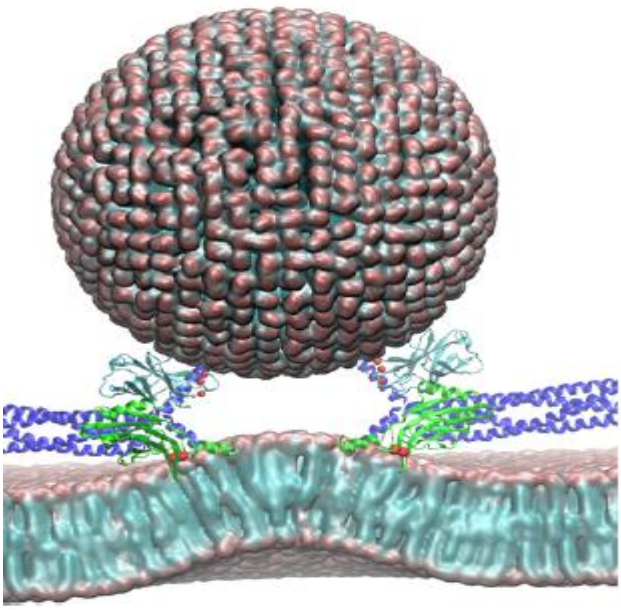
Molecular Dynamics Simulation

PREPARED BY

Amna Salah Mahmoud

UNDER SUPERVISION OF

Prof. Kholmirzo Kholmurodov



Contents

1 Introduction

1.1 What is molecular dynamics simulation?

A computational method which describes equilibrium and dynamics properties of atomic system generates configurations of the system by integration of Newton's laws of motion and calculate the time dependence of the molecular system. Generates information at the microscopic level such as atomic positions and velocities. Connects structure and function by providing additional information to X-ray crystallography and NMR. Newton's equations of motion for the system are integrated numerically. If the system is in equilibrium, static properties such as temperature and pressure are measured as averages over time. Dynamical properties such as heat transport, or relaxation of systems far from equilibrium, can also be studied. The fundamental work on this problem was done by A. Rahman, Phys. Rev. 136, A405 (1964). It was extended in many important ways by L. Verlet, Phys. Rev. 159, 98 (1967), who introduced the Verlet algorithm and the use of a neighbor list to speed up the calculation.

1.2 Historical Background

Introduced by Alder and Wainwrightin 1950's to study interaction of hard spheres (atoms that interact through perfect collisions). First simulation carried out by Rahman (1964) for liquid Argon. First protein simulation carried out by McCammon (1977) for bovine pancreatic trypsin inhibitor (BPTI). Today –solvated proteins, protein –DNA complexes, lipid systems, etc.

1.3 Statistical Mechanics

In molecular dynamics simulation we explore the macroscopic properties of a system through microscopic simulations. The connection between microscopic simulations and macroscopic properties is made via statistical mechanics, which studies a macroscopic system from a molecular point of view. The distribution of the system within the ensemble follows Boltzmann distribution.

1.4 Why Not Quantum Mechanics?

Modeling the motion of a complex molecule by solving the wave functions of the various subatomic particles would be accurate

$$\frac{-\hbar^2}{2m}\nabla^2\psi + \mathrm{U}(x,y,,z)\Psi(x,y,z) = E\Psi(x,y,z)$$

But it would also be very hard to program and take more computing power than anyone has.

1.1 Classical Mechanics

Instead of using Quantum mechanics, we can use classical Newtonian mechanics to model our system. This is a simplification of what is actually going on, and is therefore less accurate to alleviate this problem, we use numbers derived from QM for the constants in our classical equations.

1.2 The problem to solve

In atomistic simulations, the goal is to model, analyze and understand the motion of each atom in the material. the collective behavior of the atoms allows to understand how the material undergoes deformation, phase changes or other phenomena, providing links between the atomic scale to macro-scale phenomena. Extraction of information from atomistic dynamics is often challenging Vibration, change of location, connectivity and others

1.3 The Aims of Molecular Dynamics

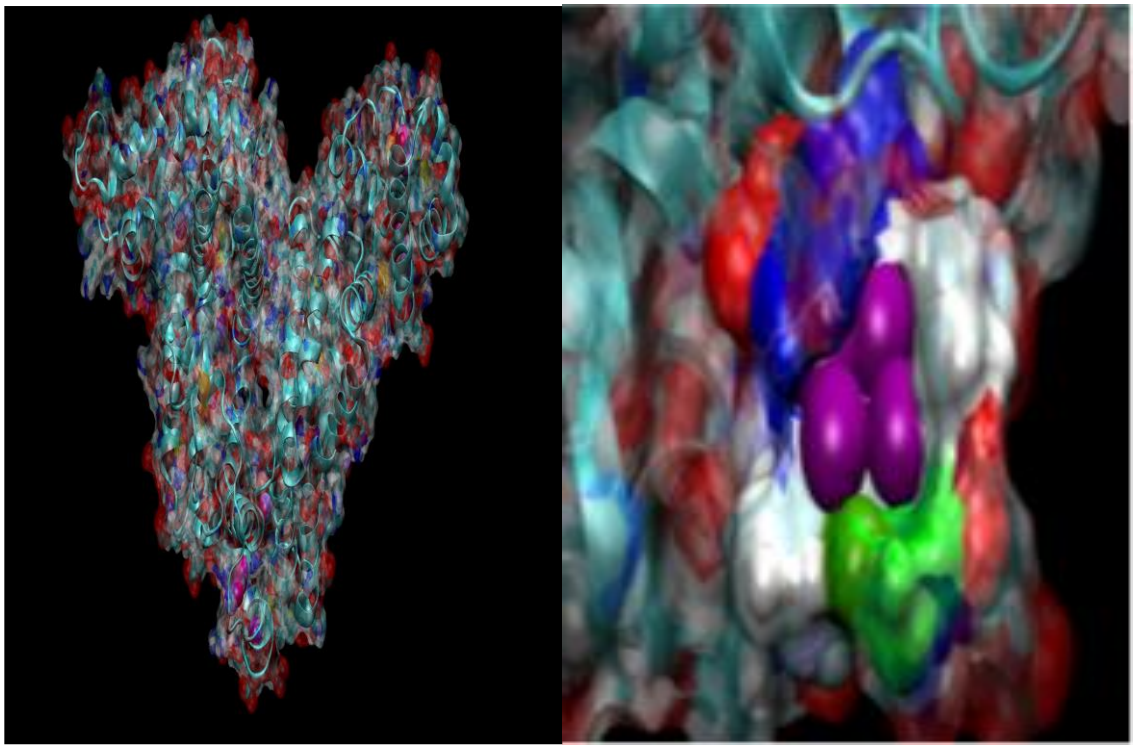
The obvious advantage of MD is that it gives a route to dynamical properties of the system: transport coefficients, time-dependent responses to perturbations, rheological properties and spectra.

Computer simulations act as a bridge between microscopic length and time scales and the macroscopic world of the laboratory: we provide a guess at the interactions between

molecules, and obtain ‘exact’ predictions of bulk properties. The predictions are ‘exact’ in the sense that they can be made as accurate as we like, subject to the limitations imposed by our computer budget. At the same time, the hidden detail behind bulk measurements can be revealed. An example is the link between the diffusion coefficient and velocity autocorrelation function (the former easy to measure experimentally, the latter much harder). Simulations act as a bridge in another sense: between theory and experiment. We may test a theory by conducting a simulation using the same model. We may test the model by comparing with experimental results. We may also carry out simulations on the computer that are difficult or impossible in the laboratory (for example, working at extremes of temperature or pressure).

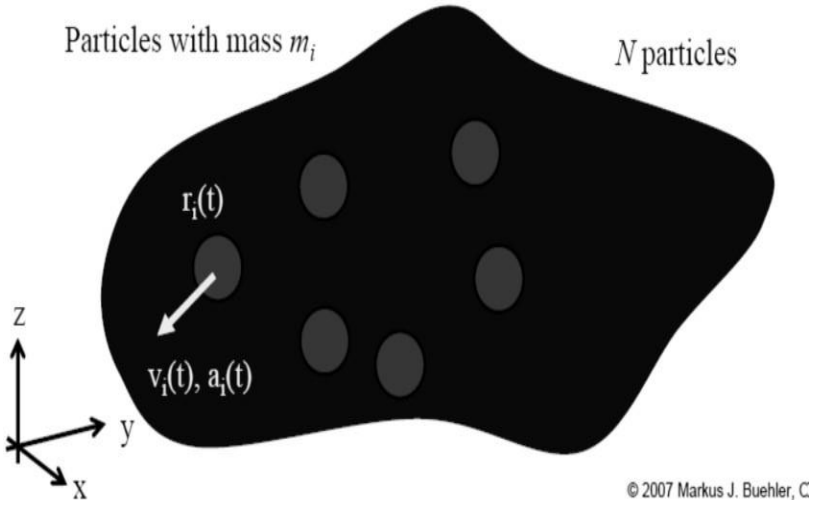
Ultimately we may want to make direct comparisons with experimental measurements made on specific materials, in which case a good model of molecular interactions is essential. The aim of so-called ab initio molecular dynamics is to reduce the amount of fitting and guesswork in this process to a minimum. On the other hand, we may be interested in phenomena of a rather generic nature, or we may simply want to discriminate between good and bad theories. When it comes to aims of this kind, it is not necessary to have a perfectly realistic molecular model; one that contains the essential physics may be quite suitable.

Using Molecular Dynamics, we can see what Biomolecules do, and what their role is in the biological system. That is the only method.



2 The basic equations and potentials

MD generates the dynamical trajectories of a system of N particles by integrating Newton's equations of motion, with suitable initial and boundary conditions, andproper interatomic potentials, while satisfying thermodynamical (macroscopic) constraints.



Total energy of system

$$E = K + U$$

$$K = \frac{1}{2}\sum_{j=1}^{N} mv_i$$

$$U = U(r_j)$$

2.1 Equations of motion & potential

The molecular dynamics simulation is based on Newton's law of motion:

$$F_i = m_i a_i$$

For each atom in a system constituted by N atoms. Here, mi is the atom mass, its $a_i = \frac{d^r r_i}{dt^2}$ acceleration, and Fi the force acting upon it due to the interactions with other atoms $ri = const$. That is, a molecule initially at rest will remain at rest and a molecule moving with a specified velocity will continue to move with that velocity until a force act on it. This is Newton's first Law.

Consider an isolated system that contains two spherical molecules,1 and 2 Hence, the total force is zero.

$$F_{total} = 0$$

Therefore, any force exerted by molecule 1 on molecule 2 must be balanced by a force exerted by 2 on 1.

$$F_{total} = F_1 + F_2 = 0$$

Hence,

$$F_1 = -F_2$$

This is Newton's third law. It's possible to determine the acceleration of each atom by knowing the force acting on each atom. Integration of the equation yields a trajectory that describes the positions, velocities and acceleration of each atom as they are varied with time. Once the positions and velocities of each atom are known, the state of the system can be predicted at any time. The force can be written as the gradient of the potential energy:

$$F_i = -\nabla_i V$$

Combine two equations to get:

$$-\frac{dV}{dr_i} = m_i \frac{d^2 r_i}{dt^2}$$

A trajectory is obtained by solving this differential equation.

2.2 What is the Potential?

A single atom will be affected by the potential energy functions of every atom in the system: bonded neighbors, and non-bonded Atoms (either other atoms in the same molecule, or atoms from different molecules).

$$V(r) = E_{bonded} - E_{non-bonded}$$

2.2.1 Non-Bonded Atoms

There are two potential functions we need to be concerned about between non-bonded atoms: Van Der Waals potential, and electrostatic potential

$$E_{non-bonded} = E_{van\ der\ waals} + E_{electrostatic}$$

2.2.2 The Van Der Waals Potential

Atoms with no net electrostatic charge will still tend to attract each other at short distances, as long as they don't get too close. Once the atoms are close enough to have overlapping electron clouds, they will repel each other with astounding force. One of the most widely used functions for the van der Waals potential is the Lennard- Jones. It is a compromise between accuracy and computability.

$$E_{lennard-jones} = \sum_{non-bonded\ pairs} \left(\frac{A_{ik}}{r_{ik}^{12}} - \frac{C_{ik}}{r_{ik}^{6}} \right)$$

The Constants A and C depend on the atom types, and are derived from experimental data

2.2.3 The Electrostatic Potential

Opposite Charges Attract Like Charges Repel The force of the attraction is inversely proportional to the square of the distance

$$E_{electrostatic} = \sum_{non-bonded\ pairs} \left(\frac{q_i q_k}{D r_{ik}}\right)$$

2.2.4 Coulomb's Law

$$F = \frac{q_1 q_2}{4\pi\varepsilon_\circ r^2}$$

Coulomb interaction decay slowly with distance, considered long range interactions r represents the distance between two atoms having charges and. ε represent the dielectric constant, a number relating the ability of a material to carry alternating current to the ability of vacuum to carry alternating current.

2.2.5 The Non-Bonded Potential

Combine the LJ and Electrostatic Potentials

$$E_{non-bonded} = E_{van\ der\ waals} + E_{electrostatic}$$

2.2.6 Bonded Atoms

There are three types of interaction between bonded atoms: stretching along, the bond bending between bonds and rotating around bonds

$$E_{bonded} = E_{bond-stretch} + E_{angel-bend} + E_{rotate-angel-bond}$$

2.2.7 Bond Length Potential

Both the spring constant and the ideal bond length are dependent on the atoms involved

$$E_{bonded} = \sum_{1,2\ pairs} k_b(r - b_o)^2$$

2.2.8 Bond Angle potential

Describe the deviation from an ideal bond angle geometry

$$E_{bond-bend} = \sum_{angels} k_\theta(\theta - \theta_o)^2$$

k_θ represent angle bending constant, θ0represent the deviation from the ideal bond angle

2.2.9 Torsion dihedral potential

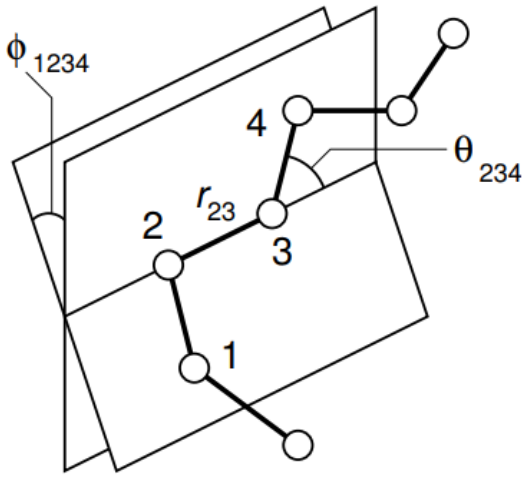
The motion associated is rotation, described by a dihedral angle around the middle bond. The potential is assumed to be periodic and expressed as a cosine function

$$E_{rotate-angel-bond} = \sum_{1,4} k_\emptyset(1 - \cos n\emptyset)^2$$

Represent rotation constant, n represent the periodicity of the rotational barrier and Φ the dihedral angle.

2.2.10 Hydrogen bonding potential

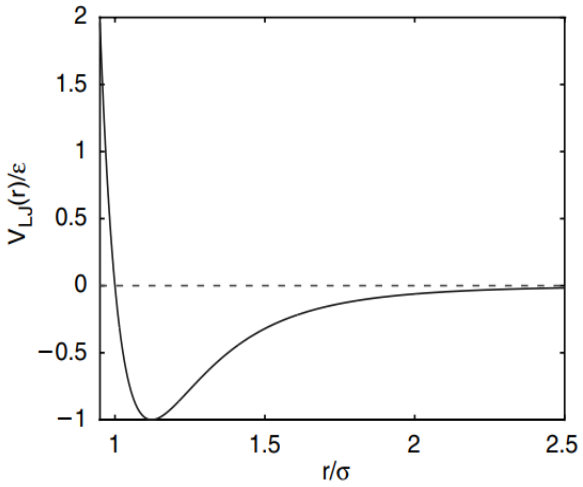
$$U_{HB} = \sum_{i,j} \left[\frac{A}{r_{i,j}^{12}} - \frac{B}{r_{i,j}^{6}} \right]$$



3 Lennard Jones potential

The interactions at the simplest level, occur between pairs of atoms and are responsible for providing the two principal features of an interatomic force. The first

$$u(r_{ij}) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right] & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases}$$



4 Methods

There are a number of software available for performing the molecular dynamic simulation of bio-molecules like GROMACS, Open Babel, VMD, UCSF Chimera, etc. We can select a software of our choice and perform the task but always remember that different software uses different Force Fields.

5 MD Simulation of Argon

5.1 Simple model of interacting Argon atoms

Consider N atoms of argon each with mass m = 6.69×10^{-26} kg. Argon is an inert gas: argons atoms behave approximately like hard spheres which attract one another with weak van der Waals forces. The forces between two argon atoms can be approximated quite well by a Lennard-Jones potential energy function

$$V(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{13} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

where r is the distance between the centers of the two atoms, ε = 1.65×10^{-21} J is the strength of the potential energy, and σ = 3.4×10^{-10} m is the value of r at which the energy is zero. The $1/r^{12}$ term represents a repulsive hard-core interaction between the argon atoms. The $1/r^{6}$ term represents an attractive dipole-dipole (van der Waals).

interaction between the non-polar atoms. The potential has its minimum

$$V\left(2^{\frac{1}{6}}\sigma\right) = -\varepsilon \qquad AT \quad r = 2^{\frac{1}{6}}\sigma$$

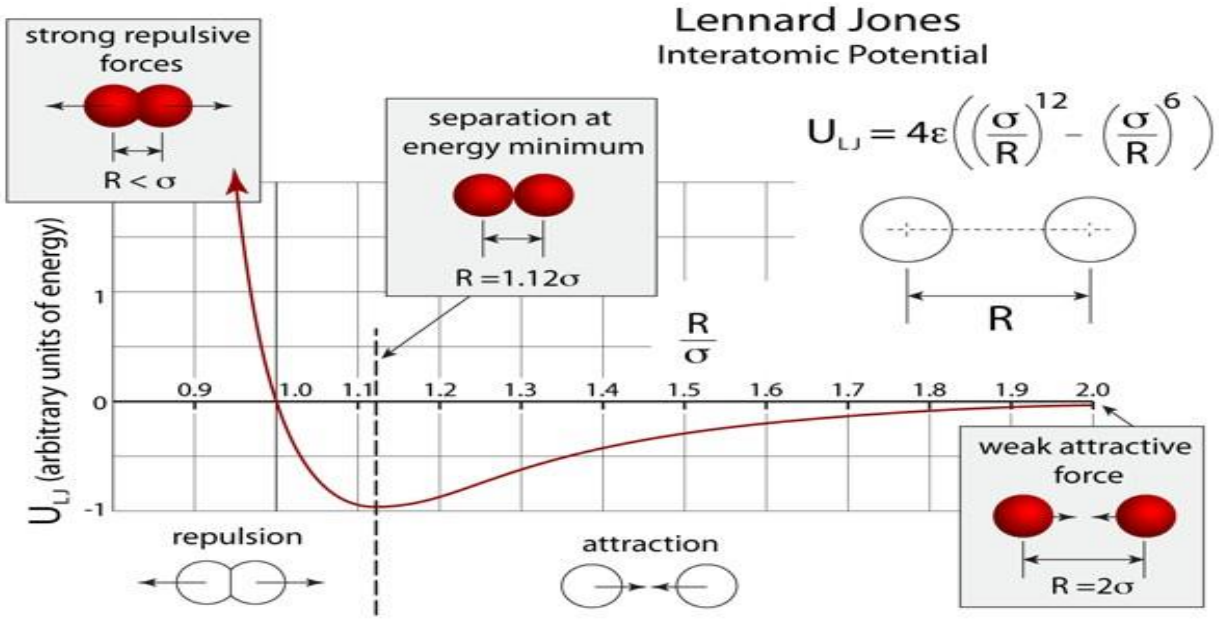
The shape of the potential and the strength of the Lennard-Jones force is:

$$F = -\frac{dV(r)}{dr} = \frac{24\varepsilon}{\sigma}\left[2\left(\frac{\sigma}{r}\right)^{13} - \left(\frac{\sigma}{r}\right)^{7}\right]$$

We will choose units of mass, length and energy so that m = 1, σ = 1, and ε = 1. The unit of time in this system is given by

$$\tau = \sqrt{\frac{m\sigma^2}{\varepsilon}} = 2.17 \times 10^{-12} s$$

which shows that the natural time scale for the dynamics of this system is a few picoseconds.



атом	ε/k_B (K)	σ (nm)
H	8.6	0.281
He	10.2	0.228
C	51.2	0.335
N	37.3	0.331
O	61.6	0.295
F	52.8	0.283
Ne	47.0	0.272
S	183.0	0.352
Cl	173.5	0.335
Ar	119.8	0.341
Br	257.5	0.354
Kr	164.0	0.383

5.2 MD program

First include some standard headers Define data structures to describe the kinematics of the system. We next need to set the initial positions and velocities of the particles. This is actually a complicated problem! Because the system can be simulated only for a few nanoseconds, the starting configuration must be very close to equilibrium to get good results. For a dense system, the atoms are usually placed at the vertices of a face- centered cubic lattice, which is tends to minimize the potential energy. The atoms are also given random velocities to approximate the desired temperature. In this preliminary program we will put the system in a cubical volume of side L and place the particles at the vertices of a simple cubic lattice.

5.3 The instantaneous temperature

This is a simulation in which the number of particles N and the volume L^3 of the system are fixed. Because the Lennard-Jones force is conservative, the total energy of the system is also constant. If the system is in thermal equilibrium, then Boltzmann's Equipartiton Theorem relates the absolute temperature T to the kinetic energy:

$$3(N-1) \times KT = \langle \frac{m}{2} \sum_{i=1}^{N} v_i^2 \rangle$$

5.4 Computer code for MD simulation of Argon

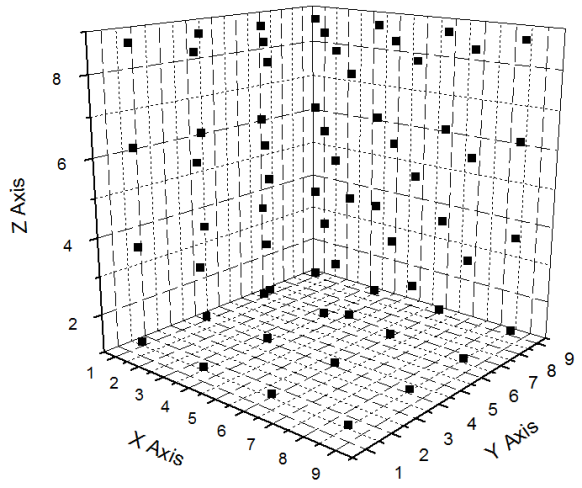
```
#include <cmath>
#include <cstdlib>
#include <fstream>
#include <iostream>
#include <string>
using namespace std;
const int N = 64; // number of particles
double r [N][3]; // positions
double v [N][3]; // velocities
double a [N][3]; // accelerations
double L = 10; // linear size of cubical volume
double Vmax = 0.1; // maximum initial velocity component
void initialize() {
// initialize positions
int n = int (ceil (pow (N, 1.0/3))); // number of atoms in each direction
 cout << "n=" << n << "      " ;
double a = L / n; // lattice spacing
cout << "a=" << a << "      " ;
```

```
int p = 0; // particles placed so far
for (int x = 0; x < n; x++)
for (int y = 0; y < n; y++)
for (int z = 0; z < n; z++)
{
if (p < N) {
r[p][0] = (x + 0.5) * a;
r[p][1] = (y + 0.5) * a;
r[p][2] = (z + 0.5) * a;
v[p][0] = Vmax * (2 * rand() / double(RAND_MAX) - 1);
v[p][1] = Vmax * (2 * rand() / double(RAND_MAX) - 1);
v[p][2] =  Vmax * (2 * rand() / double(RAND_MAX) - 1);
}
++p;
}
ofstream myfile("p.data");
for (int p = 0; p < N; p++)
 {
   myfile << r[p][0] << "     " <<  r[p][1] << "     " << r[p][2] << "     " << "\n";
   }
myfile.close();
ofstream rfile ("v.data");
for (int p = 0; p < N; p++)
{
   rfile << v[p][0] << "     " <<  v[p][1] << "     " << v[p][2] << "     " << "\n";
}
rfile.close();
}
void computeAccelerations() {
for (int i = 0; i < N; i++) // set all accelerations to zero
for (int k = 0; k < 3; k++)
a[i][k] = 0;
for (int i = 0; i < N-1; i++) // loop over all distinct pairs i,j
for (int j = i+1; j < N; j++) {
double rij[3]; // position of i relative to j
double rSqd = 0;
for (int k = 0; k < 3; k++) {
rij[k] = r[i][k] - r[j][k];
rSqd += rij[k] * rij[k];
}
double f = 24 * (2 * pow(rSqd, -7) - pow(rSqd, -4));
for (int k = 0; k < 3; k++) {
a[i][k] += rij[k] * f;
a[j][k] -= rij[k] * f;
}
}
}
void velocityVerlet(double dt) {
computeAccelerations();
ofstream nfile ("rv.data");
for (int i = 0; i < N; i++)
for (int k = 0; k < 3; k++)
 {
r[i][k] += v[i][k] * dt + 0.5 * a[i][k] * dt * dt;
if (r[i][k] > L)
  {
        r[i][k] = r[i][k] - int(r[i][k] / L) * L ;
```

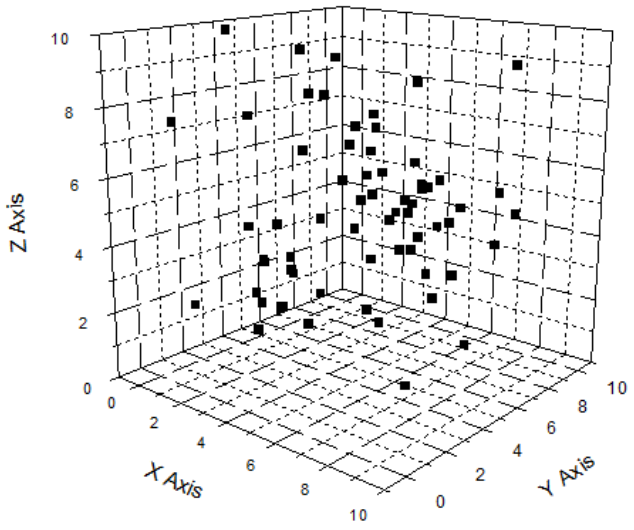
```
 }
if (r[i][k] < 0)
{
          r[i][k] = L - (r[i][k] - int(r[i][k] / L) * L) ;
          }
v[i][k] += 0.5 * a[i][k] * dt;
nfile <<  r[i][k] << "       " << v[i][k] <<"        " << "\n";
}
nfile.close();
computeAccelerations();
for (int i = 0; i < N; i++)
for (int k = 0; k < 3; k++)
v[i][k] += 0.5 * a[i][k] * dt;
}
double instantaneousTemperature() {
double sum = 0;
for (int i = 0; i < N; i++)
for (int k = 0; k < 3; k++)
sum += v[i][k] * v[i][k];
return sum / (3 * (N - 1));
}
int main() {
initialize();
 cout << "L=" << L << "      ";
 cout << "N=" << N << "      ";
 cout << "Vmax=" << Vmax << "      ";
double dt = 0.0001;
 ofstream file("T.data");
for (int i = 0; i < 200000; i++) {
velocityVerlet(dt);
file << instantaneousTemperature() << "\n";
}
file.close();
ofstream afile("a.data");
for (int i = 0; i < N-1; i++)
for (int j = i+1; j < N; j++)
 {
   afile <<  a[i][0] << "      " << a[i][1] << "      " << a[i][2] << "      " << "\n";
   afile <<  a[j][0] << "      " << a[j][1] << "      " << a[j][2] << "      " << "\n";
}
afile.close();
 ofstream tfile("pva.data");
for (int N = 0; N < 64; N++) {
tfile << r[N][0] << "      " << r[N][1] << "      " <<  r[N][2] << "      " << "\n";
tfile << v[N][0] << "      " << v[N][1] << "      " <<  v[N][2] << "      " << "\n";
tfile << a[N][0] << "      " << a[N][1] << "      " <<  a[N][2] << "      " << "\n";
}
tfile.close();
system ("pause");
}
```

5.5 Output of simple MD program

The 3D-configuration of 64 Ag atoms arranged in a simple cubic cell. After MD program run using Van-der-Waales potential we get the final configuration.



64 atoms of Argon arranged in simple cubic system.



The final configuration of Argon atoms after running simulation for 43.4 seconds.

6 Future Scope

As students of science, we all know that the macroscopic properties of elements owe a great deal to the time-dependent underlying microscopic properties and interactions among atoms and molecules. Molecular Dynamics Simulation has unbolted an incredibly huge number of doors for the research enthusiasts in the field of biological sciences, from the study of protein folding to the physics and chemistry behind the interactions among biomolecules, molecular docking to drug design, etc. The entire technique of MD Simulation depends solely on the trustworthiness of the model, force-field calculations and the thermodynamic property calculation and the ability of particular software to be able to mimic a process with as much reality as possible. Even though so many computer simulation techniques have been developed there is always the scope of improvement and simulation techniques obviously do not show cent percent accuracy in their results so better and more accurate techniques can always be developed. Also, with better results of the simulation process, the foundation for future studies on ion-exchange is being laid. Also, we need to develop more robust algorithms and one with a shorter number of steps. There is also scope for the development of lightweight and free software. Also, algorithms which are computationally less intensive are the need of the hour.

7 Conclusion

MD simulations have already more than 40 years of history. However, it was not until the recent years that MD has achieved time scales that begin to be compatible with biological processes. At present, when routine simulations are approaching the microsecond scale, conformational changes, or ligand binding can be effectively simulated. The improvement of the computational equipment, especially the use of GPUs, and the improvements made in the optimization of MD algorithms, including coarse-grained ones, allow us to move from the analysis of single structures, the basis of the molecular modeling as we know it, to the analysis of conformational ensembles. Conformational ensembles are a much better representation of real macromolecules, as they account for flexibility and dynamic properties (including all thermodynamic information) and ease the match with experimental results. Although the shift in concept is clear, and the technology is coming along, there is still a long way until biomolecular simulations, the generation of conformational ensembles, would become a routine. Tools exist that make the setup of a macromolecular system much easier, and even allow the nonexperts to enter the simulation world. However, lack of representation standards, much less optimized analysis tools, and even the difficulties in simply storing and transmitting the huge amount of trajectory data that is generated are still issues that remain to be solved. In any case, MD is already a valuable tool in helping to understand biology.

8 Acknowledgements

I would like to thank supervisor Prof. Dr. Kholmirzo Kholmurodov for giving me the opportunity to be part of a very interesting project, and teaching of introduction molecular dynamic simulation. His explanation and guidance helped me in all the time of work.

9 References:

Books

- Tamar Schlick Molecular Modeling and Simulation: An Interdisciplinary Guide 2002
- Alan Hinchliffe Molecular Modelling for Beginners 2003
- D. C. Rapaport The Art of Molecular Dynamics Simulation. 2004
- Daan Frenkel, B. Smit Understanding Molecular Simulation 2001
- M. P. Allen and D. J. Tildesley, Computer Simulation of Liquids (Oxford U. Press, Oxford, 1987).
- D. Frenkel and B. Smit, Understanding Molecular Simulation, 2nd Ed. (Academic Press, New York, 2001).
- © 2007 Markus J. Buehler, CEE/MIT
- Träskelin et al., J. Appl. Phys. 93 (2002) 1826
- W. Smith, T.R. Forester and I.T. Todorov, The DL POLY Classic User Manual, Version 1.0, December 2010.
- C. Educational, "Kholmirzo Kholmurodov MD-SIMULATION IN CHEMICAL RESEARCH : Educational level : MS," 2011.
- A. Rahman, "Correlations in the Motion of Atoms in Liquid Argon," Phys. Rev., vol. 136, no. 2A, pp. A405–A411, Oct. 1964.
- L. Verlet, "Computer 'Experiments' on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules," Phys. Rev., vol. 159, no. 1, pp. 98–103, Jul. 1967.

Websites

- Molecular Dynamics Tutorial at EMBnet http://www.ch.embnet.org/MD_tutorial/index.html
- Theoretical and Computational Biophysics Group at UIUC (home of VMD and NAMD) http://www.ks.uiuc.edu/
- http://www.weizmann.ac.il//sb/Pages/Levy/group meet/Molecular dynamics simulation.pdf7
- https://www.dovepress.com/molecular-dynamics-simulations-advances-and-applications-peer-reviewed-fulltext-article