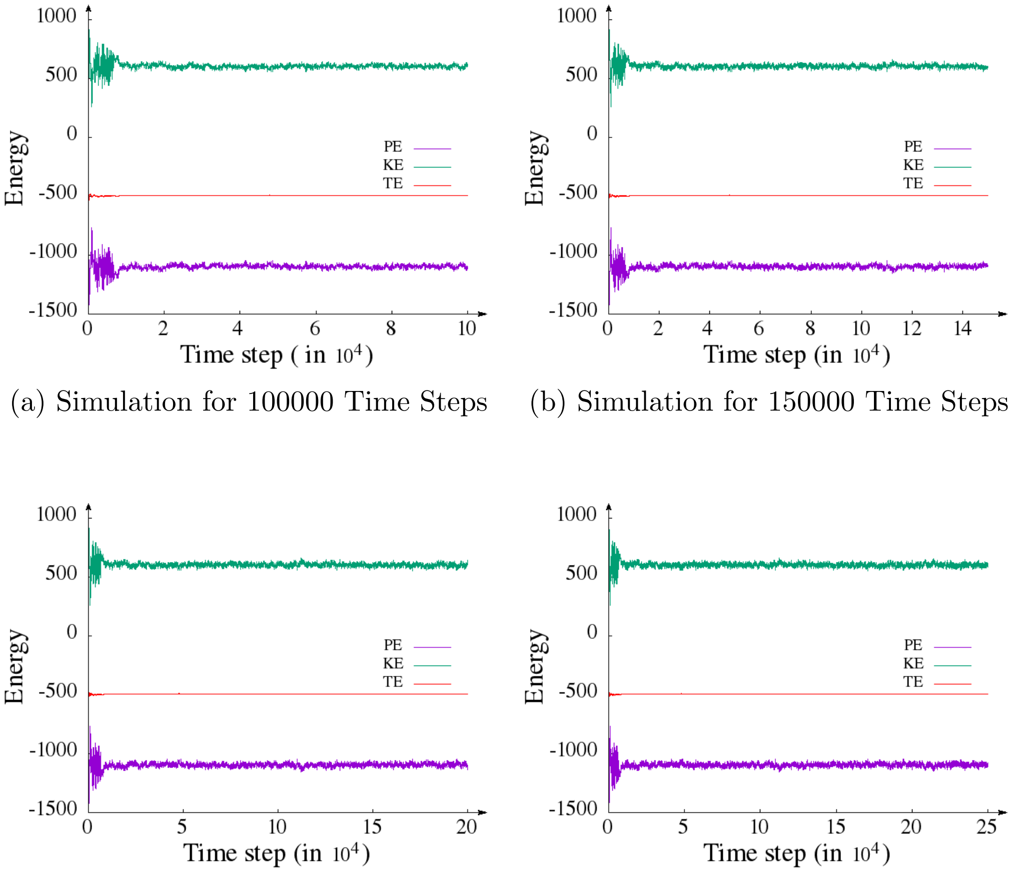
(b) Simulation for 150000 Time Steps

(c) Simulation for 200000 Time Steps

(d) Simulation for 250000 Time Steps

Figure 8: Equilibrium Curves of Total Energy (TE), Potential Energy (PE) and Kinetic Energy (KE) Obtained from Molecular Dynamics Simulation of 216 Atoms at Temperature = 1 at Different Time Steps

There is fluctuation of the energies for the first 8000 time steps when 343 atoms are simulated. It then stabilises (reaches an equilibrium energy) after 8000 time steps as shown in Figure 9. The total energy is conserved throughout the stabilised energy. The potential energy is negative and total energy is between kinetic energy and the potential energy which indicates a good results.



(a) Simulation for 100000 Time Steps

(b) Simulation for 150000 Time Steps

(c) Simulation for 200000 Time Steps

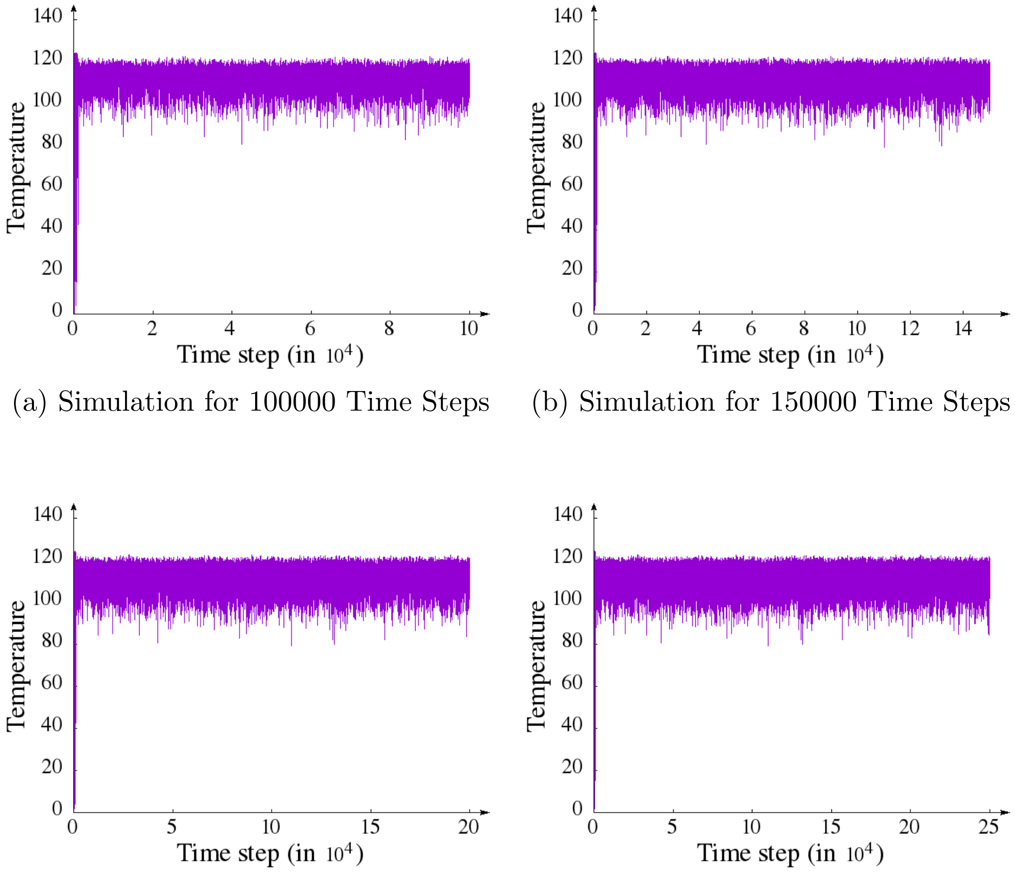
(d) Simulation for 250000 Time Steps

Figure 9: Equilibrium Curves of Total Energy (TE), Potential Energy (PE) and Kinetic Energy (KE) Obtained from Molecular Dynamics Simulation of 343 Atoms at Temperature = 1 at Different Time Steps

Temperature Plots

The Temperature obtained from the simulation results is plotted against time steps for a simulation of 27, 64, 125 216 and 343 atoms. The equilibrium plots of temperature is shown in Figures 10 to 14.

Figure 10 shows the temperature equilibrium curves obtained from molecular dynamics simulation of 27 atoms for different simulation time steps. It can be seen from the curves that temperature starts to fluctuate initially and then becomes steady. The system is equilibrated at a temperature of approximately 115 after 1500 time steps.

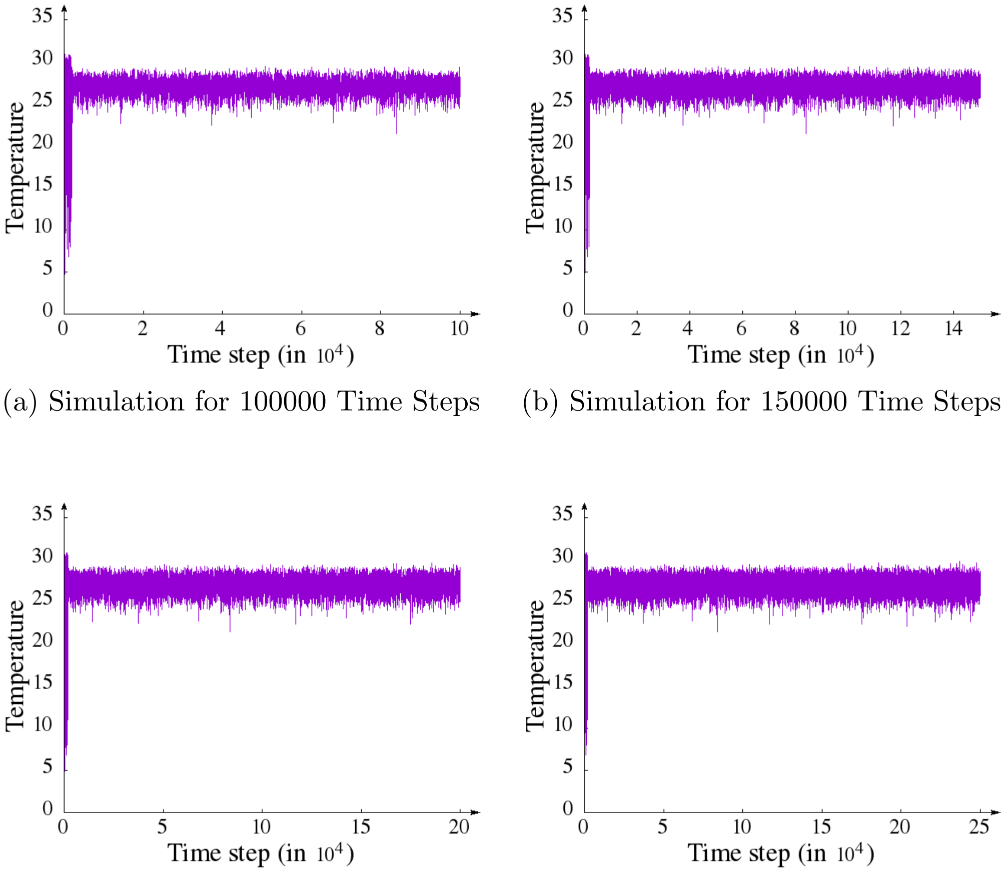


(a) Simulation for 100000 Time Steps (b) Simulation for 150000 Time Steps

(c) Simulation for 200000 Time Steps (d) Simulation for 250000 Time Steps

Figure 10: Temperature Equilibrium Curve Obtained from Molecular Dynamics Simulation of 27 Atoms at Initial Temperature = 1 at Different Time Steps

The temperature equilibrium curves obtained from molecular dynamics simulation of 64 atoms for different simulation time steps are shown in Figure 11. The temperature starts to fluctuate initially and then becomes equilibrated at a temperature of approximately 27 after about 3000 time steps.



(a) Simulation for 100000 Time Steps

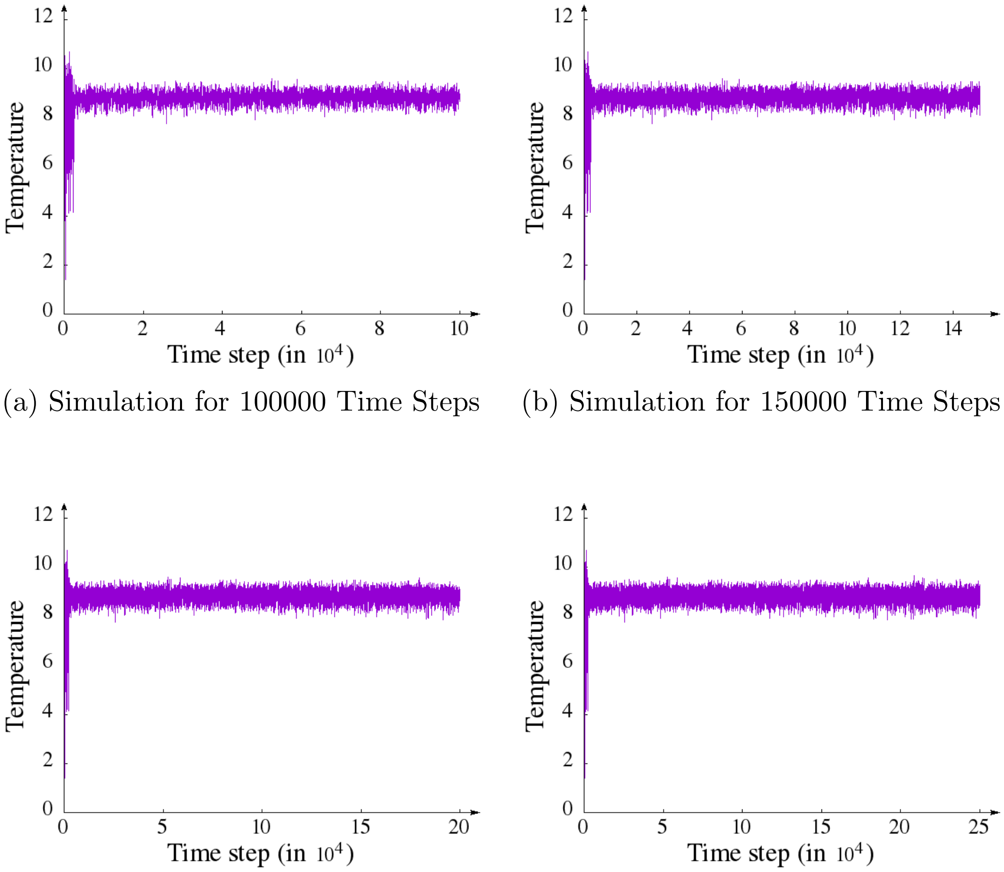
(b) Simulation for 150000 Time Steps

(c) Simulation for 200000 Time Steps

(d) Simulation for 250000 Time Steps

Figure 11: Temperature Equilibrium Curve Obtained from Molecular Dynamics Simulation of 64 Atoms at Initial Temperature = 1 at Different Time Steps

The plots in Figure 12 shows equilibrium behaviour of temperature obtained from molecular dynamics simulation of 125 atoms for different simulation time steps. The temperature is unstable for the first 3500 time steps and then stabilises thereafter at a temperature of approximately 8.8.

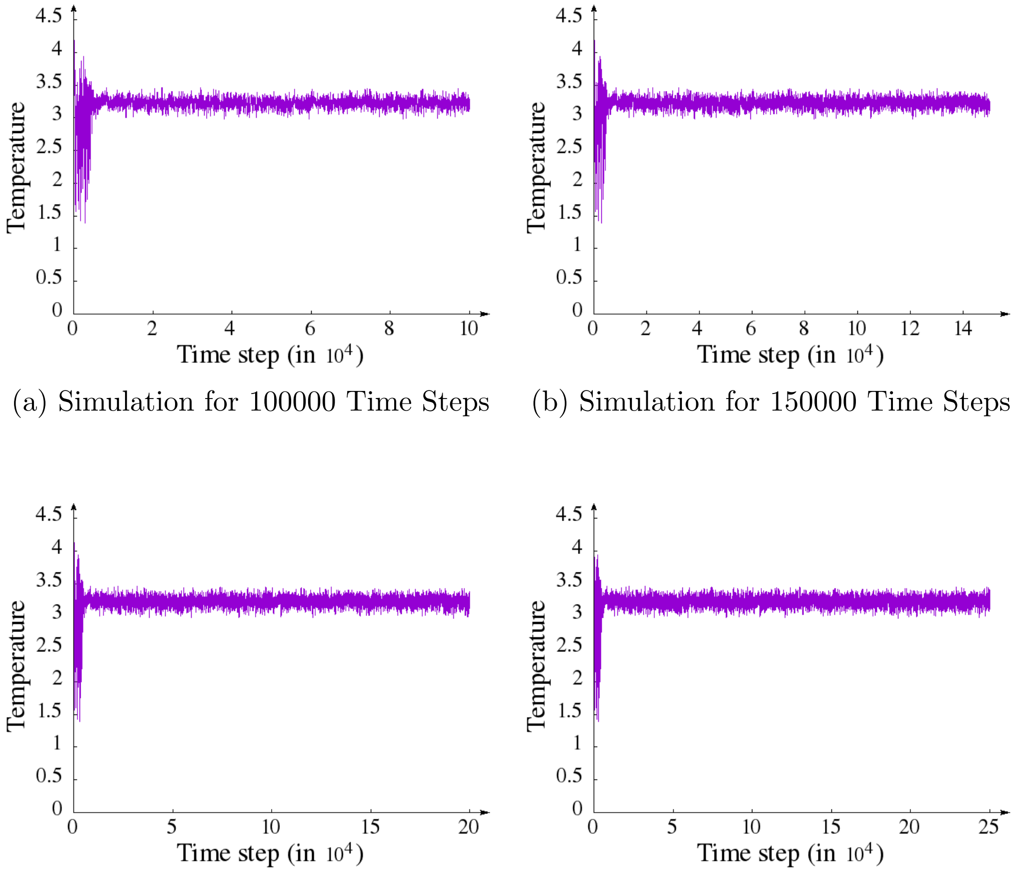


(a) Simulation for 100000 Time Steps (b) Simulation for 150000 Time Steps

(c) Simulation for 200000 Time Steps (d) Simulation for 250000 Time Steps

Figure 12: Temperature Equilibrium Curve Obtained from Molecular Dynamics Simulation of 125 Atoms at Initial Temperature = 1 at Different Time Steps

Figure 13 shows the temperature equilibrium curves obtained from molecular dynamics simulation of 216 atoms for different simulation time steps.The temperature fluctuates initially and equilibrated after about 5000 time steps at a temperature of approximately 3.2.



(a) Simulation for 100000 Time Steps

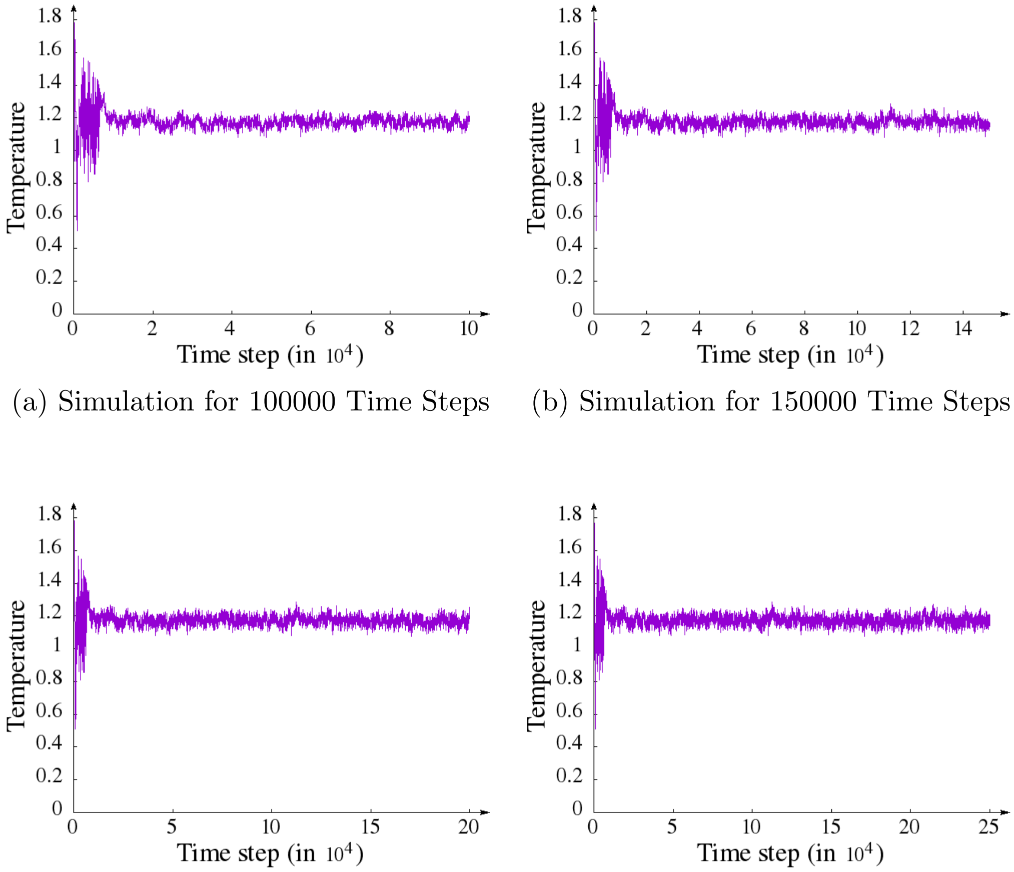
(b) Simulation for 150000 Time Steps

(c) Simulation for 200000 Time Steps

(d) Simulation for 250000 Time Steps

Figure 13: Temperature Equilibrium Curve Obtained from Molecular Dynamics Simulation of 216 Atoms at Initial Temperature = 1 at Different Time Steps

The plots of temperature obtained from molecular dynamics simulation of 343 atoms against time for different simulation time steps are shown in Figure 14. The temperature starts to fluctuate initially and then becomes equilibrated at an approximated temperature of 1.1 after about 8000 time steps.



(a) Simulation for 100000 Time Steps

(b) Simulation for 150000 Time Steps

(c) Simulation for 200000 Time Steps

(d) Simulation for 250000 Time Steps

Figure 14: Temperature Equilibrium Curve Obtained from Molecular Dynamics Simulation of 343 Atoms at Initial Temperature = 1 at Different Time Steps

Average Energies

The Tables 1 to 5 show the average potential energy, average kinetic energy and average total energy for 27, 64, 125, 216 and 343 atoms respectively at temperature = 1. The averages for each system is calculated after it has been stabilised. These averages are used to further show the relationships between some of the dynamic properties.

Table 1: The Average Energies of Simulation with 27 Atoms at Temperature = 1

Simulation Time	**Average PE**	**Average KE**	**Average TE**
100000	369.9402 ± 194.5350	4556.5370 ± 194.2768	4926.4772 ± 2.4749
150000	369.3416 ± 194.9532	4557.4819 ± 194.7028	4926.8235 ± 2.4444
200000	369.0096 ± 194.0697	4558.0151 ± 193.8210	4927.0248 ± 2.4057
250000	369.0494 ± 194.2222	4557.9968 ± 193.9752	4927.0461 ± 2.3564

Table 2: The Average Energies of Simulation with 64 Atoms at Temperature = 1

Simulation Time	**Average PE**	**Average KE**	**Average TE**
100000	134.1180 ± 83.2068	2604.0703 ± 83.1794	2738.1883 ± 0.2898
150000	132.8792 ± 82.8992	2605.2805 ± 82.8725	2738.1597 ± 0.2892
200000	132.6000 ± 83.2818	2605.5426 ± 83.2537	2738.1426 ± 0.2895
250000	133.2232 ± 83.8973	2604.8839 ± 83.8687	2738.1071 ± 0.2986

Table 3: The Average Energies of Simulation with 125 Atoms at Temperature = 1

Simulation Time	**Average PE**	**Average KE**	**Average TE**
100000	-154.4433 ± 42.3510	1656.4203 ± 42.3296	1501.9770 ± 0.2931
150000	-154.6579 ± 42.4859	1656.6536 ± 42.4697	1501.9957 ± 0.2871
200000	-154.3018 ± 42.4386	1656.3171 ± 42.4183	1502.0154 ± 0.2892
250000	-154.2962 ± 42.7693	1656.3204 ± 42.7495	1502.0242 ± 0.2869

Table 4: The Average Energies of Simulation with 216 Atoms at Temperature = 1

Simulation Time	**Average PE**	**Average KE**	**Average TE**
100000	-545.7625 ± 22.6347	1046.9626 ± 22.5990	501.2001 ± 0.3756
150000	-546.0245 ± 22.2915	1047.2333 ± 22.2581	501.2088 ± 0.3713
200000	-545.7453 ± 22.3818	1046.9647 ± 22.3443	501.2194 ± 0.3714
250000	-545.7389 ± 22.4430	1046.9742 ± 22.4053	501.2353 ± 0.3720

Table 5: The Average Energies of Simulation with 343 Atoms at Temperature = 1

Simulation Time	**Average PE**	**Average KE**	**Average TE**
100000	-1097.1947 ± 14.6684	604.6603 ± 14.3351	-492.5371 ± 0.7262
150000	-1097.3319 ± 14.6644	604.7831 ± 14.3124	-492.5488 ± 0.7374
200000	-1096.4304 ± 14.5143	603.9464 ± 14.1838	-492.4840 ± 0.7094
250000	-1096.4777 ± 14.5419	603.9876 ± 14.2092	-492.4900 ± 0.7118

The Relationships Between Some of the Dynamic Properties

Figure 15 shows the regression plot of number of atoms against average potential energy. It can be seen that there is a negative relationship between the number of atoms and potential energy and hence, as the number of atoms increases, the average potential energy decreases. The slope of the best fit is −4.55146.

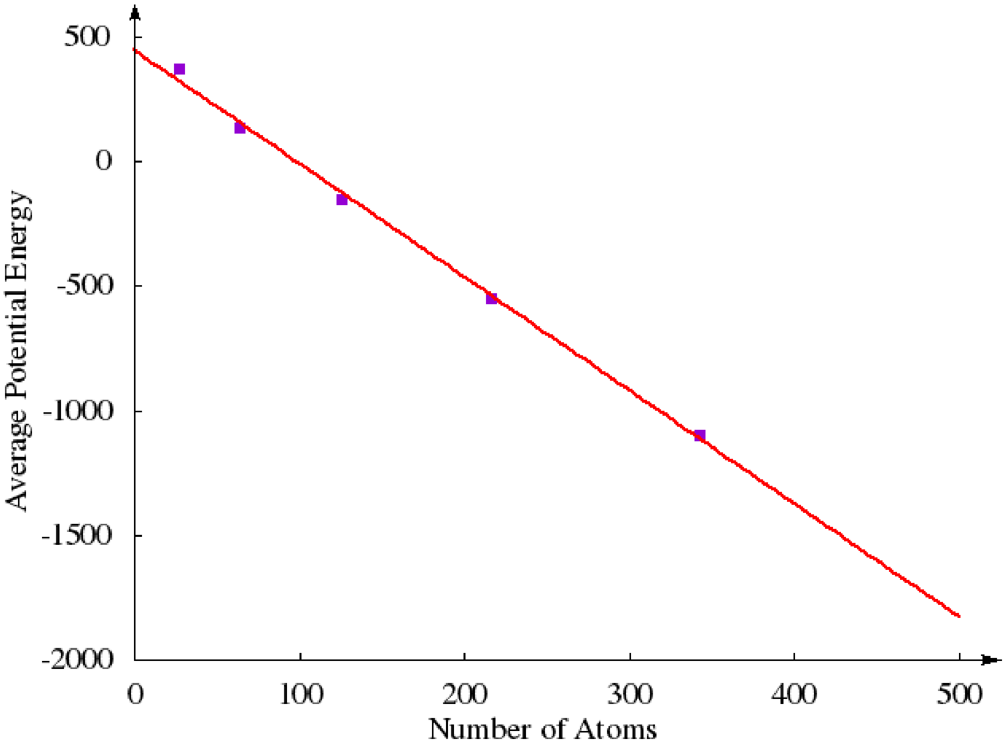


Figure 15: Regression Plot of Number of Atoms Against Potential Energy

The relationship between kinetic energy and temperature of a system is shown by the regression plots in Figures 16 to 20. It can be seen that temperature is positively related to the kinetic energy in all these simulations. This confirms the relationship between temperature and kinetic energy discussed earlier. The slope of the best fit for the simulation of 27, 64, 125, 216 and 343 are 0.0245, 0.0104, 0.0048, 0.0030 and 0.0019 respectively.

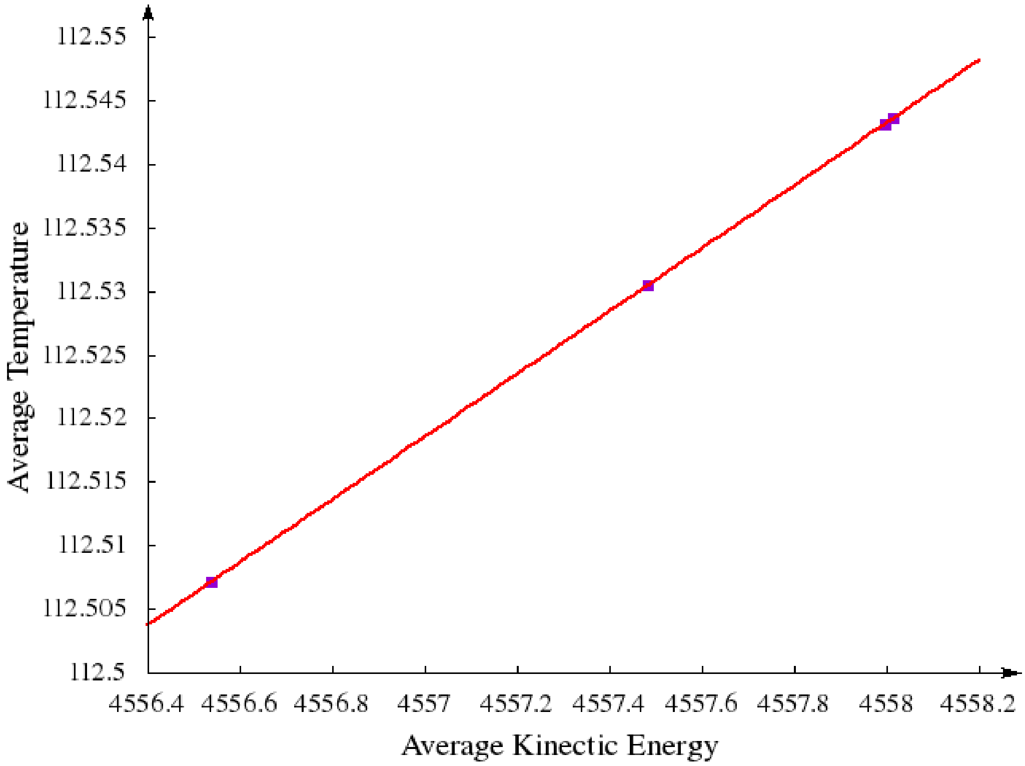


Figure 16: Regression Plot of Average Kinetic Energy against Average Temperature for Simulation of 27 Atoms

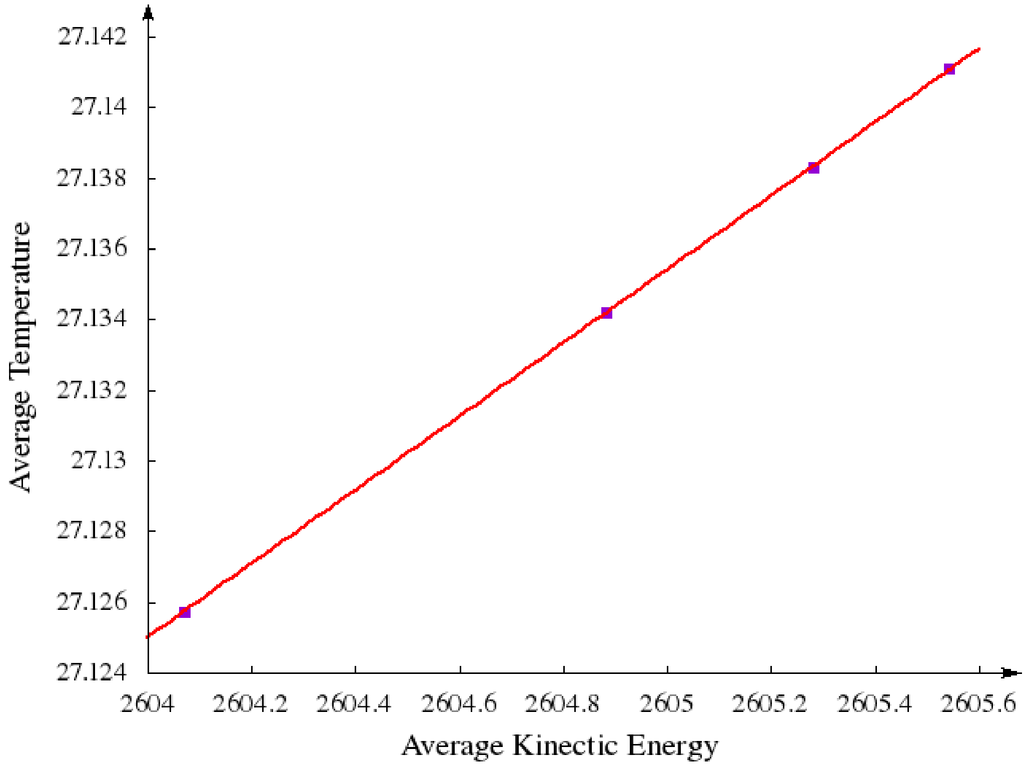


Figure 17: Regression Plot of Average Kinetic Energy Against Average Temperature for Simulation of 64 Atoms

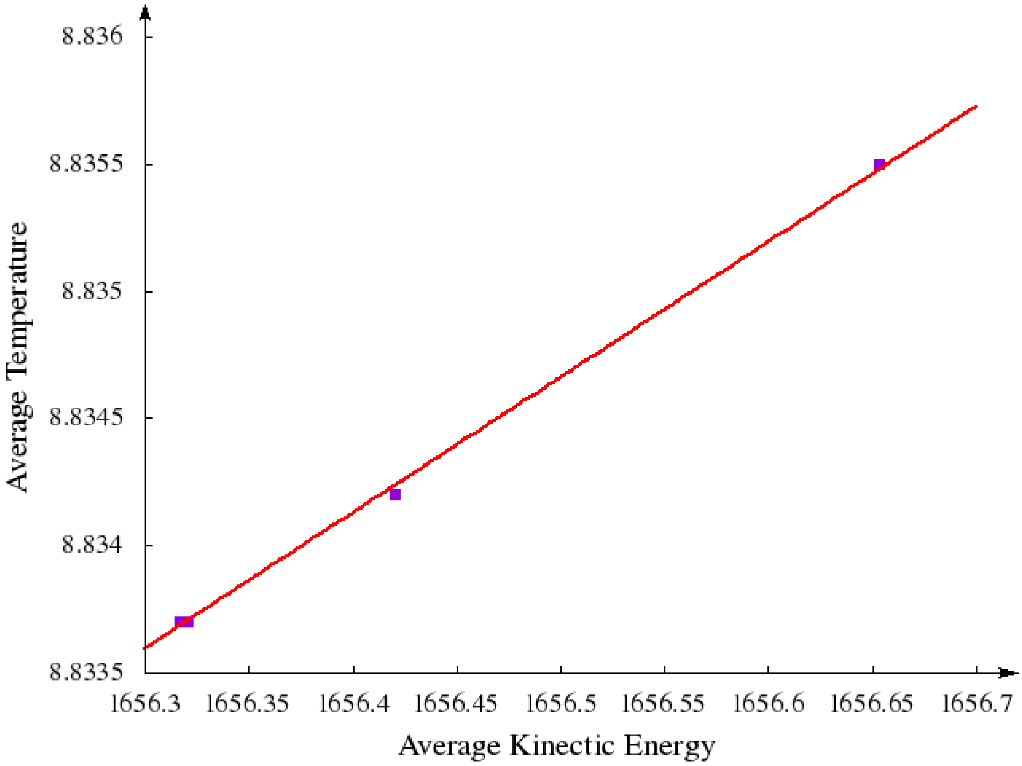


Figure 18: Regression Plot of Average Kinetic Energy Against Average Temperature for Simulation of 125 Atoms

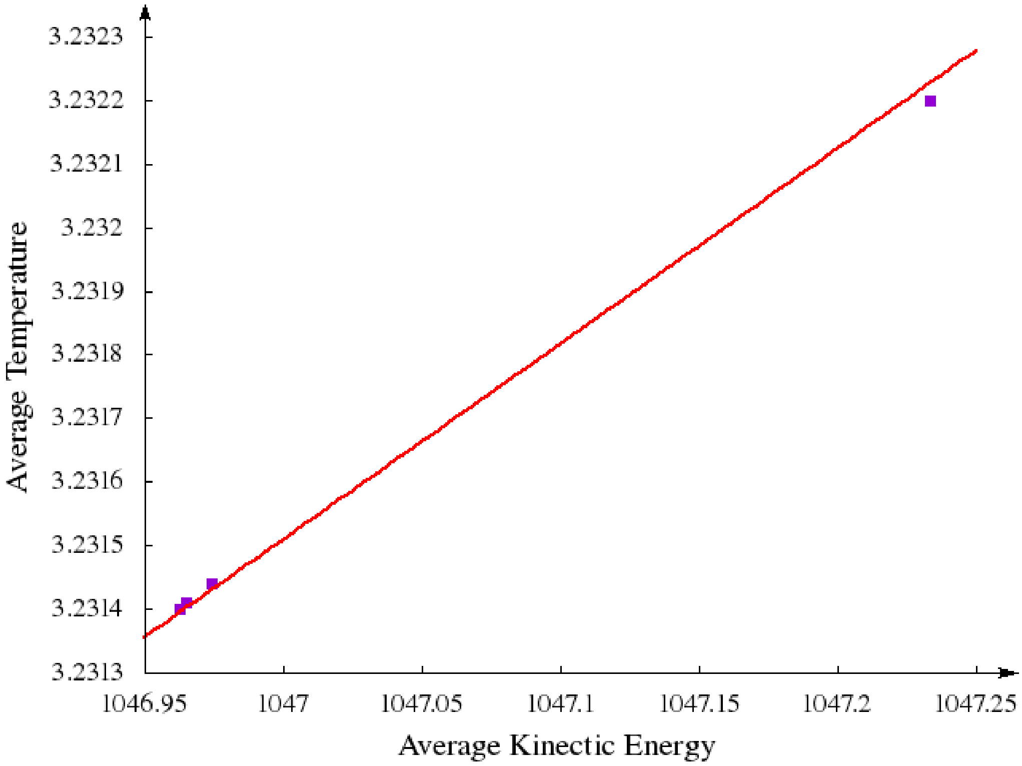


Figure 19: Regression Plot of Average Kinetic Energy Against Average Temperature for Simulation of 216 Atoms

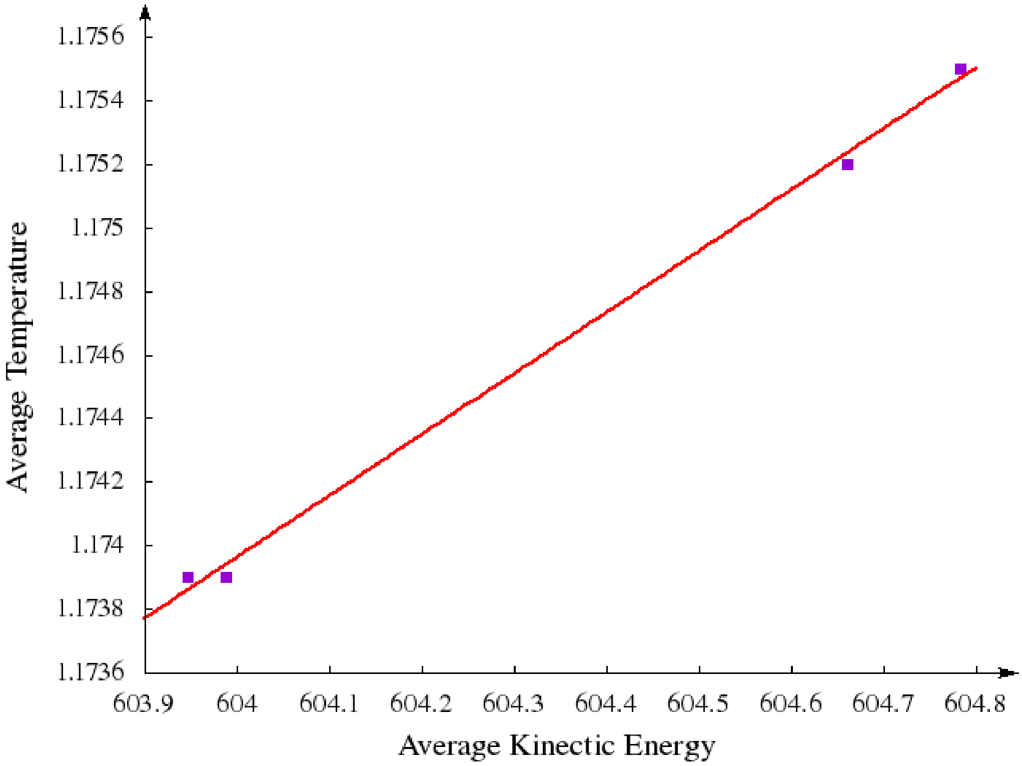


Figure 20: Regression Plot of Average Kinetic Energy Against Average Temperature for Simulation of 343 Atoms

Averege Potential Energy against Time Steps

Average potential energy is plotted against time steps in Figure 21 to 25. It can be seen in Figures 21 and 22 that the minimum average potential energy for 27 and 64 atoms are obtained at 200, 000 time step. It is also seen in Figures 23, 24 and 25 that the minimum average potential energy for 125, 216 and 343 atoms are obtained at 150, 000 time step.

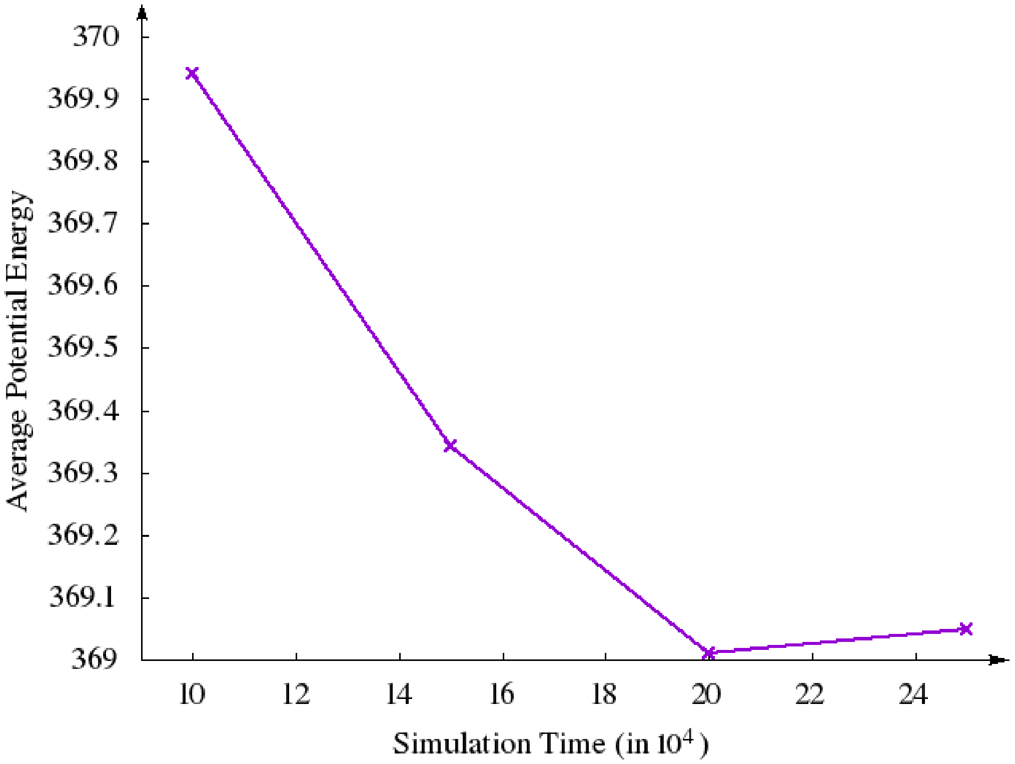


Figure 21: A Plot of Average Potential Energy Against Time Steps for Simulation of 27 Atoms

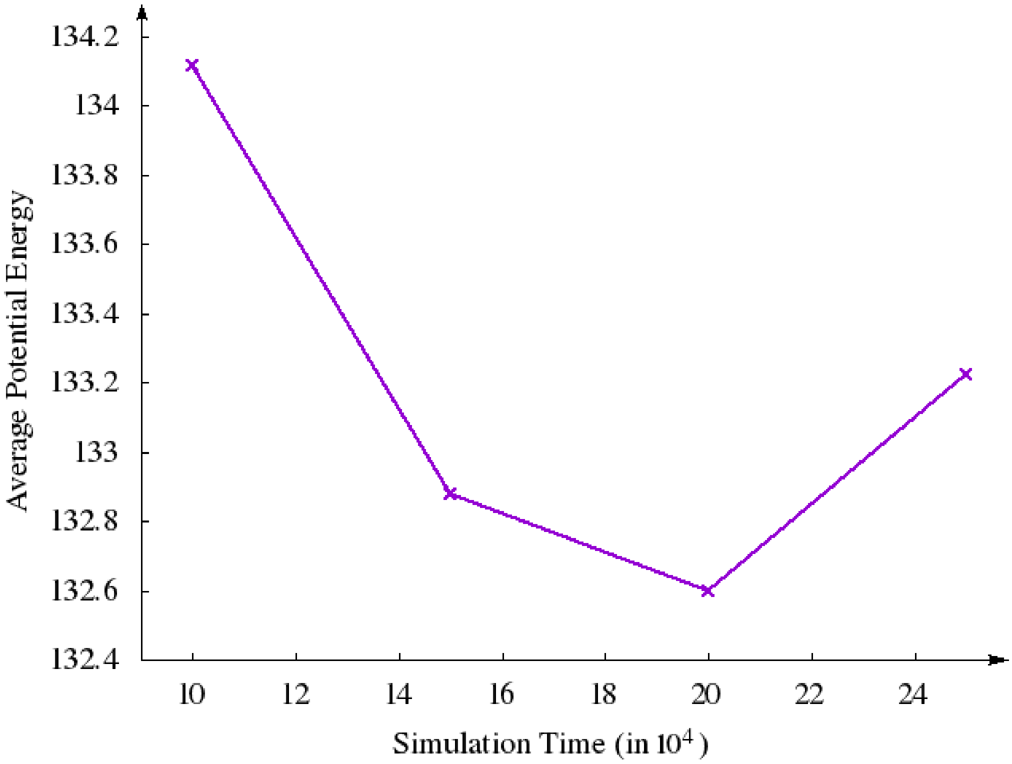


Figure 22: A Plot of Average Potential Energy Against Time Steps for Simulation of 64 Atoms

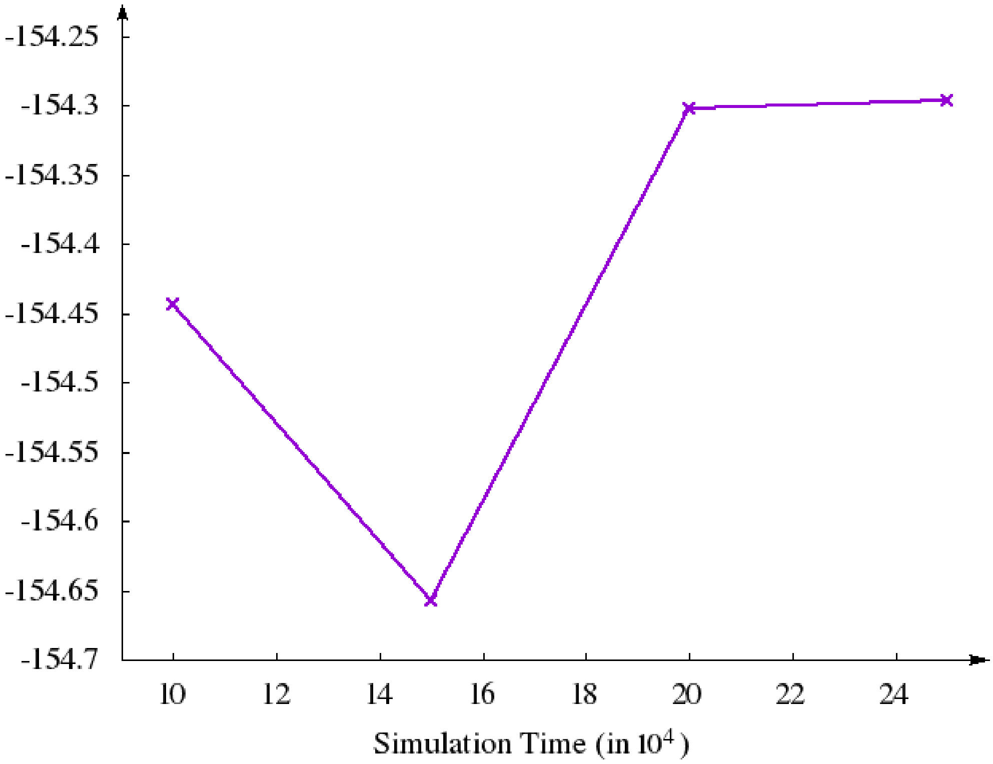


Figure 23: A Plot of Average Potential Energy Against Time Steps for Simulation of 125 Atoms

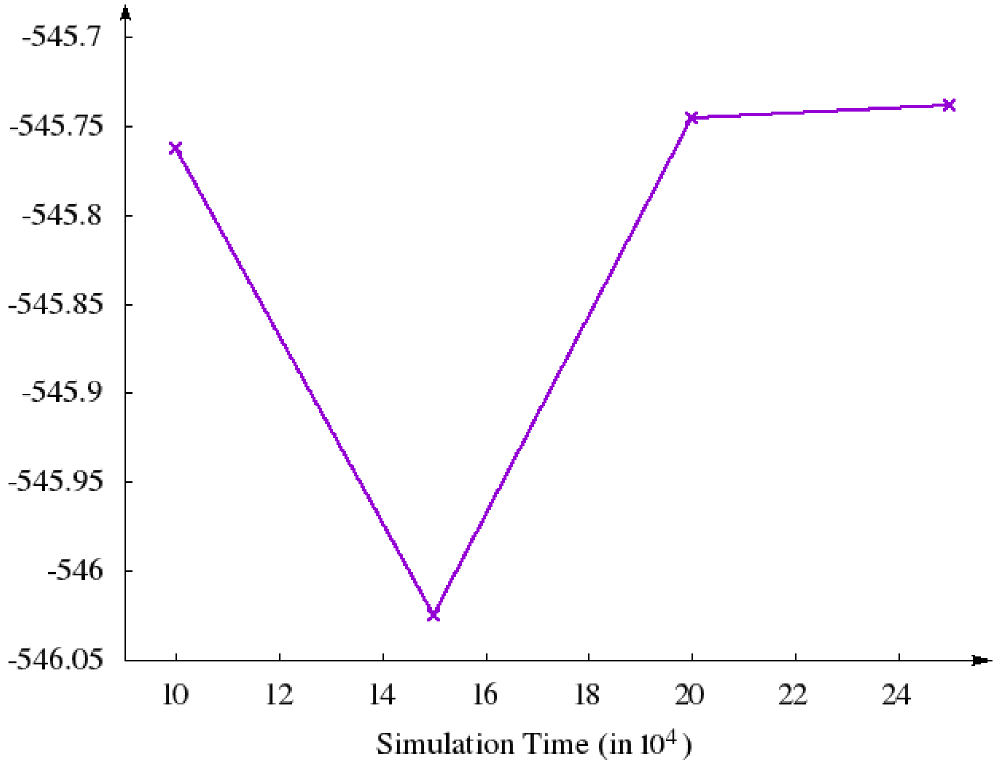


Figure 24: A Plot of Average Potential Energy Against Time Steps for Simulation of 216 Atoms

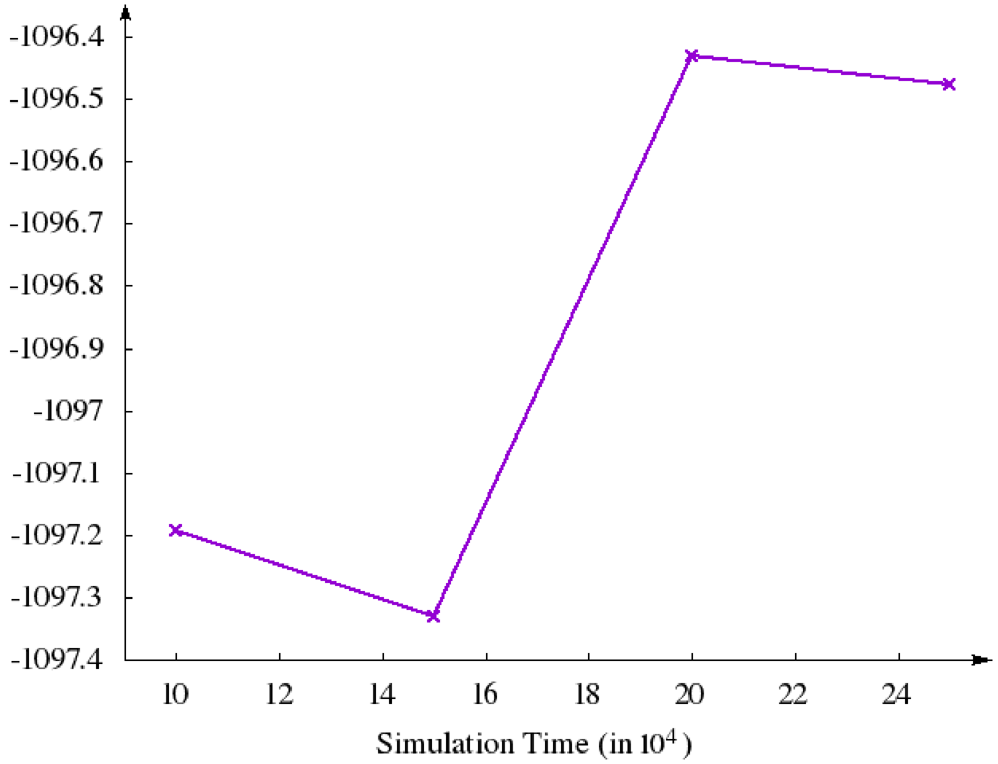


Figure 25: A Plot of Average Potential Energy Against Time Steps for Simulation of 343 Atoms

Computational Speed Between Euler Integration and Velocity Verlet Integration Algorithms

To compare the computational speed between Euler method and Velocity Verlet algorithm, we implemented both methods in MATLAB and run simulations of different systems for both algorithms. The plots in Figures 26 to 29 shows the plots from these simulations. It can be seen that most of the points for Euler's algorithm are above that of Velocity Verlet algorithm for all the plots except in Figure 28. This shows that Velocity Verlet is faster than the Euler's algorithm. It can also be seen that the computational time for Euler's algorithm is higher than that of Velocity Verlet for higher number of atoms except for Figure 28. This means Velocity Verlet is better than Euler algorithm for higher number of atoms.

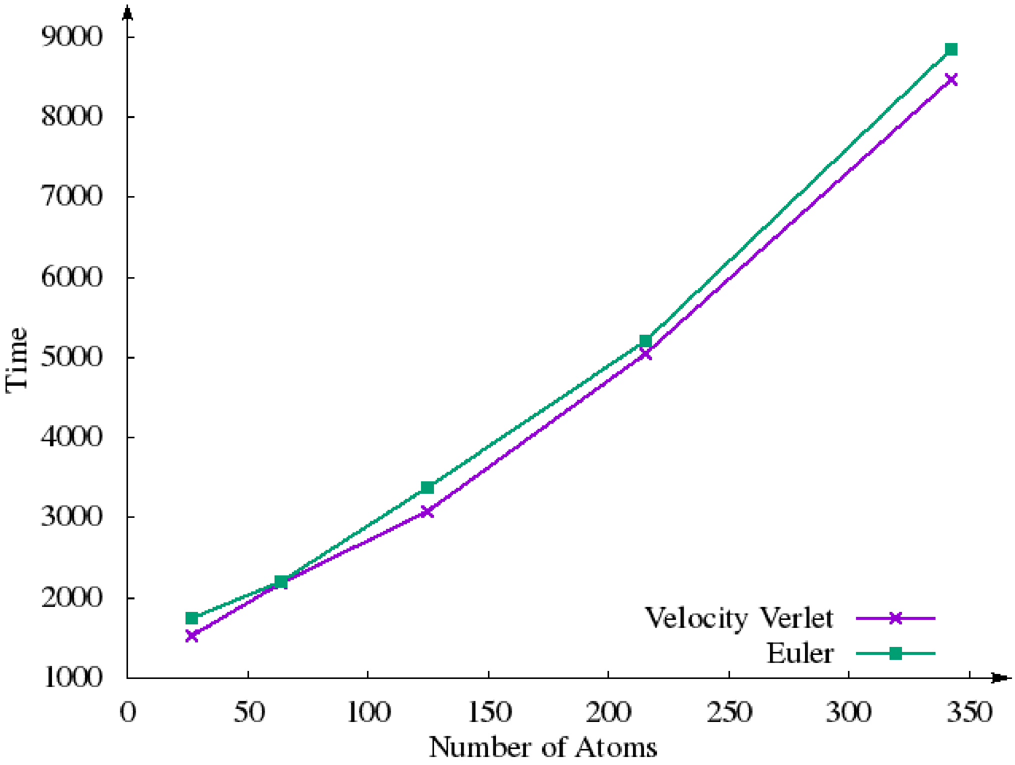


Figure 26: Plot Showing the Computational Times of Euler Integration and Velocity Verlet Integration Algorithms at Temperature = 0.1

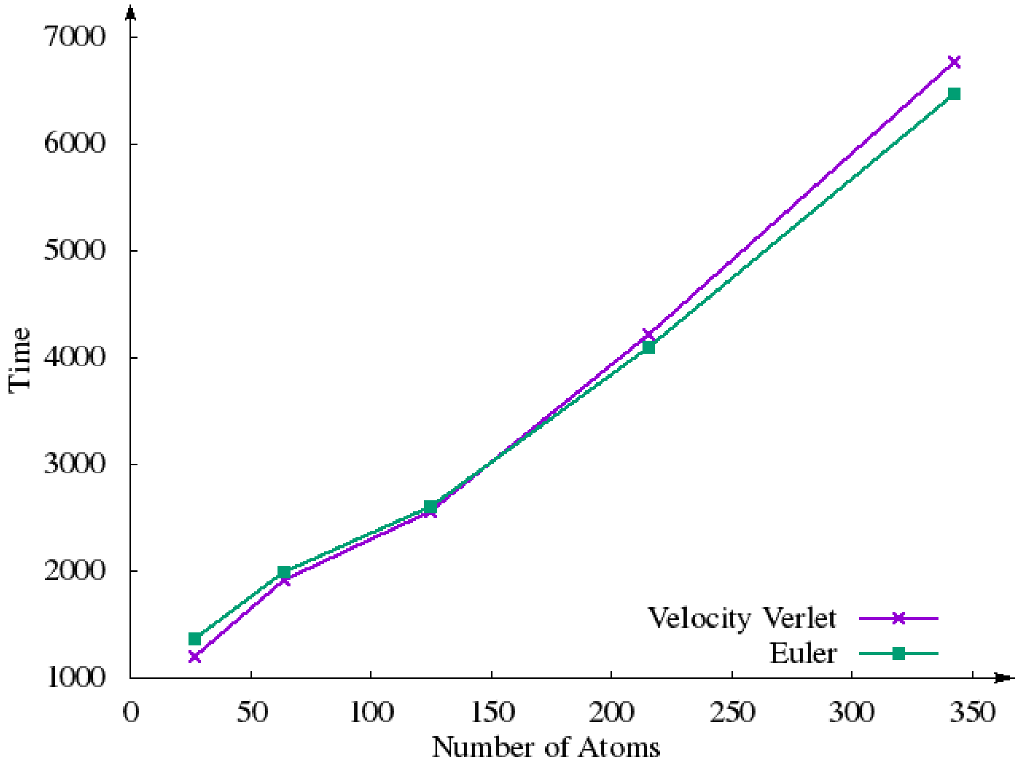


Figure 27: Plot Showing the Computational Times of Euler Integration and Velocity Verlet Integration Algorithms at Temperature = 0.3

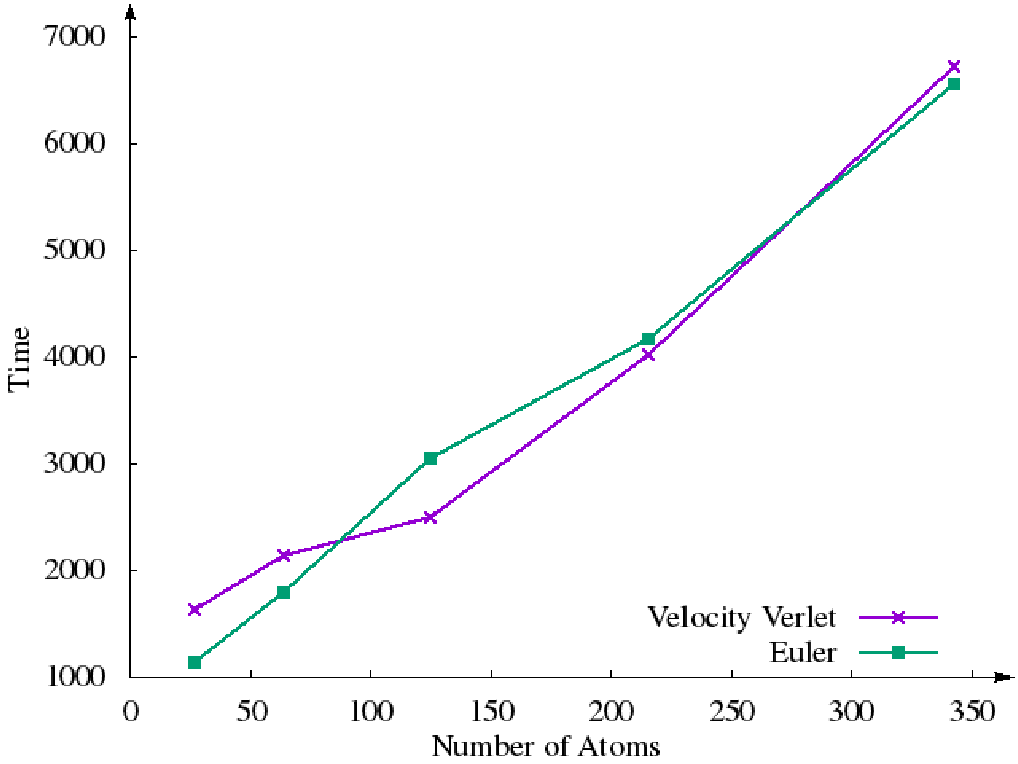


Figure 28: Plot Showing the Computational Times of Euler Integration and Velocity Verlet Integration Algorithms at Temperature = 1.0

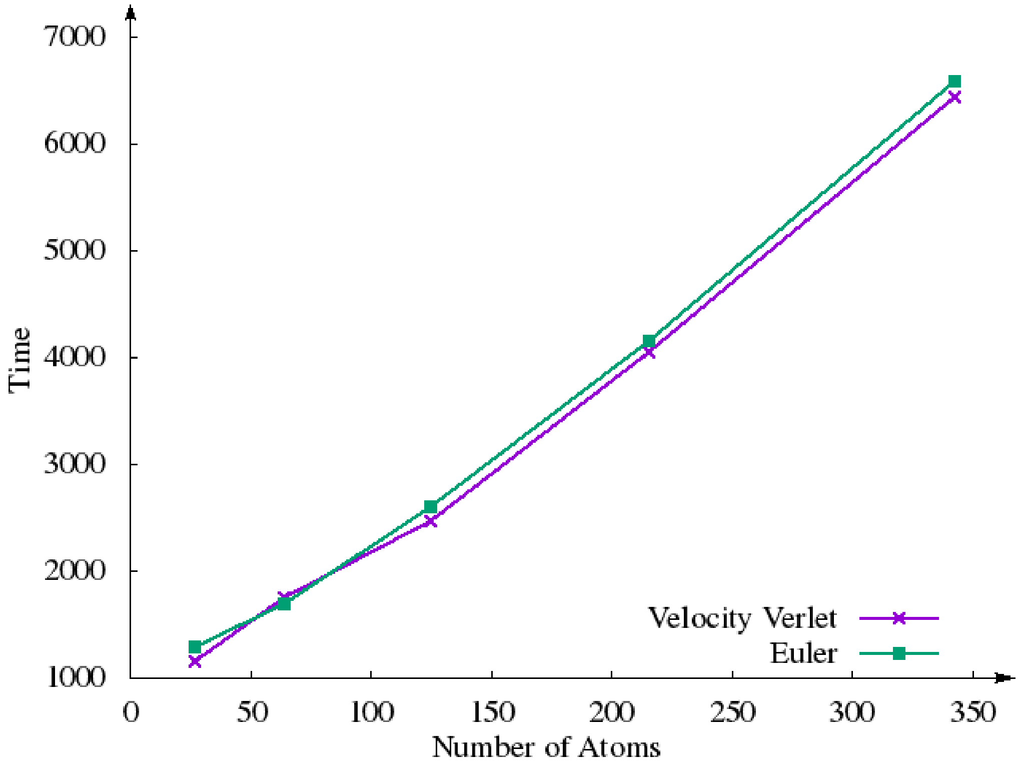


Figure 29: Plot Showing the Computational Times of Euler Integration and Velocity Verlet Integration Algorithms at Temperature = 1.5

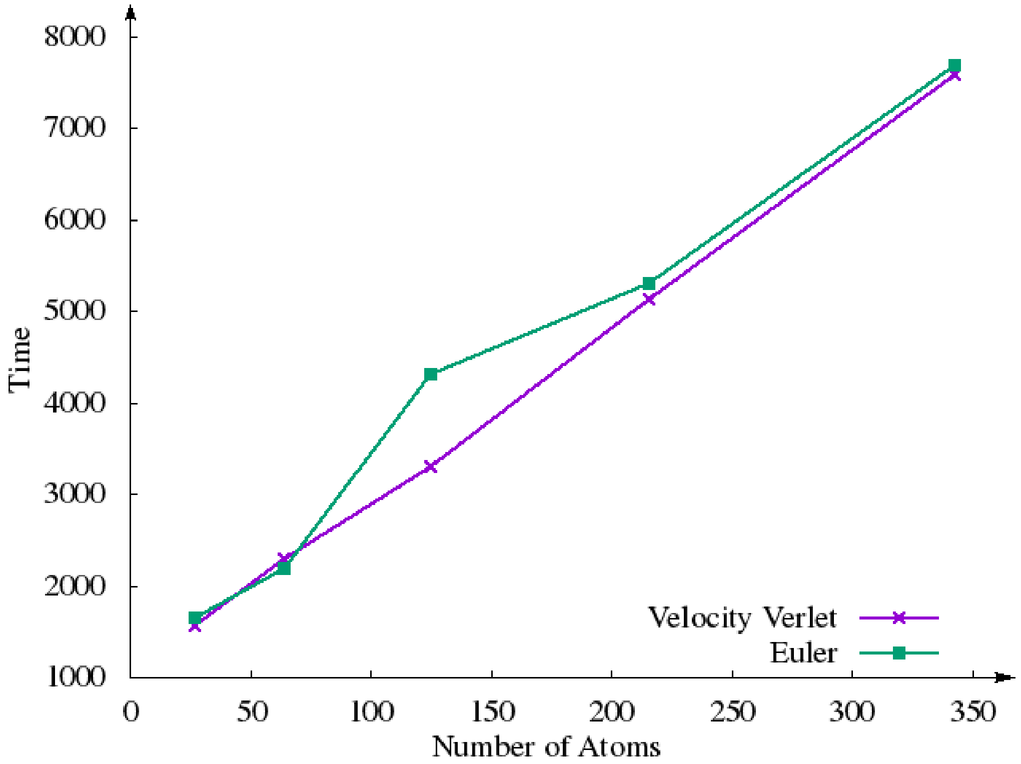


Figure 30: Plot Showing the Computational Times of Euler Integration and Velocity Verlet Integration Algorithms at Temperature = 2.0

Application of Molecular Dynamics Simulation to Protein Folding

This section present an application of molecular dynamics simulation using the AMBER package to study protein folding. Protein have to be folded into a three-dimensional shape to become biologically active to perform effectively and also reproduce. Protein folding study is important to know the functions and some dynamics of protein. It can for example help to understand the replication and control of viruses.

Amino Acids and Proteins

Amino acids are organic molecules that are made up of basic amine ($-NH_2$), acidic carboxyl ($-COOH$) functional groups and side chain (R group) that is unique to each amino acid. Carbon (C), Hydrogen (H), Oxygen (O) and Nitrogen (N) are the key elements of amino acid although there can be other elements found in the side chain of amino acid. When the carbon atom is a attached to the carbonyl, it is called α-carbon and where the amino group is attached to the secondary carbon instead of the α-carbon is known as the β-carbon. Each amino acid is unique based on the particular chemical structure of the R group. The R group is the part of the amino group other than the amino group, the acid group and the central carbon.

Amino acids consist of two main components, the backbone or main chain and a side chain. The backbone chains are made up of central chains of covalently bonded atoms which consist of the amino and carboxyl acid links to the alpha carbon of each amino acid. The side chain is a hydrocarbon branching element of a molecule that is joined to the backbone. The identity of the amino acid is identified by its side chain. When an amino acid chain is ended by a free carboxyl group it is called C-terminus or carboxyl-terminus and if the amino acid chain is ended by a free amine group it is called N-terminus or amine-terminus.

Amino acids are used by the body to break down food, grow, repair body tissue and serve as a source of energy to the body. They are been used in industries to produce feed, drugs, nutritional supplements, biodegradable plastics, fertilizers for agriculture, cosmetics etc.

Amino acids forms the building blocks for proteins. When they carboxyl group of one amino acid are linked to the amine group of another by a peptide bond, it forms polypeptide and protein. Peptides are short chains of amino acids join together by peptide bonds and a polypeptide is a linear peptide chain. Proteins are made up of one or more polypeptides. The basic difference between peptides and protein is a size and structure (peptides are smaller than proteins and usually contain approximately 50 or less amino acids).

Proteins are large biological molecules or macromolecules that are contained in all living organisms. They consist of hundreds or thousands of smaller units called amino acids linked together through covalent peptide bond to form long chains. There are about 20 different types of amino acids that occur naturally to make a protein. These are shown in the appendix.

Proteins differ from one another primary in the number and type of amino acids that are attached in forming a polypeptide chain. Each protein has a unique sequence of amino acids. The specific sequence of amino acids of a protein determines its folding into a specific $3-$dimensional structure and function. The natural folding of protein into its unique $3-$dimensional structure is called native conformation.

Protein can be classified based on the shape into fibrous proteins and globular proteins. Fibrous proteins which are also known as scleroproteins proteins are elongated and fibrous in nature and parallel to one another. Its role include providing structure, protection, support and forming connective tissue and muscle fibre. They are insoluble in water. Globular proteins are also known as spheroproteins and are formed in a spherical shape. They serve as catalyst to organic reaction, transporters of other particles through membrane and transmitting messages to regulate biological processes. Globular proteins dissolves in water.

Proteins perform important function in every aspect of cellular life. It function as a catalyst to chemical reactions, transmit signals to coordinate biological process between different cells, bind and transport atoms and smaller molecules within cells and throughout the body, build different structures and protect body from foreign pathogens (Encyclopaedia Britannica contributors, 2019; Wikipedia contributors, 2019b).

Protein Folding

Protein folding is a physical process by which a polypeptide chain is folded or coiled to a 3-dimensional structure. When protein is coiled or folded into a specific three-dimensional shape, it becomes biologically active to perform effectively and also reproduce. The function of a protein is determined by the protein structure. Thus, the structure of protein can predict the biological activities of living organisms. It can be applied to study the spread and prevention or control the replication of viruses.

There are four distinct structures of protein structures. They are the primary structure, the secondary structure, the tertiary structure and the quaternary structure. The primary structure of protein is the linear sequence of amino acids in each polypeptide chain that makes up the protein with no regard for the arrangement of the peptide chain in space. The secondary structure is the arrangement of the main peptide chain in space of adjacent amino acid residues with no regard for the conformation of side chains or other segments of the main chain. The α-helices and β sheets are the major secondary structural elements. The tertiary structure is made up of the side chains and other adjacent segments of the main chain, with no regard for other peptide chains. Every protein will have its specific three-dimensional shape at this level. The quaternary structure is used for the arrangement multiple subunits of a large protein in which each subunit is a separate peptide chain (Encyclopaedia Britannica contributors, 2019; Wikipedia contributors, 2019b).

As mentioned earlier the structure of a protein is determined by the specific sequence of amino acids. This means to understand the functions and some dynamics of protein, it is important to determine its structure. Simply put, we need to know the relationship between the sequence of amino acids and the native-structure to be able to understand the function of protein.

Since the functional properties of proteins depend on their $3-$dimensional structure, a number of methods have been developed to predict the folding behaviour of a proteins. Some experimental methods include X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, electron microscopy, circular dichroism, dual polarisation interferometry etc. These methods have their various strength and weaknesses. There are some of the systems that cannot be studied by the experimental methods mentioned above and hence require complex computational methods. Some of these methods include but not limited to Molecular dynamics and Monte Carlo simulations. In this research, we will used molecular dynamics simulation to study the folding dynamics of a protein.

The AMBER Model Potential

AMBER stands for Assisted Model Building with Energy Refinement. It is a force fields for molecular dynamics of biomolecules originally developed by Peter Kollman's group at the University of California, San Francisco. To use the AMBER force field, it is necessary to have values for the parameters of the force field (e.g. force constants, equilibrium bond lengths and angles, charges). A number of these parameter sets exist and each parameter set has a name, and provides parameters for certain types of molecules. They include General AMBER force field (GAFF), GLYCAM force fields, ff14SB, ff12SB, protein.ff14SB etc (Wikipedia contributors, 2019a). In this thesis, the Amber ff14SB force field is used for the protein folding application. The functional form of the AMBER force field uses the following Hamiltonian (Cornell et al., 1995)

$$\begin{aligned} V(r^N) &= \sum_{\text{bonds}} k_b(l-l_0)^2 \\ &+ \sum_{\text{angles}} k_\theta(\theta-\theta_0)^2 \\ &+ \sum_{\text{torsions}} \sum_{n} \frac{1}{2} V_n[1+\cos(n\omega-\gamma)] \\ &+ \sum_{j=1}^{N-1} \sum_{i=j+1}^{N} \left\{ \epsilon_{ij} \left[\left(\frac{A_{ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{B_{ij}}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right\}. \end{aligned} \tag{4.1}$$

The terms k_b, l_0, k_θ, θ_0, V_n, γ, A_{ij}, B_{ij} are parameters to be specified based on the various Amber force fields. The first term on the right of Equation 4.1 is the harmonic term for bond stretching. It represents the energy between covalently bonded atoms with force constant k_b, instantaneous bond lengths, l and l_0 is the value for the bond length at equilibrium that is considered a parameter. The term summing over angles represents the energy due to the bending of two contiguous bonds. Angle bending terms are parameterized by a force constant k_θ and the equilibrium angle value θ_0 in degrees. The third term, summing over torsions angles, models the energy for twisting three contiguous bonds with the force constant V_n, the multiplicity n, a phase shift γ, and the torsion angle ω. The term (double summation over i and j) represents the non-bonded energy between all atom pairs, which can be decomposed into van der Waals and electrostatic energies. The van der Waals potentials take into account repulsion between atoms at small separations accounting for the excluded volume between atoms and also weak attraction at larger distances. The common form of this potential for a pair of atoms i and j is given by a Lennard-Jones function:

$$V_{i,j} = 4\epsilon_{i,j}\left[\left(\frac{\sigma_{i,j}}{r_{i,j}}\right)^{12} - \left(\frac{\sigma_{i,j}}{r_{i,j}}\right)^{6}\right].$$

Here again, $r_{i,j}$ is the distance separating the two atoms, $\epsilon_{i,j}$ is the depth of the potential well for the interaction of atoms i and j, and $\sigma_{i,j}$ is the distance where the potential is exactly zero. The electrostatic or Coulomb potential describes the interactions between pairs of partial charges. q_i and q_j are the partial charges on the atoms i and j and ϵ_0 is the dielectric constant of vacuum.

Simulation Results of a Protein with 20 Amino Acid Sequence

The AMBER package was used for the simulation of small protein (peptide) of 20 amino acids sequence. The sequence of the amino acids used is NLYIQWLKDGGPSSGRPPPS (see appendix). The results of these simulations are shown in Figures 31 to 36 . Figure 31 shows the initial configuration of the simulation. The ends of the proteins in Figure 31 with blue colour is the $N-$terminal and the part with red colour is the $C-$terminal. After running the simulation for 500000 iterations, the result in Figure 32 was obtained. It can be seen that the protein does not really fold. The results for 3000000, 4000000 and 5000000 iterations are shown in Figures 32, 33 , 34 and 35 respectively. The protein folds and helix starts to form in Figure 33 and more of the helices formed in Figure 35. Figure 36 shows all the protein structures superimposed on each other to show how the changes are formed along the simulation. With this folded form, we can predict its biological functions. If there is a viral protein with this sequence of amino acids, its replication can be predicted since it have been folded into its $3D-$structure and hence control measures can be taken to stop or slow its replication.

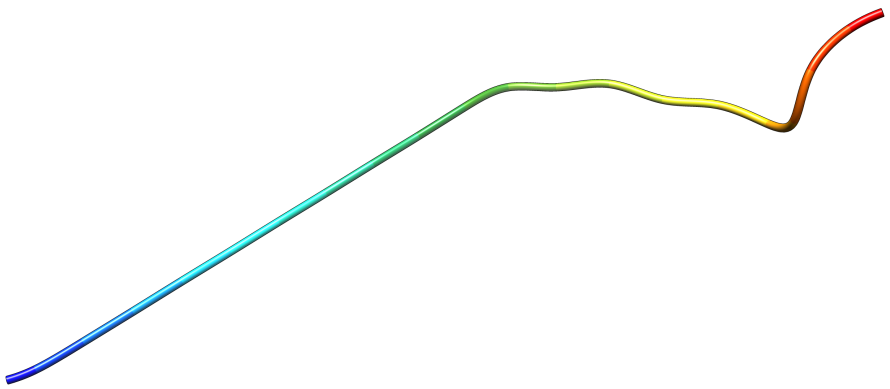


Figure 31: The Initial Configuration of 20 Amino Acid Protein

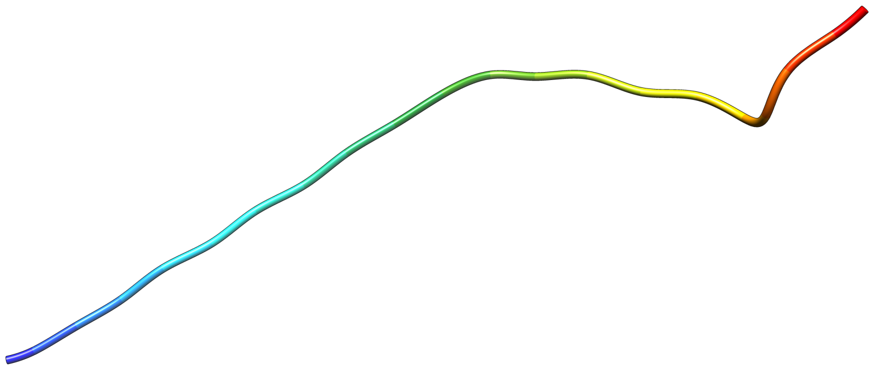


Figure 32: The Simulation Results of a 20 Amino Acid Protein Showing the Folding Structure After 500000 Iterations

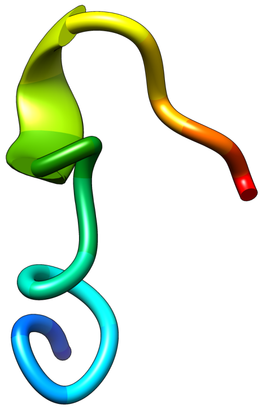


Figure 33: The Simulation Results of a 20 Amino Acid Protein Showing the Folding Structure After 3000000 Iterations

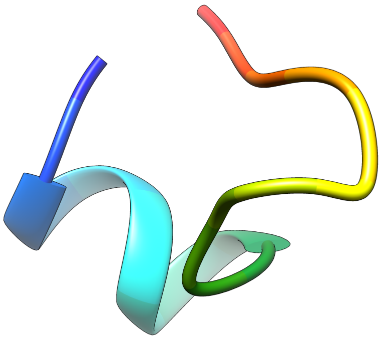


Figure 34: The Simulation Results of a 20 Amino Acid Protein Showing the Folding Structure After 4000000 Iterations

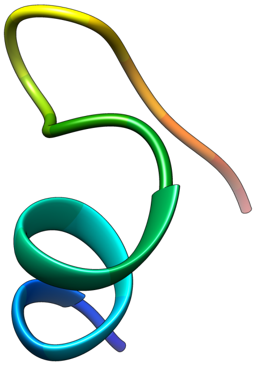


Figure 35: The Simulation Results of a 20 Amino Acid Protein Showing the Folding Structure After 5000000 Iterations

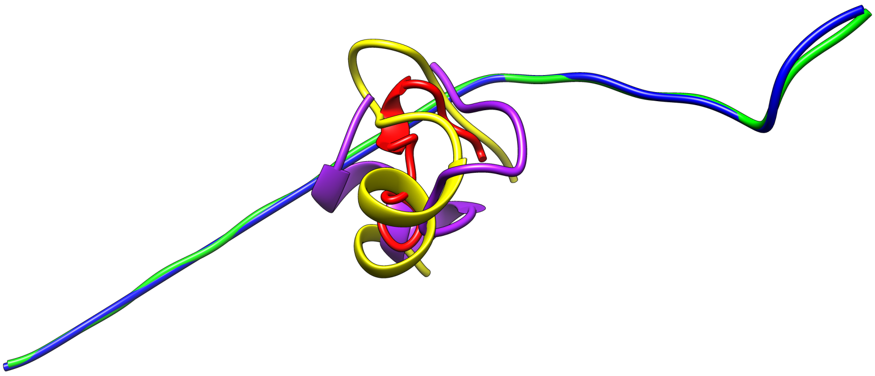


Figure 36: The Simulation Results of a 20 Amino Acid Protein Showing the Folding Structure at Different Time Steps Superimposed on Each Other. The Blue Shows the Initial Configuration and the Green, Red, Purple and Yellow Represent the Structures After 500000, 3000000, 4000000 and 5000000 Iterations Respectively

Chapter Summary

In this chapter, we performed a molecular dynamics simulation of a given number of atoms. We implemented the Velocity Verlet and the Euler's algorithms in MATLAB using the Lennard-Jones potential. The simulations were carried out under the microcanonical ensemble(the number of atoms, the volume and energy are constant). We analysed the results obtained from these simulations and then calculated and discussed some dynamical properties.

The results showed that the system becomes stable after some time steps which is usually dependent on the the number of atoms. The energies were found to be conserved for all systems. We also observed that the energies of systems with higher number of atoms were better conserved as compared to the systems with lower number of atoms. As the number of atoms increases, the potential energy falls. Also, the equilibrated temperature reduces as we increase the number of atoms in a system.

We also compared the computational speed of the Velocity Verlet and the Euler algorithms by implementing both methods in MATLAB. We run simulations for different systems using both algorithms for each given system. We observed from the results that, the simulations for the Euler's algorithm takes more time to complete as compared to the Velocity Verlet algorithm. There were some few exceptional cases. This means that, the Velocity Verlet algorithm is faster than the Euler's algorithm.

The molecular dynamics simulation technique was also used for an application to protein folding. We considered a small protein(peptide) of 20 amino acid sequence and performed a simulation on it using the AMBER package for different time steps to observe how it folds.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

Overview

Molecular dynamics simulation is a powerful research tool that is been used to replace experiments that are dangerous or expensive to carry out, to provoke, explain and understand experiments and bridge the gap between theory and experiment. It is applied to the mathematical modelling of many natural systems in physics, chemistry, biology, material sciences, engineering, economics, psychology and the social sciences.

In this thesis, we presented the theoretical background of molecular dynamics and implemented a typical MD simulation of N-atoms in MATLAB using Lennard-Jones model potential together with Euler and Velocity Verlet integration algorithms. We performed a comparative study between the Velocity Verlet and Euler's algorithms and then applied the concept of molecular dynamics to study the folding structure of a protein using AMBER package. In this chapter, we summarise the work and draw some conclusions from the findings. We also present recommendations for further studies.

Summary

Some simulations were run from the implementation of molecular dynamics concept and we observe from the results that the systems become stable after some given time steps usually dependent on the number of atoms. The energy is conserved for all the systems and the energies of systems with higher number of atoms were better conserved as compared to the systems with lower number of atoms. We went ahead to calculate and discuss some dynamical properties of the molecular dynamics simulation.

To compare the computational speed between Euler and Velocity Verlet integration algorithms, we implemented both methods in MATLAB and run simulations for different systems using both algorithms for each given system. The results showed that Velocity Verlet is faster than Euler's algorithm. There were some few exceptional cases.

An application of molecular dynamics simulation to protein folding was also presented. A small protein (peptide) of 20 amino acids sequence was simulated using the AMBER package to observe how it is folded.

Conclusions

The concept of molecular dynamics was presented, implemented in MATLAB using Lennard-Jones potential and a couple of simulations run from this implementation. The systems were found to be stable and the energies of systems with higher number of atoms are better conserved than that of systems with lower number of atoms. The analysis of the physical properties calculated from these simulations follows some already existing computational results.

The comparative study conducted between the Velocity Verlet and Euler's integration algorithms from the implementation showed that Velocity Verlet integration algorithm is faster than Euler's integration algorithm as most other research has claimed. This result will provide an informed decision for researchers in the choice of integration method.

Molecular dynamics was also applied to study the folding structure of protein of 20 amino acids sequence. The protein was found to be folded after the simulations. Knowing their structure form will enable the prediction of their biological functions. If for example, there is a viral protein with this sequence of amino acids, it will be possible to predict its replication since we know the structural form and this can help in drug design.

Recommendations

The results from the comparative study between Velocity Verlet and Euler's integration algorithms show that Velocity Verlet algorithm is better than Euler's algorithm in terms of computational speed. We therefore recommend that researchers should consider Velocity Verlet over Euler's integration algorithm in performing molecular dynamics simulation.

In this research, we considered a protein with random sequence of amino acids but every protein has a unique sequence of amino acids. In future research, molecular dynamics can be used to investigate the replication of a specific virus using its associated protein. Also, this study did not consider the interaction of protein with other molecules. Proteins behave differently in different environment such as water or viscous solvents. This is an interesting direction to investigate in future studies with respect to a specific protein relating to a virus.

REFERENCES

Alder, B. J., & Wainwright, T. E. (1957). Phase transition for a hard sphere system. *The Journal of Chemical Physics*, *27*(5), 1208-1209.

Allan, F., & Slobodan, P. (2016). Experiment in physics. In *The stanford encyclopedia of philosophy.* Stanford: Metaphysics Research Lab, Stanford University.

Allen, M., & Tildesley, D. (1989). *Computer simulation of liquids.* Oxford: Clarendon Press.

Atkins, P., & de Paula, J. (2006). *Atkins' physical chemistry* (8th ed.). New York: W. H. Freeman and Company.

Beveridge, D. L., McConnell, K. J., Young, M. A., Vijayakumar, S., & Ravishanker, G. (1995). Molecular dynamics simulations of dna and a protein-dna complex including solvent. *Molecular Engineering*, *5*(1), 255–269. Retrieved from `https://doi.org/10.1007/BF00999594` doi: 10.1007/BF00999594.

Biswas, R., & Hamann, D. R. (1987). New classical models for silicon structural energies. *Phys. Rev. B*, *36*, 6434–6445. Retrieved from `https://link.aps.org/doi/10.1103/PhysRevB.36.6434` doi: 10.1103/PhysRevB.36.6434.

Chakraborty, K., Mantha, S., & Bandyopadhyay, S. (2013). Molecular dynamics simulation of a single-stranded dna with heterogeneous distribution of nucleobases in aqueous medium. *The Journal of Chemical Physics*, *139*(7), 075103. Retrieved from `https://doi.org/10.1063/1.4818537` doi: 10.1063/1.4818537.

Cornell, W. D., Cieplak, P., Bayly, C. I., Gould, I. R., Merz, K. M., Ferguson, D. M., ... Kollman, P. A. (1995). A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, *117*(19), 5179-5197. Retrieved from `https://doi.org/10.1021/ja00124a002` doi: 10.1021/ja00124a002.

Daw, M. S., & Baskes, M. I. (1983). Semiempirical, quantum mechanical calculation of hydrogen embrittlement in metals. *Phys. Rev. Lett.*, *50*, 1285–1288. Retrieved from `https://link.aps.org/doi/10.1103/PhysRevLett.50.1285` doi: 10.1103/PhysRevLett.50.1285.

Daw, M. S., & Baskes, M. I. (1984). Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. *Phys. Rev. B*, *29*, 6443–6453. Retrieved from `https://link.aps.org/doi/10.1103/PhysRevB.29.6443` doi: 10.1103/PhysRevB.29.6443.

Dezfoli, A. A., Mehrabian, M. A., & Hashemipour, H. (2015). Molecular dynamics simulation of heavy metal ions in aqueous solution using lennard-jones 12-6 potential. *Chemical Engineering Communications*, *202*(12), 1685-1692. Retrieved from `https://doi.org/10.1080/00986445.2014.970251` doi: 10.1080/00986445.2014.970251.

Durrant, J. D., & McCammon, J. A. (2011). Molecular dynamics simulations and drug discovery. *BMC biology*, *9*, 71.

Dzugutov, M. (2001). Erratum: A universal scaling law for atomic diffusion in condensed matter (nature (1996) 381 (137-139)). *Nature*, *411*.

Encyclopaedia Britannica contributors. (2019). *Protein — Encyclopaedia Britannica, encyclopaedia britannica, inc.* `https://www.britannica.com/science/protein`. [Online; accessed 20-July-2019].

Ferguson, D. M., Radmer, R. J., & Kollman, P. A. (1991). Determination of the relative binding free energies of peptide inhibitors to the hiv-1 protease. *Journal of Medicinal Chemistry*, *34*, 2654-2549.

Frenkel, D., & Smit, B. (1996). *Understanding molecular simulation: From algorithms to applications.* San Diego: Academic Press.

Gunsteren, W. F. V., & Berendsen, H. J. C. (1988). A leap-frog algorithm for stochastic dynamics. *Molecular Simulation*, *1*(3), 173-185. Retrieved from `https://doi.org/10.1080/08927028808080941` doi: 10.1080/08927028808080941.

Heile, J. M. (1992). *Molecular dynamics simulations: Elementary methods.* New York: John Wiley.

Hernandez, E. R. (2008). Molecular dynamics: from basic techniques to applications (a molecular dynamics primer). *AIP Conference Proceedings*, *1077*, 95.

Jensen, E. J., Kristensen, W. D., & Cotterill, R. M. J. (1973). Molecular dynamics studies of melting : I. dislocation density and the pair distribution function. *The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics*, *27*(3), 623-632. Retrieved from `https://doi.org/10.1080/14786437308219236` doi: 10.1080/14786437308219236.

Karplus, M., & Kuriyan, J. (2005). Molecular dynamics and protein function. *Proceedings of the National Academy of Sciences*, *102*(19), 6679–6685. Retrieved from https://www.pnas.org/content/102/19/6679 doi: 10.1073/pnas.0408930102.

Karplus, M., & Petsko, G. A. (1990). Molecular dynamics simulation in biology. *Nature Publishing Group*, *347*, 631-639.

Kumar, H., & Maiti, P. K. (2011). Introduction to molecular dynamics simulation. In S. B. Santra & P. Ray (Eds.), *Computational statistical physics: Lecture notes, guwahati serc school* (pp. 161–197). Gurgaon: Hindustan Book Agency. Retrieved from https://doi.org/10.1007/978-93-86279-50-7_6 doi: 10.1007/978-93-86279-50-7_6.

Lennard-Jones, J. E. (1924). On the Determination of Molecular Fields. II. From the Equation of State of a Gas. *Proceedings of the Royal Society of London Series A*, *106*, 463-477. doi: 10.1098/rspa.1924.0082.

Levesque, D., & Verlet, L. (1993). Molecular dynamics and time reversibility. *Journal of Statistical Physics*, *72*, 519.

Lide, D. R. (1991). Handbook of chemistry and physics. In *CRC handbook of chemistry and physics.* Boca Raton, Florida, USA: CRC Press.

Los, J. H., Kroes, J. M. H., Albe, K., Gordillo, R. M., Katsnelson, M. I., & Fasolino, A. (2017). Extended tersoff potential for boron nitride: Energetics and elastic properties of pristine and defective *h*-bn. *Phys. Rev. B*, *96*, 184108. Retrieved from https://link.aps.org/doi/10.1103/PhysRevB.96.184108 doi: 10.1103/PhysRevB.96.184108.

Meller, J. (2001). *Molecular dynamics.* Encyclopedia of Life Sciences, Nature Publishing Group.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., & Teller, A. H. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*, 1087.

Oh, S.-K. (2013). Modified lennard-jones potentials with a reduced temperature-correction parameter for calculating thermodynamic and transport properties: Noble gases and their mixtures (he, ne, ar, kr, and xe). *Journal of Thermodynamics*, *2013*. doi: 10.1155/2013/828620.

Patodia, S., Bagaria, A., & Chopra, D. (2014). Molecular dynamics simulation of proteins: A brief overview. *J Phys Chem Biophys*, *4*, 6.

Perez, A., Luque, F. J., & Orozco, M. (2012). Frontiers in molecular dynamics simulations of dna. *Accounts of chemical research*, *45(2)*, 196-205.

Rahman, A. (1964). Correlations in the motion of atoms in liquid argon. *Physical Review*, *136(2A)*, A405-A411.

Rahman, A., & Stillinger, F. (1971). Molecular dynamics study of liquid water. *The Journal of Chemical Physics*, *55(7)*, 3336-3359.

Rajasekaran, G., Kumar, R., & Parashar, A. (2016). Tersoff potential with improved accuracy for simulating graphene in molecular dynamics environment. *Materials Research Express*, *3*(3), 035011. Retrieved from `http://stacks.iop.org/2053-1591/3/i=3/a=035011`.

Seco, J., Luque, F. J., & Barril, X. (2009). Binding site detection and druggability index from first principles. *Journal of Medicinal Chem-*

istry, *52*(8), 2363-2371. Retrieved from https://doi.org/10.1021/jm801385d (PMID: 19296650) doi: 10.1021/jm801385d.

Semironi, D. T., & Azimian, A. R. (2010). Molecular dynamics simulation of liquid-vapor phase equilibrium by using the modified Lennard-Jones potential function. *Heat and Mass Transfer*, *46*, 287-294. doi: 10.1007/s00231-009-0566-x.

Sha, Z., Branicio, P., Pei, Q., Sorkin, V., & Zhang, Y. (2013). A modified tersoff potential for pure and hydrogenated diamond-like carbon. *Computational Materials Science*, *67*, 146 - 150. Retrieved from http://www.sciencedirect.com/science/article/pii/S0927025612005344 doi: https://doi.org/10.1016/j.commatsci.2012.08.042.

Stillinger, F. H., & Weber, T. A. (1985). Computer simulation of local order in condensed phases of silicon. *Phys. Rev. B*, *31*, 5262–5271. Retrieved from https://link.aps.org/doi/10.1103/PhysRevB.31.5262 doi: 10.1103/PhysRevB.31.5262.

Stoddard, S. D., & Ford, J. (1973). Numerical experiments on the stochastic behavior of a lennard-jones gas system. *Phys. Rev. A*, *8*, 1504–1512. Retrieved from https://link.aps.org/doi/10.1103/PhysRevA.8.1504 doi: 10.1103/PhysRevA.8.1504.

Swegat, W., Schlitter, J., Kruger, P., & Wollmer, A. (2003). Md simulation of protein-ligand interaction: Formation and dissociation of an insulin-phenol complex. *Biophysical Journal*, *84*(3), 1493-1506. Retrieved from https://doi.org/10.1016/S0006-3495(03)74962-5 doi: 10.1016/S0006-3495(03)74962-5.

Swope, W. C., Andersen, H. C., Berens, P. H., & Wilson, K. R. (1982). A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *jcp*, *76*, 637-649. doi: 10.1063/1.442716.

Tersoff, J. (1986). New empirical model for the structural properties of silicon. *Phys. Rev. Lett*, *56(6)*, 632.

Tersoff, J. (1988). New empirical approach for the structure and energy of covalent systems. *Phys. Rev. B*, *37(12)*, 6991.

Verlet, L. (1967). Computer ”experiments” on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, *159*, 98–103. Retrieved from `https://link.aps.org/doi/10.1103/PhysRev.159.98` doi: 10.1103/PhysRev.159.98.

Vivo, M. D., Masetti, M., Bottegoni, G., & Cavalli, A. (2016). Role of molecular dynamics and related methods in drug discovery. *Journal of Medicinal Chemistry*, *59*, 4035-4061.

Wan, H., Hu, J., Li, K., Tian, X., & Chang, S. (2013). Molecular dynamics simulations of dna-free and dna-bound tal effectors. *PLOS ONE*, *8*(10), 1-12. Retrieved from `https://doi.org/10.1371/journal.pone.0076045` doi: 10.1371/journal.pone.0076045.

Wikipedia contributors. (2019a). *Amber — Wikipedia, the free encyclopedia.* `https://en.wikipedia.org/w/index.php?title=AMBER&oldid=903983597`. [Online; accessed 20-July-2019].

Wikipedia contributors. (2019b). *Protein — Wikipedia, the free encyclopedia.* `https://en.wikipedia.org/w/index.php?title=Protein&oldid=906221261`. [Online; accessed 20-July-2019].

APPENDIX

The 20 Amino Acids Found in Nature Together with their Three and One Letter Code

Amino Acid	Three letter code	One letter code
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartate	Asp	D
Cysteine	Cys	C
Glutamine	Gln	Q
Glutamate	Glu	E
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophane	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

Source: Lide (1991)