Chapter 1

Computer Simulation of Molecular Systems

Contents

1.1 Introduction .................................................. 2
1.2 Nucleic Acids and their Dynamics ....................... 3
1.3 Molecular Modeling ......................................... 4
  1.3.1 Quantum Mechanical Methods ..................... 5
  1.3.2 Molecular Mechanics ............................... 8
  1.3.3 Molecular Dynamics Simulations ................. 13
1.4 MD Simulations of Nucleic Acids ....................... 26
1.5 MD Simulations using AMBER ......................... 26
1.6 Computational Resources ............................... 32
1.7 Overview of The Thesis ................................. 32
1.1 Introduction

The long-range goal of molecular biology is the description of living systems in terms of physics and chemistry. The first step towards this goal is the determination of the 3-dimensional structures of biomolecules of interest, which is essential for the understanding of their function. X-ray crystallography is one of the most important experimental methods to determine the atomic resolution structure of biomolecules like proteins and nucleic acids. NMR spectroscopy is another important technique used for the determination of biomolecular structures. X-ray diffraction studies on DNA fibers revealed the right handed double helical structures of A and B-DNA [1, 2]. Sequence dependent variations from the ideal A-form and B-form geometries [3] and the formation of left handed helices [4] are revealed from the X-ray analysis of crystals. There are a variety of nucleic acid structures other than double helix, which include single strands, bulged loops, hairpin loops, cruciforms, knots, and branches in the double helix.

The structures obtained from the experimental methods such as X-ray crystallography as well as NMR spectroscopy may give an idea that these molecules are relatively rigid. But actually all molecular systems have a dynamic nature as a result of the atomic motions. For a complete understanding of the function of biomolecules, we should know the kinetics of conformational changes and thermal equilibrium fluctuations occurring in the structures along with the 3D structures of the ground state and transition states.

Computer simulations, namely Molecular Dynamics (MD) and Monte Carlo (MC) simulations of the proteins and nucleic acids changed our view of rigid structures by a dynamic picture. The first MD simulation of a biological macromolecule was that of bovine pancreatic trypsin inhibitor (BPTI) published in 1975 by McCammon et al. [5]. This study was instrumental in replacing our view of rigid structures of the biomolecules by a dynamic picture. It is now
well established that the internal motions of biomolecules have an important role in their functions. The atomic motions and the associated changes in the active sites of proteins are, for example, important for the binding of ligands. Computer simulations can provide atomic level details of a molecular system as a function of time, which is hard to obtain experimentally. The MD simulation methods like simulated annealing methods are now commonly used for the X-ray structure refinement and NMR structure determination [6].

In the macromolecular realm, there are three important applications of simulations, namely the sampling of conformational space, description of the system at equilibrium, and the calculation of thermodynamic parameters, like binding free energy and entropy. In the first two cases, MD and MC simulation methods can be applied, but for the last type of applications, where the time evolution of the system is important, only MD methods can be applied.

1.2 Nucleic Acids and their Dynamics

DNA is the hereditary material in living organisms. The double helical structure of the nucleic acid has become an icon of molecular biology due to its role as the information encoding molecule. The biological functions of the DNA namely information storage, translation, and replication are linked to their sequence, energetics, structure and molecular motions. The study of the properties of nucleic acids is an interdisciplinary area with contributions from biology, biochemistry, physics, chemistry and computer science. Molecular dynamics simulation is one of the most rigorous quantitative techniques now available to model the dynamics of nucleic acids [7]. Recent developments in the methodology in modeling different interactions in the molecular system, mainly in treating electrostatic interactions, enables us to model nucleic acids which are highly charged polyanions.
1.3 Molecular Modeling

Molecular systems can be modeled using a variety of techniques like the mechanical ball and stick models and computer modeling. The mechanical model building is a very useful method as it can provide a three dimensional model of the system. The classic example of the usefulness of the mechanical model building is that by Watson and Crick to determine the structure of DNA. The development in computer graphics as well as in the computational power now allow us to use computational techniques in simulating actual molecular systems and to visualize the results in an interactive way.

The computer modeling of molecular systems can be broadly classified into quantum mechanical methods and molecular mechanics, which uses classical mechanics. In both cases a mathematical model is used to calculate the energy of the system as a function of its structure. In quantum mechanical methods we try to solve the Schrödinger equation of the molecular system, but the accurate solution of this equation is possible only for one electron systems like hydrogen atom and singly ionized helium. So approximation methods should be adopted for the solution of any molecular system. Even with these approximations, the method is applicable only to systems which consist less than \( \sim 100 \) atoms. Biomolecular systems contain relatively large number of atoms and are usually simulated with explicit representation of its solvent environment. So the application of quantum mechanical methods are not feasible in this case and we have to adopt the classical representation of the system with its limitations. These methods are called molecular mechanics. In molecular mechanics we use a potential energy function to represent the system.

In the following sections we give a brief overview of quantum mechanical methods, namely \textit{ab initio} and semi empirical methods and the basic theoretical background and practical aspects of molecular dynamics.
1.3. Molecular Modeling

1.3.1 Quantum Mechanical Methods

The most fundamental equation of a molecular system is the Schrödinger equation. Schrödinger equation tries to address the questions regarding the conformational properties and energetics of the system. That is, the positions of the electrons and nuclei of a molecule in space and the energetic properties like heat of formation, chemical reactivity, conformational stability, and spectral properties. The time dependent form of Schrödinger equation is

\[
\int \left\{ -\frac{\hbar^2}{2m} \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) + V \right\} \Psi (r, t) = i\hbar \frac{\partial \Psi (r, t)}{\partial t}
\] (1.1)

Here \(\Psi(r,t)\) is the time independent wavefunction which contains all the information of the system and \(\left\{ -\frac{\hbar^2}{2m} \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) + V \right\}\) is the Hamiltonian operator \(H\), the sum of kinetic and potential energies. The Hamiltonian of a molecule has the form,

\[
H = \sum_{i}^{\text{ele}} \frac{-\hbar^2}{2m_e} \nabla_i^2 + \sum_{A}^{\text{nuc}} \frac{-\hbar^2}{2m_A} \nabla_A^2 - \sum_{i}^{\text{ele}} \sum_{A}^{\text{nuc}} e^2 Z_A \frac{1}{r_{iA}} - \sum_{i,j}^{\text{ele}} \sum_{A,B}^{\text{nuc}} e^2 Z_A Z_B \frac{1}{r_{AB}} \] (1.2)

If we consider \(\Psi\) as a stationary state, the Schrödinger equation assumes the time independent form,

\[
H\Psi = E\Psi
\] (1.3)

Here \(E\) is the energy eigen value of the Hamiltonian operator \(H\) and \(\Psi\) is the wavefunction. The exact solution of this equation is possible only for hydrogen atom and singly ionized helium. The multi-electron wave functions like molecular wavefunctions are influenced by nucleus-electron attraction and this tends to spatially contract the electron density distribution towards the nucleus which make the orbitals larger and diffuse. Different types of approximation methods are required to solve the Schrödinger equation of molecular systems.
1.3.1.1 Born-Oppenheimer (BO) Approximation

The Born-Oppenheimer approximation [8], which is indispensable in quantum chemical calculations of molecular wavefunctions, consist of two steps. In the first step the nuclear kinetic energy is neglected, that is, the nuclear kinetic energy operator $T_n$ is subtracted from the total molecular Hamiltonian. In the remaining electronic Hamiltonian $H_e$, the nuclear positions enter as parameters. The electron-nucleus interactions are not removed and the electrons still "feel" the Coulomb potential of the nuclei fixed at certain positions in space. The second step of the BO approximation involves separation of vibrational, translational, and rotational motions. The eigenvalue $E$ is the total energy of the molecule, including contributions from electrons, nuclear vibrations and overall rotation and translation of the molecule.

1.3.1.2 Hartree-Fock Method

In Hartree-Fock method, the Hamiltonian considers each electron $i$, in the average field of all other electrons in the molecule. The Hamiltonian which describes this approximation is called the Fock operator, given by

$$F_i = -\frac{1}{2} \nabla^2 - \sum_{j=1,N} \frac{Z_j}{r_{ji}} + \sum_{l=1,k} 2J_l - K_i. \quad (1.4)$$

Here $J$ is the Coulomb’s integral which reflects the average interaction potential of electron $i$ due to all other electrons and $K$ is the exchange integral. $F_i$ is a function of one-electron molecular orbitals, as $J$ and $K$ are functions of one-electron orbitals.

In the Hartree-Fock method, the molecular wavefunctions are built from one electron molecular orbitals which are represented by basis sets. In Linear Combination of Atomic Orbitals (LCAO) approximation, the molecular
1.3. Molecular Modeling

Orbitals $\psi$ are built as a linear combination of one electron wavefunctions $\varphi$ as $\psi = \sum_{j=1}^{m} C_{i,j} \varphi_{i,j}$. Here $C_{i,j}$ are numerical coefficients that determine the amount of each one-electron wavefunction in each molecular orbital and $m$ is the size of the basis set. The molecular wavefunctions can be built from a set of one-electron wavefunctions and this set of one-electron wavefunctions is called a basis set. Slater Type Orbitals (STO) and Gaussian Type Orbitals (GTO) are two commonly used wavefunctions. GTOs are mathematically simpler than STOs but the accuracy is comparatively less.

1.3.1.3  *Ab initio* Methods

Hartree-Fock theory can be used to perform quantum mechanical calculations of molecular systems. These calculations can be classified into ab-inito methods and semi-empirical methods. *Ab initio* means from the first principles or from the beginning, and here we require only the physical constants like speed of light, Planck’s constant, masses of elementary particles etc. as inputs. *Ab initio* calculations use full Hartree-Fock equations, without neglecting or approximating any terms in the Hamiltonian.

1.3.1.4  Semi Empirical Methods

Semi empirical methods use approximations to Hartree-Fock theory and use empirical parameters to reproduce experimental results. Various approximations used are: account only for the valence electrons, neglect some integrals in the HF equation and the use of empirical parameters, use of minimal basis sets, and the use of non-iterative solution processes. Most common semi empirical methods are Parsier-Parr-Pople Molecular Orbital Theory (PPPMO), Extended Huckel Molecular Orbital Theory (EHMO), Complete Neglect of Differential Overlap (CNDO), Intermediate Neglect of Differential Overlap (INDO), Mod-
ified INDO, Modified Neglect of Diatomic Overlap (MNDO) Austin Model 1 (AM1), Parametric Model 3 (PM3), PM5, and Semiempirical *Ab initio* Model 1 (SAM1). Among this, AM1 and PM3 are the most commonly used methods.

### 1.3.1.5 Density Functional Theory (DFT)

In density functional theory, electron density is considered rather than the wavefunction. Instead of attempting to calculate the full electron wavefunction, DFT attempts to calculate the total electronic energy and the overall electron density distribution. A functional is a function whose argument is also a function and it enables the function to be mapped into a number. The ground state energy $E$ of an $N$ electron system is a functional of the electron density $\rho$ and $E$ is a minimum when evaluated with the exact ground state electron density. Commonly used functionals are B3LYP, B3PW91, PBE-96, VWN#91, and BP86.

### 1.3.2 Molecular Mechanics

In molecular mechanics, the electronic motions are ignored and the energy of the system is calculated as a function of the nuclear positions. The system is treated as a classical system which obeys Newton’s equations of motion [9]. Different interactions in the system are represented using a potential energy function, called the force-field. The force-field equation represents a multidimensional surface with $3N$ coordinates for a molecule containing $N$ atoms. This surface is called the potential energy surface or hypersurface.

Statistical mechanical methods can be used to derive a partition function from which the thermodynamic properties can be calculated, if all the minimum energy points on the energy surface can be obtained. This is possible only for relatively small molecules in gas phase. In the case of a macroscopic system,
which contains extremely large number of atoms, the number of minimum energy points on the potential energy surface are enormous and it is not possible to quantify the minimum energy points on the energy surface. Computer simulation techniques use small replications of these systems, with manageable number of atoms or molecules to study their properties. Molecular dynamics (MD) simulation and Monte Carlo simulation (MC) techniques are the two molecular mechanics techniques which will generate a number of possible conformations of the system, based on the potential energy function.

Molecular dynamics mimics the real dynamics of the system from which the time averages of different dynamical variables can be calculated. The first MD simulations used simple potentials like hard sphere potential, but more realistic models are used today. Newton’s equations of motion are used to propagate the dynamics; the force on each atom is computed and combined with the current positions and velocities to generate new positions and velocities. The atoms are moved to the new positions and then the new sets of forces are calculated and so on. That is, in an MD simulation, the successive configurations of the system are connected in time. This will provide a trajectory which describes how the dynamical variables changes with time.

Monte Carlo simulation also uses the potential energy function to generate the conformations of the system, but there is no time dependence to the generated configurations. Each configuration depends only on its predecessor and conformations are generated randomly. The Boltzmann factor of the conformation $\exp \left\{ -\frac{V(r^N)}{k_B T} \right\}$ is calculated where $V(r^N)$ is the potential energy of the system, calculated using the potential energy function. If the generated conformation have higher energy than that of its predecessor, a random number between 0 and 1 is generated and compared with the Boltzmann factor of the energy difference: $\exp \left\{ \frac{(W_{\text{new}}(r^N)-V(r^N))}{k_B T} \right\}$. If the random number is higher than the Boltzmann factor, the move is rejected and if it is lower the move is
accepted. This permits the move to higher energies and allows better sampling of the conformational space.

There are some important differences between molecular dynamics and Monte Carlo simulations. MD simulation gives information about the time evolution of the properties of the system but there is no time dependence in a Monte Carlo simulation. MD has kinetic energy contribution to the total energy but in Monte Carlo, only potential energy is calculated. MD usually samples microcanonical (NVE) ensemble, whereas Monte Carlo simulation samples canonical (NVT) ensemble.

1.3.2.1 Force-Fields

Molecular mechanics models evaluate the energy of the system as a function of atomic positions alone and ignore electronic motions. These methods are useful for dealing with biomolecular systems like proteins and nucleic acids which contains significantly large number of atoms where the quantum mechanical methods are not a practical choice. Force-field models are based on rather simple model of interactions within the system such as stretching of bonds, angle bending, rotations about bonds etc.

The potential energy equation as well as the parameters in the equation are together called a force-field. Force-fields are always empirical and there is nothing like a correct force-field. The common form of a force-field is [9],

\[
V (r^N) = \sum_{\text{bonds}} \frac{k_{li}}{2} (l_i - l_{i,0})^2 + \sum_{\text{angles}} \frac{k_{\theta i}}{2} (\theta_i - \theta_{i,0})^2 + \sum_{\text{torsions}} \frac{V_n}{2} (1 + \cos (n \omega - \gamma)) \\
+ \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left( 4 \varepsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \frac{q_i q_j}{4 \pi \varepsilon_0 r_{ij}} \right)
\] (1.5)
In this equation the values of $l$, $\theta$, $\gamma$ etc. are obtained from the 3-dimensional structure of the molecule and the values of $k_l$, $k_0$, $l_0$, $\theta_0$ etc. are the force-field parameters usually obtained from the experimental structures of molecules or from quantum mechanical calculations, which are assigned to reproduce the properties of the molecules like vibrational spectra and various structural parameters. The force-field parameters are determined by the properties of the particular atoms, like their atomic number, hybridization, and the atoms bonded with it (chemical environment). So different parameters allow for different types of atoms and different molecular connectivities to be treated with the same potential energy function. The force-field parameters are assigned to the atoms by specifying an atom type for the particular atom. Figure 1.1 shows the graphical representation of different terms in a force-field.

Figure 1.1: Representation of different terms in a molecular mechanical force-field.
The first term in the equation is the interaction between pairs of bonded atoms and the second term is the energy changes with the change of the angle between bonds, both modeled by a harmonic potential which gives increase in energy as the bond length or the bond angle deviates from the reference value. The third term is the torsional potential which models how the energy changes as the bond rotates. The fourth term is a non bonded term which is calculated between any two atoms of different molecule or of the same molecule which are separated by more than two bonds. These are the van der Waals interactions modeled by the Lennard-Jones potential and the electrostatic interactions modeled by the Coulomb potential. The treatment of electrostatic interactions is a crucial issue, especially in nucleic acids, because of the poly-anionic nature of the molecule. Particle mesh Ewald [10] method is an advanced method for treating electrostatic interactions, which uses the Ewald summation [11] method used in crystallography.

The transferability of the functional form as well as the parameters is an important feature of force-fields. MM2/MM3 force-fields of Allinger and co-workers [12, 13] are commonly used for small molecules. AMBER [14] and CHARMM [15] are the two commonly used force-fields for the simulations of proteins and nucleic acids. Even if AMBER and CHARMM have similar functional form, there are differences in the parameters and the philosophy of parametrization is also different. Other important force-fields commonly used for biomolecular simulations are GROMOS [16] and OPLS [17].

1.3.2.2 Energy Minimization

Energy minimization is one fundamental aspect of molecular mechanics, which determines the coordinates of the atoms for which the system is having minimum energy. If the function $f$ depends on a number of independent variables $x_1, x_2, \ldots, x_i$, minimization tries to find the values of those variable for which
the function has a minimum value. Mathematically, the first differential of the function $f$ with respect to the coordinates $\frac{\partial f}{\partial x_i} = 0$ and the second differential $\frac{\partial^2 f}{\partial x_i^2} > 0$.

The energy minimization algorithms can be classified into non-derivative methods, which does not use the derivative of energy with respect to the coordinates and derivative methods which in turn can be classified into first order methods and second order methods. Simplex method and sequential univariate method are two non-derivative methods. Conjugate gradient method and the steepest descent method are two commonly used minimization algorithms which belong to the first order minimization methods. Newton-Raphson method is an example of second order method.

Energy minimization will bring the system, typically to the nearest minimum to the starting conformation and so, it is not possible to sample the conformational space of the molecule with ordinary minimization techniques.

1.3.3 Molecular Dynamics Simulations

Molecular dynamics (MD) simulation, which provides a methodology for detailed microscopic modeling of molecular systems on the atomic scale, is a powerful and widely used tool in physics, chemistry and material science. It is a scheme for the study of the natural time evolution of the system which allows prediction of the static and dynamic properties of substances directly from the interactions between atoms and molecules.

MD simulations monitor time-dependent processes in molecular systems in order to study their structural, dynamic and thermodynamic properties by numerically solving an equation of motion.
1.3.3.1 Statistical Mechanics Basis of MD Simulations

The classical Hamiltonian $H$ which is a function of coordinates $\mathbf{r}$ and momenta $\mathbf{p}$ describes a classical system. For a regular molecular system, where the potential energy fluctuation is independent of time and velocity, the Hamiltonian is equal to the total energy,

$$H(r,p) = K(p) + U(r) = \sum_i \frac{p_i^2}{2m_i} + U(r) \tag{1.6}$$

$K(p)$ is the kinetic energy and $U(r)$ is the potential energy, $p_i$ and $m_i$ are respectively the momentum and mass of the $i$th particle. So the microscopic state of the system is characterized by a set of values $\{r,p\}$ which correspond to a point in phase space. The probability distribution of the system among the points in the phase space is given by the Boltzmann distribution function,

$$\rho(r,p) = \frac{\exp[-H(r,p)]}{Z} \tag{1.7}$$

The canonical partition function $Z$ is an integral over all phase space of the Boltzmann factors of $\exp[-H(r,p)/k_bT]$. Any dynamical variable of interest $A(r,p)$ can be calculated if this distribution function $\rho(r,p)$ is known. Since the averages of these dynamical variables take into account every possible state of the system, they are called thermodynamic averages or ensemble averages and is given by

$$\langle A(r,p) \rangle_Z = \int_{-\infty}^{\infty} dr \int_{-\infty}^{\infty} dp \rho(r,p) A(r,p) \tag{1.8}$$

But obtaining the thermodynamic averages, is computationally extremely difficult as it requires to know Boltzmann probability for every state simultaneously.

Another method is to follow the motion of a single point through phase space as a function of time and the averages are calculated only over those
points that are visited and the averages calculated in this way are called dy-
namic averages. Starting from a point \{r(0), p(0)\} the integration of equation of
motion of the system yields the set of points \{r(t), p(t)\} which describes the state
of the system at any successive time \(t\). Dynamic averages of any dynamical
variable \(A(r,p)\) can be calculated along the trajectory as,

\[
\langle A(r,p) \rangle_\tau = \frac{1}{\tau} \int_0^\tau A(r(t), p(t)) \, dt
\]  

where \(\tau\) is the duration of simulation.

Ergodic hypothesis \cite{18} states that for a stationary random process a large
number of observations made on a single system at \(N\) arbitrary instants of time
have the same statistical properties as observing \(N\) arbitrarily chosen systems
at the same time from an ensemble of similar systems. That is,

\[
\lim_{\tau \to \infty} \langle A(r,p) \rangle_\tau = \langle A(r,p) \rangle_Z
\]

By stationary we mean that the form of the probability distribution functions
does not depend on a shift of the origin of time.

If the trajectory is sufficiently long, we assume that the points generated
cover all of the phase space and the ergodic hypothesis holds. This is the
theoretical justification for using molecular dynamic simulations for calculating
thermodynamic averages of molecular systems.

### 1.3.3.2 Newtonian Molecular Dynamics

Newton’s second law is \(F_i = m_i \ddot{r}_i\) where \(F_i\) is the force acting on the \(i^{th}\) particle
of mass \(m_i\) and \(\ddot{r}\) is the second derivative of position \(r\) with time. The force
\(F_i\) is the gradient of the potential \(V_r\), our potential energy function given as the
equation of the force-field.
That is,

\[ F_i = -\nabla V_r \]  

(1.11)

which is a second order differential equation. The Newton’s equation can be written in a more general form using the Hamiltonian \( H \) of the system [18]. Then the equation of motion can be written as

\[ \dot{r}_k = \frac{\partial H(r, p)}{\partial p_k} \quad \text{and} \quad \dot{p}_k = -\frac{\partial H(r, p)}{\partial r_k}. \]

Now the second order differential equation is turned into two first order differential equations which are easier to solve.

Newton’s equations of motion have some characteristic properties like conservation of energy, conservation of linear and angular momentum. Another important property is the time reversibility. That is by changing the signs of velocities will cause the molecule to retrace its trajectory. These properties help to ensure that our numerical solution is correct.

1.3.3.3 Computational Algorithms

In molecular dynamics Newton’s equations of motion are solved by numerical integration techniques. Usually a finite difference method is used to solve a differential equation. The molecular coordinates and velocities at a time \( t + \Delta t \) are obtained from the molecular coordinates at an earlier time \( t \). The position at time \( t + \Delta t \) can be represented as a Taylor series expansion about the position at time \( t \).

\[ r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 + \ldots. \]  

(1.12)

Since the integration is done in a stepwise fashion, the above equation is rewritten in a discrete form. If \( r_n \) is the position at step \( n \) and \( r_{n+1} \) that at step \( n+1 \), then \( r_{n+1} \) can be written as

\[ r_{n+1} = r_n + v_n \Delta t + \frac{1}{2} \left( \frac{F_n}{m} \right) \Delta t^2 + O\left(\Delta t^3\right) \]  

(1.13)
where $O(\Delta t^n)$ is the terms of order $\Delta t^n$ or smaller. From this information, the velocity $v_{n+1}$ at time $n+1$ is roughly estimated as,

$$v_{n+1} = \frac{(r_{n+1} - r_n)}{2}$$  

(1.14)

Starting from position $r_n$, velocity $v_n$ and the force $F_n$ at time $n$, using these equations we can determine the position $r_{n+1}$ and velocity $v_{n+1}$ at time $n+1$. This method however results in large errors in results. There are some other more accurate algorithms.

The **Verlet algorithm** [19, 20] is based on two Taylor expansions, a forward expansion $(t+\Delta t)$ and a backward expansion $(t-\Delta t)$,

$$r_{n+1} = r_n + v_n \Delta t + \frac{1}{2} \left( \frac{F_n}{m} \right) \Delta t^2 + O(\Delta t^3)$$  

(1.15)

and

$$r_{n-1} = r_n + v_n \Delta t + \frac{1}{2} \left( \frac{F_n}{m} \right) \Delta t^2 - O(\Delta t^3)$$  

(1.16)

Adding the above two equations gives the new position,

$$r_{n+1} = 2r_n - r_{n-1} + \left( \frac{F_n}{m} \right) \Delta t^2 + O(\Delta t^4)$$  

(1.17)

This algorithm is executed in two steps: the current position $r_n$ is used to calculate the current force $F_n$ and then the current and previous positions $r_n$ and $r_{n-1}$ are used together with the current force $F_n$ to calculate the position in the next step $r_{n+1}$. This is repeated for every time step for each atom in the system for calculating the velocities,

$$v_n = \frac{r_{n+1} - r_{n-1}}{2\Delta t} + O(\Delta t^2)$$  

(1.18)
In Verlet algorithm, the position integration is quite accurate and is independent of the velocity propagation and it requires only a single force evaluation per integration cycle. This formulation also guarantees time reversibility as it is based on forward and backward expansions. But it has some disadvantages: relatively large errors in velocity propagation, on the order of $O(\Delta t^2)$, numerical errors due to the addition of $O(\Delta t^2)$ term to $O(\Delta t^0)$ term, and it is not a self starting algorithm.

The **leap-frog integrator** [19, 21] is a modification of the Verlet algorithm, especially to improve velocity evaluation, also called half-step algorithm. Velocities are evaluated at the mid point of position evaluation and positions at the midpoint of the velocity evaluation. The algorithm can be written as

\[
\begin{align*}
    r_{n+1} &= r_n + v_{n+1/2} \Delta t \quad (1.19) \\
    v_{n+1/2} &= v_{n-1/2} + \frac{F_n}{m} \Delta t \quad (1.20)
\end{align*}
\]

Here $v_{n\pm1/2}$ is the velocities at the mid steps $t \pm (1/2) \Delta t$. The leap-frog algorithm is executed in the following steps: Calculate the current force $F_n$ using the current position $r_n$. The current force $F_n$ and the previous mid-step velocity $v_{n-1/2}$ are used to calculate the next midstep velocity $v_{n+1/2}$ and from the current position $r_n$ and next midstep velocity $v_{n+1/2}$, the position in the next step $r_{n+1}$ is calculated.

Leap-frog integrator improves velocity evaluation and reduces numerical errors as $O(\Delta t^1)$ terms are added to $O(\Delta t^0)$ terms. The simulation temperature can be controlled by velocity scaling as the velocities are evaluated directly. It does not handle velocities in a satisfactory manner as velocities at time $t$ were approximated by the equation $v_n = \left(\frac{v_{n+1/2} + v_{n-1/2}}{2}\right)$ and is computationally a little more expensive than Verlet algorithm.
The velocity Verlet integrator [22] is a variant of Verlet algorithm, which calculates velocities better. This algorithm stores positions, velocities and accelerations at the same time \( t \) and minimizes round off errors. The basic equations are,

\[
    r_{n+1} = r_n + v_n \Delta t + \frac{1}{2} \left( \frac{F_n}{m} \right) \Delta t^2 \tag{1.21}
\]

\[
    v_{n+1} = v_n + \frac{1}{2} \left( \frac{F_n}{m} + \frac{F_{n+1}}{m} \right) \Delta t \tag{1.22}
\]

The Velocity Verlet algorithm proceeds by calculating the position \( r_{n+1} \) at time \( t+\Delta t \) and then calculates the velocity at mid-step

\[
    v_{n+1/2} = v_n + \frac{1}{2} \left( \frac{F_n}{m} \right) \Delta t \tag{1.23}
\]

Then calculate the force \( F_{n+1} \) at time \( t+\Delta t \) and velocity at \( t+\Delta t \)

\[
    v_{n+1} = v_{n+1/2} + \frac{1}{2} \left[ \frac{F_{n+1}}{m} \right] \Delta t \tag{1.24}
\]

So kinetic energy at \( t+\Delta t \) also can be calculated.

This algorithm is stable, easy to program and evaluates the velocities accurately, but it is computationally expensive.

The predictor-corrector algorithm [23] forms a general family of integration algorithms from which we can select a scheme that is correct to the required order. New positions, velocities, accelerations and higher terms are predicted according to the Taylor expansions and the forces are evaluated at new positions to give accelerations \( a(t+\Delta t) \). The difference between the predicted and the calculated accelerations are then used for correcting the positions, velocities etc. in the correction step.

\[
    \Delta a (t + \Delta t) = a^c (t + \Delta t) - a (t + \Delta t) \tag{1.25}
\]
Then the corrected positions, velocities, etc. are,

\[ r_c (t + \Delta t) = r (t + \Delta t) + c_0 \Delta a (t + \Delta t) \] (1.26)
\[ v_c (t + \Delta t) = v (t + \Delta t) + c_1 \Delta a (t + \Delta t) \] (1.27)
\[ a_c (t + \Delta t) / 2 = a (t + \Delta t) / 2 + c_2 \Delta a (t + \Delta t) \] (1.28)
\[ b_c (t + \Delta t) / 6 = b (t + \Delta t) / 6 + c_3 \Delta a (t + \Delta t) \] (1.29)

The set of coefficients \( c_0, c_1, c_2, \ldots \) depends upon the order of the Taylor expansion.

The selection of an appropriate integration algorithm for our purpose depends on various factors like computational cost, energy conservation of the system, memory required for calculations, synchronization of position and velocities, and whether the algorithm is self starting. For lengthy simulations we may require to use large time steps, but in some cases we must use small integration time steps for stability of simulation. For shorter time steps predictor-corrector methods may be more accurate, but for longer time steps Verlet algorithm is better [23].

### 1.3.3.4 Simulation at Constant Temperature and Pressure

Molecular dynamics simulation is commonly performed in NVE ensemble, however we may require to run simulation in other ensembles for various systems. There are methods to simulate the system in constant temperature and pressure, using some thermostats and barostats.

Berendsen coupling and Anderson coupling are two commonly used methods for constant temperature simulations. In the first method the velocity is rescaled to couple the system to a heat bath at a temperature \( T_0 \). Coupling the system to an external heat bath which supplies or removes heat energy from
the system will serve to keep the system at constant temperature. The rate of change of velocities is proportional to the difference in temperature between the bath and the system by scaling the velocities,

\[
\frac{dT(t)}{dt} = \frac{1}{\tau}(T_{\text{bath}} - T(t))
\]  

(1.30)

where \( \tau \) is the coupling parameter and higher \( \tau \) means weak coupling. The change in temperature between successive steps is

\[
\Delta T = \frac{\delta t}{\tau}(T_{\text{bath}} - T(t))
\]  

(1.31)

and the scaling factor

\[
\lambda = \left[ 1 + \frac{\delta t}{\tau} \left( \frac{T_{\text{bath}}}{T(t)} - 1 \right) \right]^{1/2}
\]  

(1.32)

In the method of Anderson the coupling to the heat bath is represented by stochastic impulsive forces that act occasionally on randomly selected particles.

In constant pressure simulation, the volume of the system must fluctuate. The method for keeping pressure a constant, proposed by Anderson, couples the system to an external variable, the volume of the system. The volume is coupled to a pressure bath with a coupling factor

\[
\lambda_p = 1 - \kappa \frac{\delta t}{\tau_p} (P(t) - P_{\text{bath}})
\]  

(1.33)

\( \kappa \) is the isothermal compressibility, the pressure equivalent of the heat capacity and \( \tau_p \) is the coupling constant.
1.3.3.5 Periodic Boundary Conditions

Periodic boundary conditions allow us to perform a simulation in a bulk fluid with relatively small number of particles. Here the system is represented by the central cell of any arbitrary shaped unit cell which can be periodically repeated to form an infinite system. Cubic cell is the simplest periodic system to visualize. In the central unit cell, any atom having position \( r_i = (x_i, y_i, z_i) \) now has infinite number of images with positions \( r_i+nL= [(x_i+n_1L)+(y_i+n_2L)+(z_i+n_3L)] \) for all the integer triples \( n = (n_1, n_2, n_3) \). In other words if a particle leaves the central cell, it is replaced by its image particle that enters from the opposite side. So the number of particles within the central cell remains constant. All the short-range interactions like van der Waals are usually restricted between the neighboring pairs. But long range electrostatic interactions are usually treated with particle mesh Ewald method.

1.3.3.6 Solvent Model

The treatment of the solvent environment of the system is an important issue. The most accurate method is to use explicit representation of the atomic models of the solvent molecules. The solvent of interest to biomolecular systems is water. There are three types of water models. Simple models with rigid geometry with interaction sites, flexible models which allow conformational changes in the molecule, and the models which explicitly include the effects of polarization and many-body effects. The water models commonly used for the explicit solvent representation are SPC [24], SPC/E, TIP3P and TIP4P [25]. These all are simple water models with 3 interaction sites per molecule and have rigid geometries.

The explicit representation of water is computationally demanding and so, implicit solvent models are used in many simulations where the interactions
with the solvent are not needed. In these models, the solvent effects will be incorporated in the force-field itself.

1.3.3.7 Assigning Initial Values

Newton’s equation of motion is a second order differential equation that requires a set of initial coordinates \( \{r(0)\} \) and a set of initial velocities \( \{v(0)\} \) for each degree of freedom in order to initiate the integration.

The initial coordinates \( \{r(0)\} \) are usually obtained from experimentally determined molecular structures by X-ray crystallography or NMR spectroscopy or from model building. There are algorithms for reasonable placement of hydrogen atoms in a crystallographic structure as it does not contain hydrogen atom positions.

In order to relieve local stresses due to non-bonded overlaps and to relax bond length and bond angle distortions in the initial structures, usually the structure is subjected to initial energy minimization.

Initial velocity distribution of atoms in a structure is determined by the system temperature \( T \). The initial velocities are usually assigned randomly from the standard Maxwellian velocity distribution at a temperature \( T \) as

\[
P(v)dv = \left( \frac{m}{2\pi k_b T} \right)^{1/2} \exp \left[ -\frac{mv^2}{2k_b T} \right] dv
\]  

(1.34)

In this random assignment process, there may not be any correlation between neighboring atoms and may accidentally assign high velocities to a localized cluster of atoms, creating a ‘hot spot’ that make the simulation unstable. To overcome this problem, velocities are initially assigned at low temperatures and then gradually increased to the desired temperature, allowing dynamic relaxation. This heating process and the simulation requires a measurable definition of system temperature \( T(t) \). According to the equipartition theorem,
the temperature $T(t)$ at any given time $t$ is defined in terms of mean kinetic energy by

$$T(t) = \frac{1}{k_B N_{\text{dof}}} \sum_{i=1}^{N_{\text{dof}}} m_i |v_i|^2$$

(1.35)

where $N_{\text{dof}}$ is the unconstrained degrees of freedom in the system. If $T_0$ is the desired temperature, then scaling the velocities by a factor of $[T_0/T(t)]^{1/2}$ will result in a mean kinetic energy corresponding to the temperature $T_0$.

Another problem of the initial velocity assignment is the formation of large total linear momentum $P$ and total angular momentum $L$, which causes a drift of center of mass of the molecules relative to the reference frame. This drift hampers the efficiency of the simulation and this problem is avoided by periodically zeroing these momenta during equilibration phase.

### 1.3.3.8 Integration Time Step

Small integration time step $\Delta t$ results in better integration quality, but the time required for a given length of simulation is large. So the time step selected is a trade-off between economy and accuracy. It should be small compared to the fastest motion in the system being simulated. If $\tau$ is the period of fastest motion then the integration time step $\Delta t \approx \tau/20$. For biomolecules the fastest motions are the stretching and vibrations of bonds connecting hydrogen and heavy atoms. The periods of such motions are about 10 fs and so the time step for integration is 0.5 fs. So the computer time required for a simulation of biomolecular system with explicit representation of water is extremely large. In order to avoid this, the bonds involving hydrogen were usually constrained using SHAKE [26, 27] or RATTLE [28] algorithms, which assumes that the length of $X-H$ bond can be considered constant. The deviation of the current length $d_k(t)$ of the $k^{th}$ bond from its ideal bond length $d_k^0$ must be smaller than
some tolerance value $\varepsilon,$

$$s_k = \frac{[d_k(t)^2 - d_0^2]}{d_0^2} < \varepsilon$$ \hspace{1cm} (1.36)

The SHAKE algorithm, the atomic positions are adjusted after each integration step iteratively until $s_k$ is smaller than $\varepsilon$ for all values of $k$. Using SHAKE algorithm to bonds involving hydrogen, integration time step of 2 fs can be used, and this will allow us to do simulations of longer time scales. The schematic flowchart of a molecular dynamics is shown in Figure 1.2 [29].

Figure 1.2: The flowchart of a typical MD simulation.
1.4 MD Simulations of Nucleic Acids

Nucleic acids are central to molecular biology as they are the information encoding molecules. The interactions of nucleic acids with proteins and other molecules are very important and an understanding of their structure and dynamics are essential for getting an insight into this.

The influence of solvent and ion environment on structure of nucleic acids is evident from the variety of canonical DNA forms namely A, B, C and Z under different solvent and ion conditions [30]. So the treatment of the electrostatic interactions and the solvent environment representation of counterions is very important in nucleic acid simulations. The recent developments in computational power now allows us to incorporate accurate evaluation of electrostatic interactions and explicit representation of solvent and counterions. This along with the developments in nucleic acid force-fields enable us to obtain stable simulations of nucleic acid systems.

1.5 MD Simulations using AMBER

In order to perform a molecular dynamics simulation, we require a potential energy function and the associated parameter set, which is collectively called the force-field. The force-field which we use, should be parametrized to reproduce the properties of the system we wish to simulate. Along with the force-field we need a program to solve the Newton’s equations of motion by numerical integration to propagate the dynamics of the system. Commonly used programs for biomolecular MD simulations are AMBER [14], CHARMM [15], GROMACS [16], NAMD [31], etc. We have used AMBER suite of programs to do the MD simulations in this thesis. The name AMBER is generally used for the suite of programs as well as the force-field used for biomolecular simulations.
To perform a MD simulation of a system, we need the 3-dimensional structure of the molecule we wish to simulate, obtained from experimental techniques or from model building. In AMBER, we require the input structure in *pdb* format, taken from some databases like protein data bank (PDB) [32] or nucleic acid database (NDB) [33], which store the structure of proteins and nucleic acids, solved using X-ray crystallography or NMR spectroscopy. We can also use the structures obtained by model building: for example from homology modeling of protein structure or using software like NUCGEN, for the generation of nucleic acid structures.

Once we have a reasonable 3-dimensional structure of the system of interest, we have to assign the force-field parameters. In other words we have to provide information like bond parameters, angle parameters, torsion angle parameters, improper torsions and partial atomic charges. In practice all these informations are provided by assigning an atom-type to each atom in the molecule, which is determined by various factors like the hybridization of the atom, atoms attached to it, its chemical environment etc. All these information for the standard residues of proteins and nucleic acids as well as some solvents and ions are usually included in the program database.

We have to assign the force-field parameters of modified and non standard residues using some methodology which is consistent with the original parametrization, since these parameters are not available in the program database. For this, the molecule is geometry optimized using some quantum mechanical software like GAUSSIAN [34] or GAMESS [35] at the required level of theory, usually 6/31-G*. The electrostatic potential (ESP) of the molecule is obtained from the quantum mechanical calculations. The ESP is used for assigning the partial atomic charges, which is usually done with restrained electrostatic potential fit (RESP) [36]. These things can be done with the ANTECHAMBER [37] module from the AMBER suite. From ANTECHAMBER,
we can produce prepi and frcmod files which can be loaded into the LEaP for modifying the default force-field parameters.

The main programs used for a MD simulation in AMBER can be classified as preparatory programs, simulation programs and analysis programs. LEaP and ANTECHAMBER are the two preparatory programs which are used for preparing the inputs for the simulation programs. The main simulation programs are SANDER, PMEMD and NMODE. PTRAJ and MM-PBSA are the two main analysis programs. In addition to these we may require some other programs, which are not part of the AMBER suite, depending upon our aim. GAUSSIAN, DOCK 6 [38], CHIMERA [39], VMD [40], MMTSB [41], and IED [42] are the programs we have used along with AMBER suite for parametrization, preparation, and for analysis. Figure 1.3 shows the flowchart of the methodology we have used for the simulations.

If our molecule contains only the standard residues or if we have loaded all the parameters for the non-standard residues to LEaP, we can load the coordinate file of the molecule into LEaP in pdb format. From LEaP we can save the parameter-topology file of our system, which contains the force-field parameters and partial atomic charges, and the input coordinate file, which is the starting coordinates of the molecule. If we are performing an implicit solvent calculation, only our molecule is loaded into LEaP and the required files are saved. If we are using explicit solvents, we have to add counterions to charge neutralize the system along with pre-equilibrated solvent molecules. The water model usually used is TIP3P water. Once the force-field parameters are assigned successfully, we can save the parameter-topology file and the starting coordinate files, which are the two input files necessary for the simulation.
1.5. MD Simulations using AMBER

Figure 1.3: Flow-chart of MD Simulation and analysis using AMBER suite of programs.

SANDER, the main MD engine and energy minimizer in AMBER, require the parameter-topology file as well as an input file which contains the procedural options to do the minimization and dynamics. In the input file we can specify our options like minimization or MD, boundary conditions, cutoffs, heating procedures, pressure, type of ensemble, how to treat electrostatic interactions, integration time steps, how many steps of dynamics etc.

PMEMD is another dynamics engine in amber suite, which can only be used for simulations with particle-mesh Ewald method to treat electrostatic interactions. This has a better scalability for parallel processing. NMODE is another program, used for normal mode analysis, which uses Newton-Raphson method.
The main analysis program in AMBER suite is PTRAJ, which can be used to post process the trajectory generated by SANDER. Using PTRAJ we can calculate the rms deviation of the trajectory from the starting structure and can be used for hydrogen bond and hydration analysis and for imaging solvent molecules into the initial box, calculating and analyzing covariant matrices, to track various distances and angles etc. Using PTRAJ, we can strip solvent and the counterions if required, or can keep the desired number of solvent molecules, usually a particular number of solvent molecules which are closest to the solute. It can be used to change the file format of the trajectory, that is, we can change the trajectory from the AMBER trajectory format to various other formats like *binpos* or *pdb* format. Particular frames from the trajectory can be extracted for analysis at the desired frequency for further analysis or for visualization. Using the grid option in PTRAJ we can count the number of solvent or the counterions on the grid points around the solute, which can be used for visualizing the hydration of our molecule.

MM-PBSA is post processing script which is commonly used for thermodynamic analysis of the simulated system. It can be used for the determination of relative free energy of binding of ligands with receptors. This is a continuum solvent method, which uses Poisson-Boltzmann (PB) solver or Generalized Born (GB) method for calculating the internal energy components. The binding energy can be calculated using MM-PBSA script which uses a number of programs like SANDER, NMODE, PBSA, DELPHY etc. for calculating various contributions to the binding energy.

For MM-PBSA analysis we require the snapshots of the receptor, ligand and the complex, which were extracted from the trajectory at a particular time interval. There are two strategies which can be used for obtaining these; the single trajectory approach and the multiple trajectory approach. In single trajectory approach, all the three sets of snapshots are extracted from the
single trajectory of the complex. Here we assume that there may not be much structural adaptation due to the binding of the ligand with the receptor. In separate trajectory method three separate MD simulations, ie simulation of the receptor, ligand and complex are performed and the snapshots are extracted from the trajectories.

Some software which are not the part of AMBER suite were also used for the analysis of results. VMD (Visual Molecular Dynamics) is used for the visualization of the trajectory generated and for rendering the figures. MMTSB (Multiscale Modeling Tools for Structural Biology) is used as a conformational sampling tool. Using MMTSB, the conformations generated in the MD simulations were clustered into different groups based on the RMS deviation from the centroid. This is a very useful method for analyzing the results from the simulation as the number of conformations generated are very large and the visual observation and structural parameter calculation of all the conformations generated is not feasible. The structure with least rms deviation from the centroid of the cluster is taken as the representative structure of the cluster and the structural parameters of these structures were calculated.

Principal component analysis (PCA) is a statistical method commonly used to reduce the dimensionality of a data set. Each principal component corresponds to an axis in the n-dimensional space and each one is orthogonal to all other principal components. The visualization of a MD trajectory is not that easy as it contains a lot of thermal motions of all the atoms in the system. PCA can be applied to the MD trajectory to visualize the dominant essential motions in the system, which is termed as the essential dynamics. In order to determine the essential motions, the covariant matrix of the trajectory is calculated and the eigen values of these matrices are determined. Interactive Essential Dynamics (IED) is a script which work with the visualization software VMD. Using IED, the structure, the eigen values and the projections of each snapshots into the
eigen vectors can be loaded into VMD and the motion of the system along any
eigen vector can be visualized.

The important structural parameters of nucleic acid duplexes are helical
parameters [99]. The helical parameters of the structures were calculated with
3DNA [68]. The energy terms and other information from the output file from
dynamics run were extracted and the graphs were plotted with the plotting
program XMGRACE.

1.6 Computational Resources

The initial set up of the simulation systems and analysis of the trajectories
were done on Intel Pentium processor based systems running Redhat 9 and
Fedora 10. The MD simulations were performed on solaris system based
on at Supercomputing Facility for Bioinformatics & Computational Biology
(SCFBIIO), IIT Delhi and on Xeon processor cluster, running Redhat Enterprise
Linux 3, at Mahatma Gandhi University Computer Center.

1.7 Overview of The Thesis

Chapter 2- ‘Molecular Dynamics Simulations of xDNA’, describes the MD
simulations of xDNA (expanded DNA), a modified DNA. The starting structure
is taken from the nucleic acid database. The force-field parameters of the
modified bases are derived using GAUSSIAN98W and ANTECHAMBER 1.26.

In chapter 3- ‘Molecular Dynamics Simulations of the Stem and the
Adenine Bulge of the SL2 of HIV-1 Ψ-RNA’, the results of three independent
simulations of the stem and adenine bulge of HIV-1 SL2 were reported. The
initial coordinates are from the X-ray structure, taken from the NDB and the
parameters from the program database were directly used.
1.7. Overview of The Thesis

In chapter 4- ‘Molecular Dynamics simulations and Binding Free Energy Analysis of DNA-Curcumin Complexes’, the DNA structures were taken from the NDB and the curcumin structure was taken from the Cambridge Crystallographic Data Center (CCDC). The structures of the complexes are obtained by docking the curcumin molecule to the DNA duplexes using DOCK 6. The parameters for the curcumin are developed with ANTECHAMBER1.26 and DIVCON. The thermodynamic analyses of the complexes are done with the MM-PBSA methodology.

In chapter 5- ‘Molecular Dynamics and Binding Free energy analysis of Curcumin and its Derivatives Complexed with DNA sequences’, are presented. The starting DNA structures were built with the NUCGEN program in AMBER suite and the complexes were generated by docking with DOCK 6. The force-field parameters of the curcumin and its derivatives were developed with ANTECHAMBER and DIVCON. MM-PBSA method was used for the thermodynamic analysis. The computational methodology used are discussed in detail in each chapter.