MOLECULAR SIMULATION METHODS FOR pH RESPONSIVE MOLECULES

A DISSERTATION

Submitted in the partial fulfilment of the

requirements for the award

of

INTEGRATED DUAL DEGREE

(Bachelor of Technology and Master of Technology)

in

CHEMICAL ENGINEERING

(With Specialization in Hydrocarbon Engineering)

Submitted By

PRAMANSHU RAJPUT

DEPARTMENT OF CHEMICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247667 (INDIA)

MAY, 2016

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in this dissertation report entitled "Molecular Simulation Methods for pH Responsive Molecules" in partial fulfilment of the requirements for the award of the degree of Master of Technology in Chemical Engineering (IDD) with specialization in Hydrocarbon Engineering, and submitted in the Department of Chemical Engineering of Indian Institute of Technology Roorkee, India, is an authentic record of my own work carried out during April 2015-April 2016, under the supervision of **Dr. Prateek Kumar Jha**, Assistant Professor, Department of Chemical Engineering, Indian Institute of Technology Roorkee India.

The matter embodied in this dissertation has not been submitted by me for the award of any other degree of this or any other Institute/University.

Date: 6th May 2016

Place: Roorkee (Pramanshu Rajput)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Dr. Prateek Kumar Jha
Assistant Professor
Department of Chemical Engineering
Indian Institute of Technology Roorkee
Roorkee-247667, India

ACKNOWLEDGEMENTS

I express my deep sense of gratitude and indebtedness to my revered guide **Dr. Prateek Kumar Jha**, Assistant Professor, Department of Chemical Engineering, Indian Institute of Technology Roorkee, India, who provided her whole hearted co-operation, never ending inspirations and guidance, all blended with the personal touch throughout the duration of this work. His invaluable suggestions and through discussions have immensely contributed towards the completion of this work.

I take this opportunity to put on record my respects to **Dr. C.B.Majumder**, Head of Department of Chemical Engineering, Indian Institute of Technology Roorkee, for providing various facilities during course of the present investigation.

The last but not least I am grateful to my family members and friends for their love, suggestions and moral support without which I would not have achieved this goal.

Pramanshu Rajput

Table Of Contents

Table of Figures

ABSTRACT

pH is an important parameter in many chemical systems, as it determines the protonation state of an ionisable site in a molecule and thus affects the structure, dynamics and function of the molecule in a solution. Although the Henderson-Hasselbalch equation is widely used to determine the ionisation state of a molecule at a given pH, it is based on several approximations and do not account for structural changes at the molecular scale and the local environment of the molecule. Molecular simulations have the potential to bridge this gap and thus use as a predictive tool to determine the pH-effect in novel molecules. However, in conventional molecular simulations, the protonation state of a system is fixed (and not the pH) and cannot adapt to the local environment. In this study, we have developed a more realistic and thermodynamically rigorous scheme, where the protonation state of an ionisable site in a molecule is allowed to change by continuous protonation-deprotonation processes, as will be expected at a given pH condition. This method is based on a λ−dynamics approach, where we introduce an additional dimensionless degree of freedom ("particle"), λ, for every ionisable site indicating its protonation state ($\lambda \approx 0$ for fully protonated and $\lambda \approx 1$ for fully deprotonated). This λ particle is propagated in time by the Newton's equation of motion using the interpolated forces between protonated and deprotonated states. We treat each ionisable site as a mixed state – linear combination between deprotonated and protonated states; free protons are not handled explicitly. This method relies on pre-calculated empirical functions for each ionisable group (reference free energy simulations), which are obtained by running multiple conventional molecular dynamics (MD) simulations on our system. These constant pH simulations are computationally intensive as the energetics of charge changes upon protonation and deprotonation must be rigorously modelled and such simulations must sample large number of protonation states to give reproducible results. For performing the simulations, we first developed and tested our own library of MD code in C++ and then incorporated it as a patch of an open-source MD software, GROMACS in order to make it more versatile and computationally efficient. Simulations were first performed on a simple and weak acid, hydrogen fluoride, to test this method and do a parametric study. This was followed by the simulations for a relatively complex molecule, acrylic acid, to test the accuracy of this method. The pK_a value predicted by this method is are found to have good agreement with experimental results. The method is also able to reproduce the titration curves for these acids. Finally, we performed some preliminary

simulations of polyacrylic acid, which could not be completed due to time constraints. Nonetheless, this study provides a strong foundation and a detailed parametric study to simulate more complex molecules.

1. INTRODUCTION

Molecular modelling is the general process of describing complex chemical systems in terms of a realistic atomic model, with the goal being to understand and predict thermodynamic and structural properties of materials at an atomic scale resolution. Often, molecular modelling is used to study novel materials prior to their synthesis, for which the accurate prediction of physical properties is required to access their efficiency in the targeted application. A variety of molecular simulation methods are in use, which can be classified by the length and time scales they are able to probe (Figure. 1). The choice of simulation technique depends on the target properties and on the computational feasibility of the simulation method.

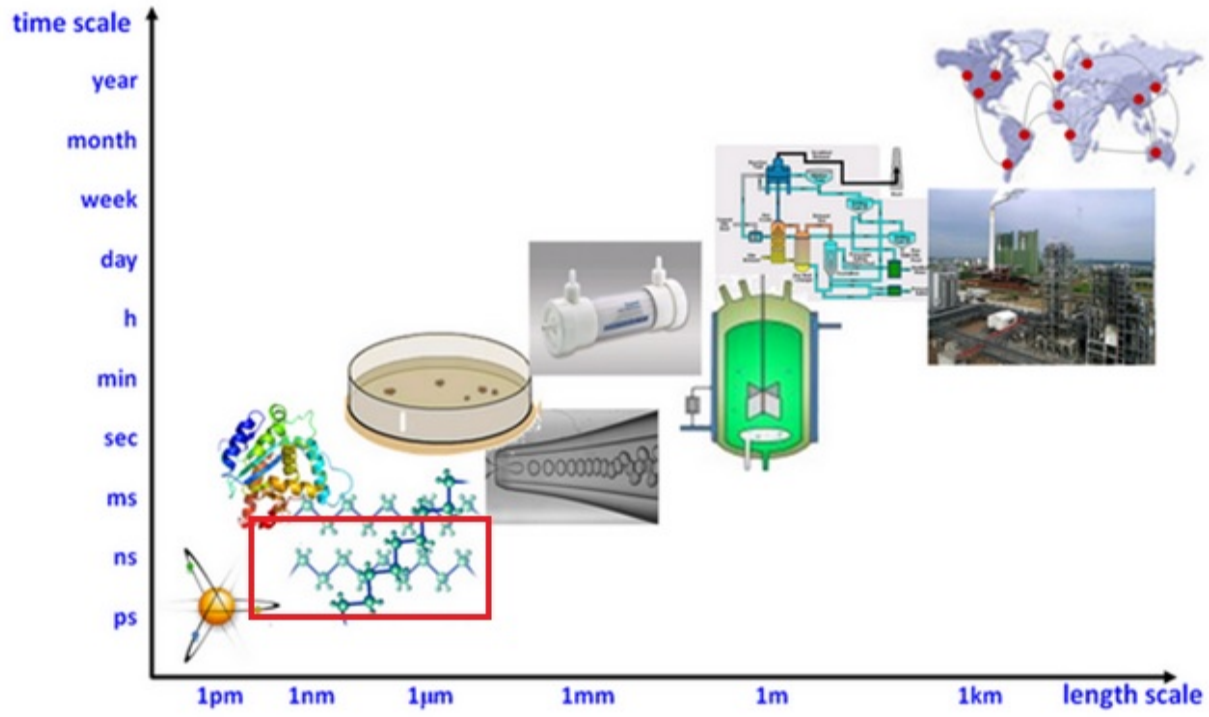


Figure. 1 Multiple time and length scales in chemical engineering. The area marked in red is our area of interest. *(Image Reference: http://you.mccormick.northwestern.edu/slide_pic/h1.png)*

In most of conventional chemical engineering, we make use of the continuum approximation to make things easier. For example, methods based on Fick's law, Navier Stokes equations, and Fourier law are used to study mass transport, momentum transport, and heat transport, respectively. However, at micro (nano)-scales, e.g. in the case of carbon nanotubes, these methods are not useful as the continuum approximation cease to be valid. Therefore, we need to employ particle simulation methods, where the particles can be atoms

for classical atomistic simulations or the nucleus and electrons in the case of quantum mechanical simulations.

To illustrate the advantage of particle simulation schemes, let us consider the example of a chemical reactions going on in a reactor. A kinetic model based on the standard reaction engineering principles will give the conversion as a function of reactor size, type, and temperature and pressure conditions, provided the experimental values of rate constants are available. On the other hand, a quantum chemistry simulation may predict the reaction rates and reaction mechanisms, with an accuracy often higher than experiments. However, these simulations are computationally feasible for very short length and time scales. In general, quantum chemistry simulations scales as $\mathcal{O}(N^4)$, which means if we double the number of particles (N), it will take 16 times more time. A useful compromise is reached by atomistic molecular dynamics methods, which scale as $\mathcal{O}(N^2)$ that can further be reduced to $\mathcal{O}(N^{1.5})$ in some implementations. It is a technique for computing the equilibrium and transport properties of a classical N-body system. By classical, we mean that we will restrict ourselves in the domain of classical physics only and we will be neglecting the quantum effects (Born-Oppenheimer assumption) for the reasons mentioned above. So we focus more on classical systems which provides us with simple systems and that too with reasonable accuracy. In these simulations we select a model system of N particles and then we solve Newton's equation of motion for this system until we reach equilibrium. After equilibrium is achieved we start our actual measurement of physical properties[1].

1.1. Motivation

The motivation for this work lies in the enormous importance of pH in various biological and biochemical processes. Like temperature and pressure, the solution pH is a property that is a driving force in various biochemical processes. For example

1. Transmembrane pH gradient is utilized by ATP synthase to synthesize cellular ATP[2] (the energy unit of a cell).
2. Multidrug efflux pump in Gram-negative bacteria undergoes conformational changes to extrude antibiotics out of the cell using transmembrane pH gradient[3].
3. The functioning of oxygen transport enzyme haemoglobin is pH-dependent.
4. Influenza M2 proton channel is activated by the low pH in the endosome to initiate viral uncoating[4].

5. The prion protein can misfold under low pH conditions to form infectious prion particles[5].
6. Many proteins denature at low pH values[6] and aggregation such as formation of amyloid fibrils in Alzheimer[7] and insulin aggregation is pH dependent[8].
7. pH is important in drug discovery. pH affects many properties of drugs like its solubility, adsorption, delivery to tissues, excretion, etc.
8. pH sensitive or pH responsive polymers are materials which will respond to the changes in the pH of the surrounding medium by varying their dimensions. Such materials increase its size (swell) or collapse depending on the pH of their environment. These materials can be used for delivering drugs to specific parts using pH as a trigger[9].

Simulations aimed at modelling pH behaviour of a molecule have some method of assigning protonation state to each ionisable residue. Since, bond breaking and formation are impossible in classical force field calculations, each residue is assigned a fixed protonation state and entire simulation is run using that. This method has two drawbacks . First, the choice of protonation state is often based on the behaviour of each ionisable residue in free solution. The protonation states have to be inferred from NMR (Nuclear Magnetic Resonance) data or PB/GB (Poisson-Boltzmann/Generalized Born Equation) calculations. This may not be true, because the environment can affect a residue's protonation state equilibrium. Second, a single protonation state may fail to represent the true ensemble of states at the desired pH. While it may seem that both drawbacks can be addressed simply by running simulations for every possible set of protonation states, this approach quickly becomes redundant. If we have N ionisable residues, there are at least 2^N distinct protonation states assuming each residue is either protonated or deprotonated. With only 10 residues this will amount to 1024 distinct simulations. While most of these states may not be found in the given ensemble, there is no way to know which ones to exclude. Changes in the pH are not taken account in the conventional MD simulations.

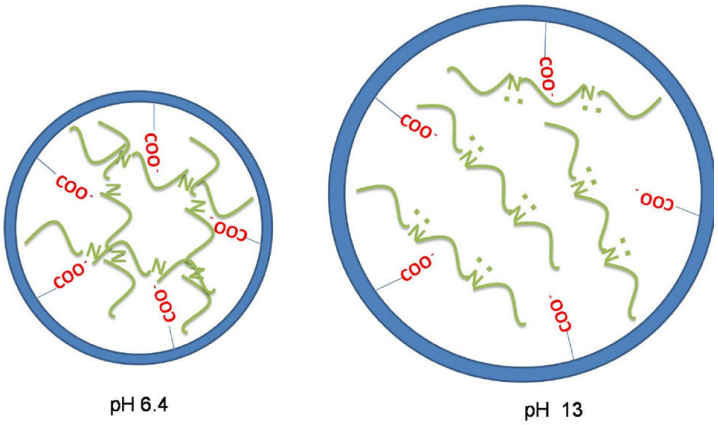
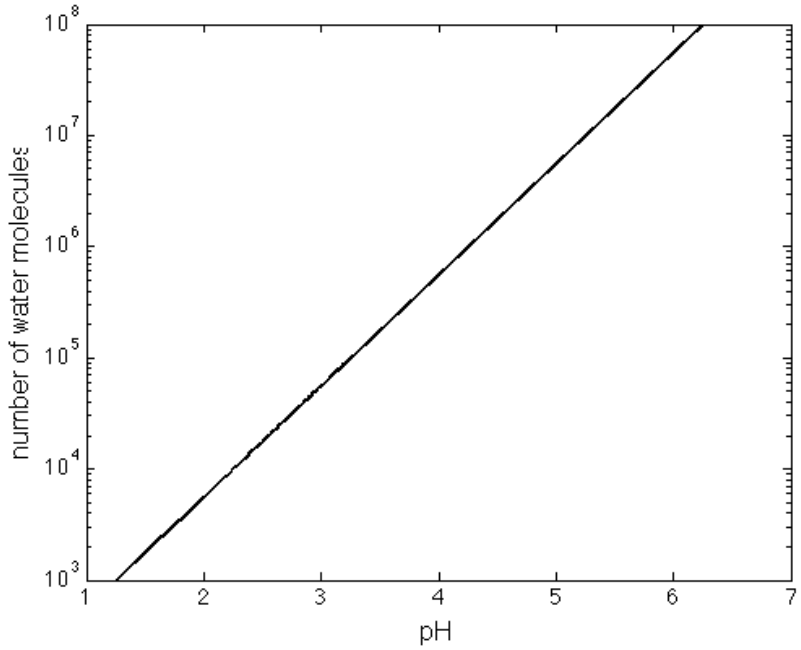


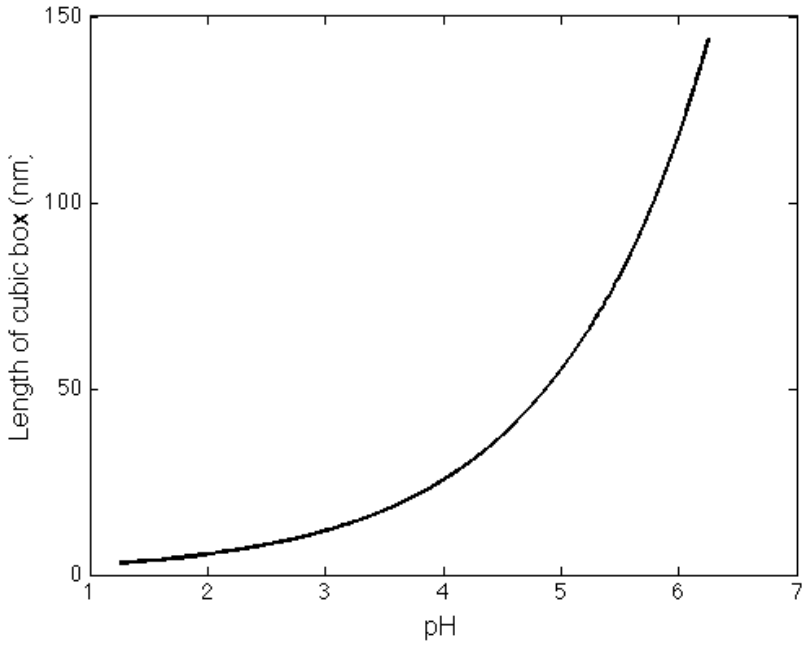
Figure. 2 Structure of a polyacrylic acid chain at different pH. At pH=6.4 it takes a more compact form closing like a ring, whereas at pH=13, it takes a linear form opening up the ring[11]. This change of shapes at different pH can prove useful for drug delivery systems.

In contrast, in a constant pH molecular dynamics simulation, the protonation state of an ionisable group of a molecule is allowed to change during the simulation according to the local electrostatic environment and the pH of the solution. The pK_a values of the ionisable groups can then be obtained from the distributions of the protonation states[12].

The most accurate way of modelling proton transfer is to describe the system in quantum mechanical term, however such calculations are computationally very expensive. A more affordable way is EVB[13,14,15] (Empirical Valence Bond) and QHOP[16] (Quantum Mechanically Derived Proton Hopping Rates) method. These methods treat only the proton transfer part as quantum mechanical system and the rest as classical. But still these approaches have a common limitation that the equilibrium state is reached at time scales that are much slower than accessible to MD simulations. Also, for these methods we need to consider abnormally long time periods and large simulation system. The concentration of pure water is roughly 55.46M (so there are 55.46 moles of water in a 1L bottle). The concentration of H^+ at pH 7 is 10^{-7}M. So for 6000 water molecules, having 1 H^+ corresponds to roughly pH of 2. Adding one more H^+ gives pH of 1.74. In other words, we cannot represent a pH of more than 2 in a box of 6000 water molecules. This amounts to a large box size which adds to computational difficulties.



(a)



(b)

Figure. 3 Effect of increasing pH on box size for 1 hydronium ion. (a) variation in number of water molecules required in box to represent the desired pH (y-axis is on logarithmic scale) (b) variation in length of cubic box(in nm) with pH

From Figure. 3, we can clearly see that to represent higher pH, we need to put in more water molecules in our box. This in turn leads to an increased box size adding to computational expense. Also, since the thermodynamic properties calculated represent an average, we need to put in more hydronium ions to get better estimates of properties. Adding more hydronium ions would mean further increasing the number of water molecules in our system.

To overcome these issues several approaches have been proposed, which can be broadly classified in two classes. Both of these classes use a titration coordinate λ whose value denotes the ionization state of a titratable group:

1. Discrete pHMD (DpHMD): These methods are generally developed by combining molecular dynamics simulation with Monte Carlo methods (or other enhanced sampling methods). In these methods, we interrupt the dynamics at certain periods given by Monte Carlo sampling during the simulations to update the protonation states[17]. At every Monte Carlo step, we compute protonation free energy of the molecule which is used as an acceptance criteria for accepting the new protonation states or not. These methods differ in the way in which this protonation free energy is evaluated.
2. Continuous pHMD (CpHMD) : In this method we introduce a λ parameter with a mass m. We assign each titratable site a fictious coordinate λ that tracks the protonation state of the molecule[18]. They are better than discrete methods in the sense that we do not need to calculate protonation free energies at every step, since these calculations are very time consuming it leads to slowing down of simulation. In Table 1 below, we give a comparison of 3 widely used CpHMD methods.

CpHMD	**Solvent Model**	**Advantages**	**Disadvantages**
Generalized Born Based[19,20,21]	GB for both conformational and protonation space	Rapid pK_a convergence	Less accurate conformational dynamics
Hybrid solvent[18]	Explicit solvent for conformational space and GB for protonation space	Rapid pK_a convergence; more accurate conformational dynamics	Mismatch of implicit and explicit solvent

Explicit solvent[22,23]	Explicit solvent for both conformational and protonation space	Accurate conformational dynamics; simulation system is charge neutral	Slow pK_a - convergence; complicated electrostatic treatment.

Table 1: Comparison of Continuous constant pH molecular dynamics methods.

An early model by Baptista et al.[24] employs MD using charges, which are averaged over protonation state distributions corresponding to selected pH, which in turn is calculated using a continuum electrostatic method. Borjesson and Huenberger[25] developed an "acidostat" method in which protonation states are coupled to a proton bath analogous to temperature bath in thermostat methods. Though pK_a calculations from this method does not give a good fit to Henderson-Hasselbalch equation. Lee et al[21] devised a new approach using λ dynamics that was originally used for calculating free energy. A potential is constructed along a coordinated λ interpolating between protonated and deprotonated states. Convergence to an intermediate charged state is avoided by using an energy barrier centred at $\lambda = \frac{1}{2}$ which forces λ to be close to 0 or 1.

1.2. Objectives

In our study we focused on CpHMD method based on λ dynamics which was introduced by Mertz and Pettitt[26] . In this approach the continuous coordinate λ was treated as an additional particle in the system which is propagated in time according to equations of motion. Protons are not explicitly transferred but instead it works like an "acidostat[25]" , the proton transfer contribution to the force acting on λ is implicitly taken into account. This proton transfer contribution depends on pH, so in this way we consider the effect of pH in our simulations. Since λ is a continuous variable, it can take fractional values also which represents unphysical states. To keep the λ value most of the time close to 0 or 1, a barrier potential[21] is used. We centre this potential around $\lambda = 0.5$ so as to reduce the sampling of values close to it. To describe protonation and deprotonation we include the effect of external pH bath on protonation and contribution to the free energy of protonation due to breakage and formation of chemical bonds.

Our initial objective was to reproduce titration curves for simple acid like hydrogen fluoride to test and validate our code. Hydrogen fluoride being a simple molecule gives us scope to see the effect of varying various parameters in our simulations as for complex molecules the time taken by these simulations is very long. Then we move on to more complex monomer molecules like acrylic acid to further validate the code for higher molecules. From there we extend our method to more complex polymer systems.

In the next chapter of this report, we discuss some basic theory about molecular dynamics and about constant pH molecular dynamics method employed by us, i.e. λ dynamics method. We also describe some of the implementation details of the methods used in our code. In chapter 3, we discuss our basic programming environment and setup. In chapter 4, we discuss the simulation results. In last chapter, we present a conclusion to our work.

2. THEORY

2.1. Conventional Molecular Dynamics

In molecular dynamics (MD), we solve Newton's equation of motion for a system which consists of N atoms which interact each other through different forces acting on them:

$$m_i \frac{\partial^2 \boldsymbol{r}_i}{\partial t^2} = \boldsymbol{F}_i \qquad i = 1 \ldots N. \tag{2.1}$$

$$\boldsymbol{F}_i = -\frac{\partial V}{\partial r_i} \tag{2.2}$$

The forces $\boldsymbol{F}_i$ are obtained from the potential function $V(\boldsymbol{r}_1, \boldsymbol{r}_2 \ldots \boldsymbol{r}_N)$

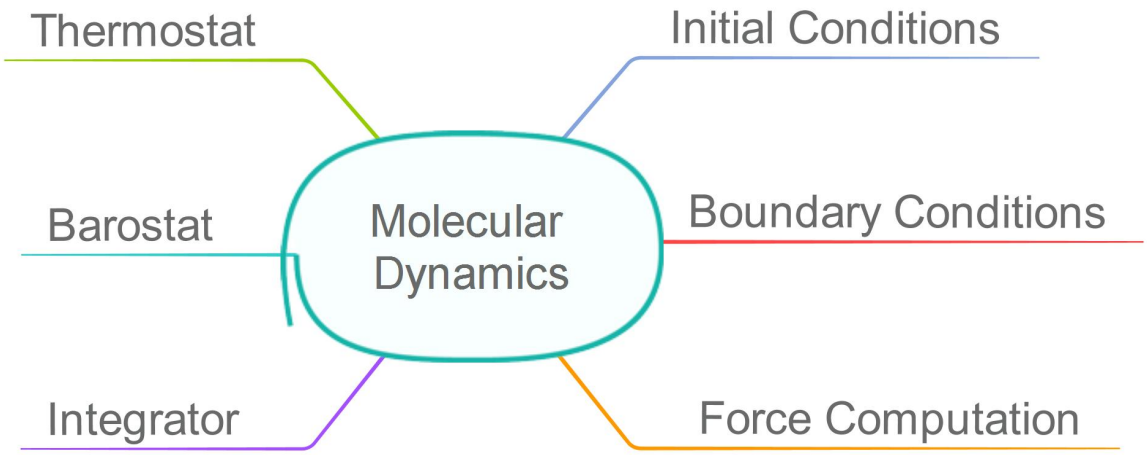


Figure. 4 Basic ingredients of a conventional molecular dynamics simulation.

These equations are solved simultaneously by using various numerical techniques in very small time steps. The system is first equilibrated and then actual measurements are taken. The coordinates $\boldsymbol{r}_i$ obtained as a solution gives us the trajectory of the atoms. Many microscopic properties can be calculated from this data about the atom trajectory.

There are certain approximations which we have to make while doing MD which are listed below:

1. We use classical physics (not quantum): Use of Newton's equation implies that. Though it is all right for most of the system at normal temperature but there are certain exceptions which we have to care about.

2. Born-Oppenheimer Approximation: We assume that when atoms change position electrons adjust instantaneously. For this reason, we use forces in our simulation which depends only on position of the atoms so the electronic motion and the nuclear motion is not considered. It is true most of the time though we cannot treat chemical reactions this way.
3. The Forces are approximate: The expressions used for calculating forces need not be exact. Approximations are made to save computational time and memory. We also use certain cut off while calculating long range forces.
4. Boundary conditions: In MD, we solve for system of about few thousand particles (due to computing constraints) so we need to avoid finite size artefacts. For this, we use periodic boundary conditions to mimic the bulk system. This may also introduce certain error for small systems.

2.1.1. Global MD Algorithm

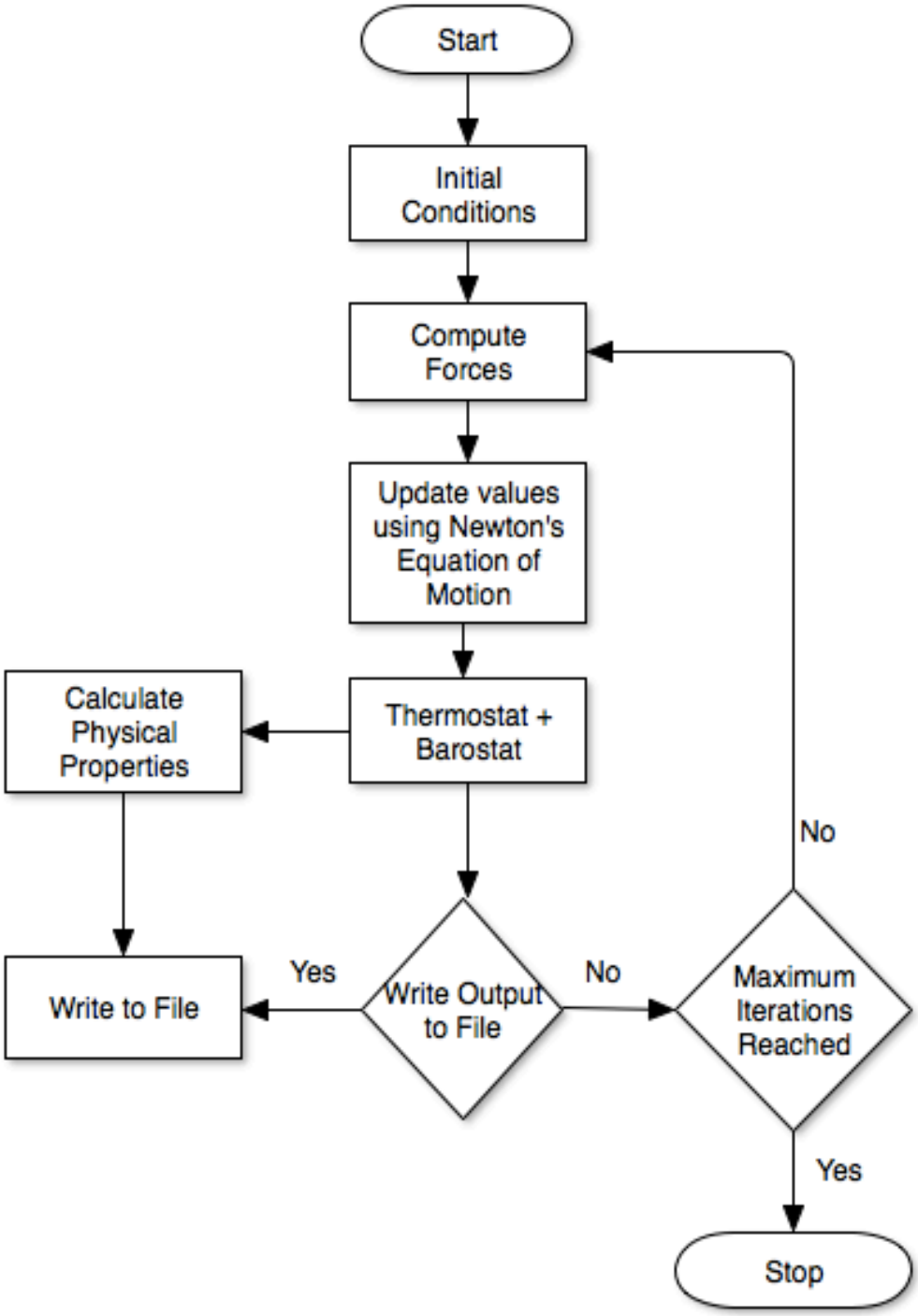


Figure. 5 Flowchart for conventional molecular dynamics

1. Initial Conditions:
    a. Potential V
    b. Position $\boldsymbol{r}$
    c. Velocity $\boldsymbol{v}$

Iterate 2,3,4 for required number of time steps

2. Compute Force (Interaction Potential)
    a. $\boldsymbol{F}_i = -\frac{\partial V}{\partial r_i}$ by using

$$\boldsymbol{F}_i = \sum_j \boldsymbol{F}_{ij} + bonded\ interactions + constraint\ force$$

3. Update values by using Newton's Equation of Motion
4. If output is required to be written to screen/file
    a. Write coordinate, velocity, energy, temperature, etc

2.2. λ Dynamics

In this approach an additional degree of freedom, λ is introduced into the equation of motion with mass m, coordinate λ and velocity $\dot{\lambda}$ which is used to track the changes in the protonation state of a molecule. Its value is updated at every time step along with position and velocity.

Any weak acid can be represented by the following reaction

$$HA \leftrightarrow H^+ + A^- \quad (2.3)$$

We introduce a term λ which quantifies the degree of protonation of our acid where $\lambda = 0$ corresponds to fully protonated acid(or group, if it is a part of a bigger molecule) and $\lambda = 1$ to fully deprotonated acid. If the pH is constant and we know pK_a then we can write concentration of various species in this reaction in terms of λ

$$[H^+] = 10^{-pH}, [A^-] = \lambda^o C_A, \text{and } [HA] = (1-\lambda^o)C_A \quad (2.4)$$

where λ^o is the equilibrium value of λ and C_A is the concentration of acid.

$$\frac{[A^-][H^+]}{[HA]} = \frac{\lambda^o 10^{-pH}}{1-\lambda^o} = 10^{-pK_a} \quad (2.5)$$

We can write the last equation as

$$\lambda^o = \frac{1}{1+10^{pK_a - pH}} \quad (2.6)$$

or

$$pH = pK_a + \log\frac{\lambda^o}{1-\lambda^o} \quad (2.7)$$

Now the Hamiltonian for the system is expressed as[27]

$$H(\lambda) = (1-\lambda)H_0 + \lambda H_1 + H_{Env} + \frac{m}{2}\dot{\lambda}^2 + U^*(\lambda) \quad (2.8)$$

The force acting on λ is

$$F_\lambda = -\frac{\partial V(\lambda)}{\partial \lambda} \quad (2.9)$$

where $V(\lambda)$ is the potential energy part of the Hamiltonian

$$V(\lambda) = (1-\lambda)V_0 + \lambda V_1 + V_{Env} + U^*(\lambda) \quad (2.10)$$

Here H_0 and V_0 indicates the values for $\lambda = 0$, that is the protonated state and H_1 and V_1 represents values for $\lambda = 1$, that is the deprotonated state.

Now since only $\lambda = 0\ and\ \lambda = 1$ have physical significance, we want the value to be close to 0 or 1 most of the time. For this we impose some constraints on λ. Moreover, we need that following things always holds:[12]

1. λ should be restricted between 0 and 1.
2. Values of λ be close to 0 or 1 for as much as possible.
3. Transition between $\lambda = 0$ to $\lambda = 1$ should be fast.
4. Residence time at a particular state should be long so that sampling can be done easily.
5. Frequency of transitions should be controllable.

For meeting condition 1 and 2 a projection of an angular coordinate system on λ space is proposed[27].

$$\lambda = r\cos\theta + \frac{1}{2} \quad (2.11)$$

Now the force that will act on this θ particle is

$$F_\theta = r\sin\theta\,\frac{\partial V(\lambda(\theta))}{\partial \lambda} \quad (2.12)$$

$$r = \frac{1}{2} + \sigma \quad (2.13)$$

where σ is a fluctuation parameter whose value is to be determined by testing different values.

For meeting condition 3 we use a parabolic biasing potential of the following form[21]

$$U^*(\lambda) = 4h\lambda(1-\lambda) \quad (2.14)$$

Also selecting a good biasing potential also helps us to achieve constraints 1 and 2.
The conditions 4 and 5 can be met by adjusting the height h of this biasing potential. The barrier height is an important parameter, it can be used to trade between protonation state transition rate and fraction of simulation time spent at intermediate λ values(more in results chapter).

Now, most of the forces are defined by our chosen force fields except two:

i. Effect of external pH bath on protonation
ii. Effect due to breakage and formation of bonds

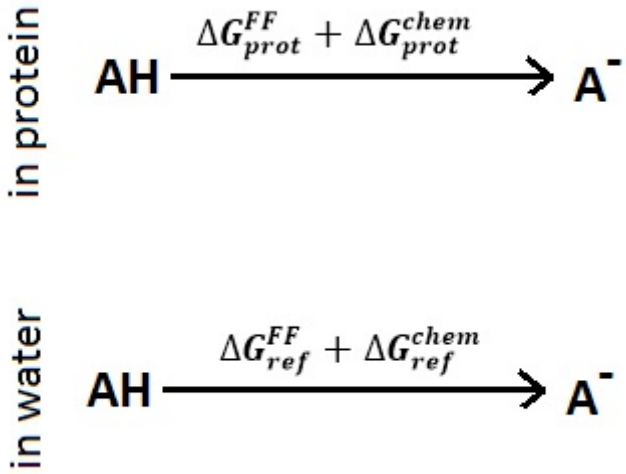


Figure. 6 Equilibria between protonated (AH) and deprotonated (A^-) forms of a titratable site in a molecule in the reference state of water.

For calculating these terms, we will use an additional term ΔG^{chem} . To determine this term, we will consider equilibrium between a protonated (AH) and a deprotonated acid (A^-) in solvated protein and in water. The equilibrium in water will be considered as a reference state because for this the deprotonation free energy is available. The free energy of these two reactions is then split in two terms

i. ΔG^{FF} – obtained due to force field calculation
ii. ΔG^{chem} – contribution (i) and (ii) as mentioned above

Here, we make an assumption that, the ΔG^{chem} in the two states won't differ significantly because of our choice of reference state[28,29]

$$\Delta G_{prot}^{chem} \approx \Delta G_{ref}^{chem} = (\ln 10)RT\left(pK_{a,ref} - pH\right) - \Delta G_{ref}^{FF} \quad (2.15)$$

Where $pK_{a,ref}$ is the measured pK_a of the reference titratable site in our reference state. The pH term describes the pH dependency of the equilibria thus accounting for the missing proton term in the equation.

The last term will be obtained by a reference free energy simulation which will be performed using conventional MD before starting our main constant pH simulations.

$$\Delta G_{ref}^{FF} = G_{ref}^{FF}(\lambda = 1) - G_{ref}^{FF}(\lambda = 0) = \int_{\lambda=0}^{\lambda=1} \langle \frac{\partial H_{ref}(\lambda)}{\partial \lambda} \rangle_{\lambda} \, d\lambda \quad (2.16)$$

where $H_{ref}(\lambda)$ is the Hamiltonian of the reference system
we use the following potential to implement the desired free energy difference in our λ dynamics calculations:

$$V^{chem}(\lambda) = \lambda(\ln 10)RT\left(pK_{a,ref} - pH\right) - \Delta\hat{G}_{ref}^{FF}(\lambda) \quad (2.17)$$

with $\Delta\hat{G}_{ref}^{FF}(\lambda)$ as a curve fit to $G_{ref}^{FF}(\lambda)$.

2.3. Implementation

2.3.1. Initial Condition

1. Select the box shape and its dimensions
2. Select the number of atoms/molecules to be put in the box
3. Input the initial coordinates of the molecules in the box.
4. Input initial velocities of the molecules which are most commonly generated from the Maxwell-Boltzmann distribution at that temperature.

$$p(\boldsymbol{v}_i) = \sqrt{\frac{m_i}{2\pi kT}} \exp\left(\frac{m_i \boldsymbol{v}_i^2}{2kT}\right) \quad (2.18)$$

5. The center of mass velocity is normally set to zero so that there is no drifting of molecules.

2.3.2. Compute Forces

This is the most time-consuming part of all MD simulations. If we consider pairwise additive forces like the Lennard-Jones Force then for any i particle we have to consider the

effect of all the rest $N-1$ molecules on it. If we are not using any cut-off range then it amounts to $N(N-1)/2$ pair interactions. So it scales as $O(N*N)$. So we use a way to reduce this effort to $O(N)$ which is using Cell lists.

The idea behind them is that our simulation box is divided into equal cells with size equal to cut-off radius. Now each particle in a given cell interacts with particles present only in that cell or the neighbouring cells. The total number of cells are independent of number of molecules in the box.

This method can be incorporated readily into our program by using linked lists.

2.3.3. Velocity Verlet Integrator

The velocity verlet algorithm is used for solving the equation of motion. It can be easily derived by the Taylor Series expansion of the coordinate vector. In velocity verlet, position vector $\boldsymbol{r}(t)$ and velocity $\boldsymbol{v}(t)$ at time t are used to integrate the equation of motion.

$$\boldsymbol{r}(t+\Delta t) = \boldsymbol{r}(t) + \boldsymbol{v}\Delta t + \frac{\Delta t^2}{2m}\boldsymbol{F}(t)$$
$$\boldsymbol{v}(t+\Delta t) = \boldsymbol{v}(t) + \frac{\Delta t}{2m}[\boldsymbol{F}(t) + \boldsymbol{F}(t+\Delta t)] \quad (2.19)$$

1. Calculate $\boldsymbol{r}(t+\Delta t)$ from equation 1.
2. Derive $\boldsymbol{F}(t)$ from the interaction potential using $\boldsymbol{r}(t+\Delta t)$.
3. Calculate $\boldsymbol{v}(t+\Delta t)$ from equation 2.

2.3.4. Periodic Boundary Conditions

It is the most common way to avoid edge effects in MD simulations. The atoms of the system are assumed to be in a space filling box which is surrounded by translated copies of itself.

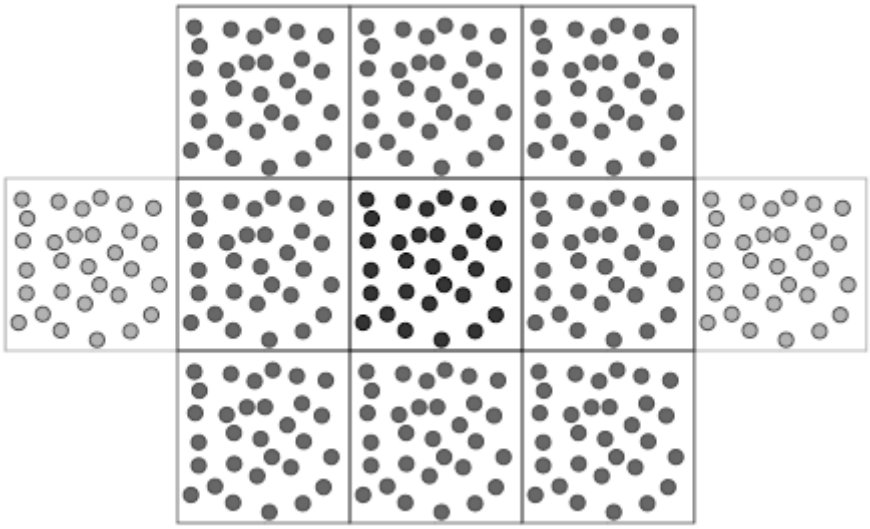


Figure. 7 The meaning of periodic boundary conditions (here, two dimensional case is shown for simplicity)

There are two effects of this:

1. An atom that leaves the box from left reappears from the right edge.
2. Wrap-around effect – an atom near a boundary interacts with the atom that is near the opposite boundary.

2.3.5. Andersen Thermostat

In Andersen Thermostat[30] the system is coupled to a heat bath set at the desired temperature by stochastic forces. These forces act at selected particles at random time intervals. This can be done in two ways:

1. Randomizing all the velocities simultaneously every $1/\nu\Delta t$ steps.
2. Randomizing every particle with some small probability every time step equal to $1/\nu\Delta t$.

where Δt is the time step for the simulation and ν is the frequency of stochastic collisions which denotes the coupling strength.

In our code it is implemented in the latter manner.

1. Start with initial set of positions $\boldsymbol{r}(t)$ and velocities $\boldsymbol{v}(t)$.
2. Integrate the equation of motion using Verlet Integrator to get the new position and

velocities.

3. Select particles that will undergo collision with heat bath with a probability $\nu\Delta t$.
4. If a particle is selected to undergo collision with heat bath then draw its new velocity from the Maxwell-Boltzmann distribution corresponding to that temperature.
5. All other particles remain unaffected.

2.3.6. Berendsen Thermostat

It is a weak coupling[31] with the molecular dynamics. It corrects the deviation from a set temperature T_0 slowly according to

$$\frac{dT}{dt} = \frac{T_0 - T}{\tau} \qquad (2.20)$$

The temperature decays exponentially with a time constant. It is mostly used for equilibration runs not for production runs as it does not resemble NVT ensemble.

2.4. Force Field

A force field describes how a molecule interacts with other molecules and also with itself (atom interacting with another atom in a molecule). There are various force fields which have been developed and which works for different molecules. A force field, in general is of the form

$$F = 2\ body\ interaction + 3\ body\ interaction + 4\ body\ \ldots\ldots \qquad (2.21)$$

A simple force field is of the form

$$\begin{aligned} U = & \sum_{bonds} k_{bond}(d - d_o)^2 \\ & + \sum_{angles} k_{angle}(\theta - \theta_o)^2 \\ & + \sum_{torsions} k_{torsions}[1 + \cos(n\chi - \delta)] \\ & + \sum_{ij} \left\{ 4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right) + \frac{q_i q_j}{r_{ij}} \right\} \end{aligned} \qquad (2.22)$$

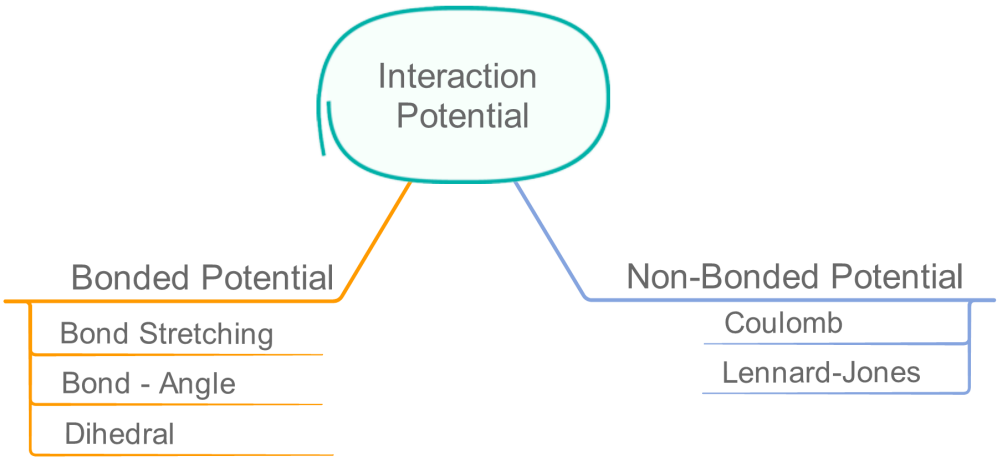


Figure. 8 Components of a typical Force Field/Interaction Potential

The various components of force field are described below:

2.4.1. Non-Bonded Interaction

These basically consist of repulsion and dispersion term which are combined together in the Lennard-Jones potential and an electrostatic interaction

Lennard-Jones Potential

Lennard-Jones potential V_{LJ} between any pair of atoms is given by

$$V_{LJ}(\boldsymbol{r}_{ij}) = \frac{C_{ij}^{(12)}}{r_{ij}^{12}} - \frac{C_{ij}^{(6)}}{r_{ij}^{6}} \qquad (2.23)$$

where the parameters $C_{ij}^{(12)}$ and $C_{ij}^{(6)}$ depends on pair of atom types.

For convenience, it is written in the form:

$$V_{LJ}(\boldsymbol{r}_{ij}) = 4\epsilon_{ij}\left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right) \qquad (2.24)$$

If i and j are two different type of atom then σ_{ij} and ϵ_{ij} can be calculated using the Lorentz-Berthelot rules

$$\sigma_{ij} = \frac{1}{2}(\sigma_{ii} + \sigma_{jj}) \qquad (2.25)$$

$$\epsilon_{ij} = (\epsilon_{ii}\epsilon_{jj})^{1/2} \quad (2.26)$$

The expression for force is given by

$$F_i(\boldsymbol{r}_{ij}) = \left(12\frac{C_{ij}^{(12)}}{r_{ij}^{13}} - 6\frac{C_{ij}^{(6)}}{r_{ij}^{7}}\right)\frac{\boldsymbol{r}_{ij}}{r_{ij}} \quad (2.27)$$

Coulomb Interaction

The coulomb interaction is given by:

$$V_e(\boldsymbol{r}_{ij}) = \frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{\epsilon_r r_{ij}} \quad (2.28)$$

and the corresponding force is given by

$$F_i(\boldsymbol{r}_{ij}) = \frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{\epsilon_r r_{ij}^2}\frac{\boldsymbol{r}_{ij}}{r_{ij}} \quad (2.29)$$

2.4.2. Bonded Interactions

Bond Stretching

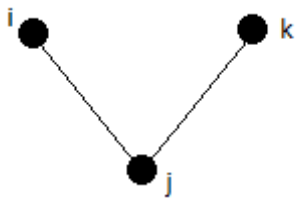
There are various models that are used for bond-stretching potential but we will be using the simplest of them for the time being. The bond stretching is represented by a harmonic function between any pair of atom

$$V_b(\boldsymbol{r}_{ij}) = \frac{1}{2}k_{ij}^b(r_{ij} - b_{ij})^2 \quad (2.30)$$

and the corresponding force is

$$F_i(\boldsymbol{r}_{ij}) = \frac{1}{2}k_{ij}^b(r_{ij} - b_{ij})\frac{\boldsymbol{r}_{ij}}{r_{ij}} \quad (2.31)$$

Bond-angle Stretching



The bond angle vibration between any triplet of atoms is given by

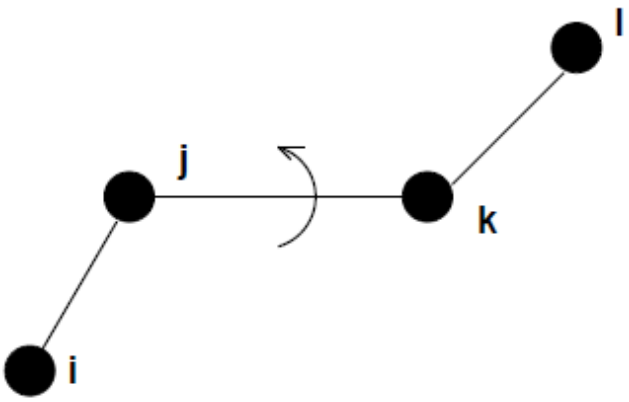
$$V_a(\boldsymbol{\theta}_{ijk}) = \frac{1}{2}k_{ijk}^{\theta}(\theta_{ijk} - \theta_{ijk}^0)^2 \quad (2.32)$$

The expression for force can be obtained by differentiation of this potential

$$F_i = -\frac{\mathrm{d}V_a(\theta_{ijk})}{d\boldsymbol{r}_i}$$
$$F_k = -\frac{dV_a(\theta_{ijk})}{d\boldsymbol{r}_k} \qquad (2.33)$$
$$F_j = -F_i - F_k$$

Where $\theta_{ijk} = \cos^{-1}\frac{(\boldsymbol{r}_{ij}.\boldsymbol{r}_{kj})}{r_{ij}.r_{kj}}$

Dihedral Potential



It is given by following function

$$V_d(\phi_{ijkl}) = k_\phi(1 + \cos(n\phi - \phi_s)) \qquad (2.34)$$

where ϕ is the angle between ijk and jkl planes.

3. PROGRAMMING ENVIRONMENT

We have written a C/C++ library to implement the above method from scratch. A brief summary of our code is provided in Appendix B. Presently our library doesn't support parallelization capabilities. It takes in an input file which contains various parameters related to the simulation conditions. It also requires 5 files, which contains various details about our molecule and system

3.1. Protein Database (PDB) File

It contains details about the initial position of the atoms in our simulation. It doesn't contain any information about which atom is bonded to which.

3.2. Protein Structure File (PSF)

It contains detail about all the links between the atoms. It specifies all the bonds, angles, dihedrals and impropers. We need two such files one each for two states i.e. one for protonated state of our molecule and other one for deprotonated.

3.3. Parameters File (PARAMS)

It contains the entire force field details about our system. One each for two states i.e. one for protonated state and one for deprotonated state. It contains detail about bonded interactions as well as the non-bonded interactions about our system.

3.4. Lambda groups file

It is used to specify the titratable part of our system. It requires name of residue and the atom numbers that are part of our titratable molecule and lambda dynamics will be performed only on those residues. It also contains a parameter initial_lambda which is used to assign the initial state of the titratable residue.

3.5. pK_a file

It contains information about each of the titratable residue that is defined in our lambda groups file. More specifically it requires the value of the barrier potential and value of 6 parameters of which 4 are found by running reference free energy simulations and rest two

are calculated based on the reference pH and temperature of our system as given by $equation\ 4.15$.

More details about the format of lambda groups, pK_a and input files are given in Appendix C.

3.6. Reference Free Energy Simulation

To calculate the deprotonation energy of our system we need to run a free energy simulation. For this we have used GROMACS 5.0.2, which is an open-source molecular dynamics package. For free energy calculations we require two things in the first place

1. Two end states i.e. protonated and deprotonated for our case
2. The pathway connected the two states

The pathway is specified by using a keyword fep-lambdas which is a vector of the lambda values which connects state 1 to state 2, for eg. [0 0.25 0.5 0.75 1]. GROMACS currently supports two methods for calculating free energy, slow growth method and Thermodynamic Integration method. In our work we have used the later method as it is proved to give more accurate results. More details about this method has been given in Appendix A.

3.7. Constant pH Simulations

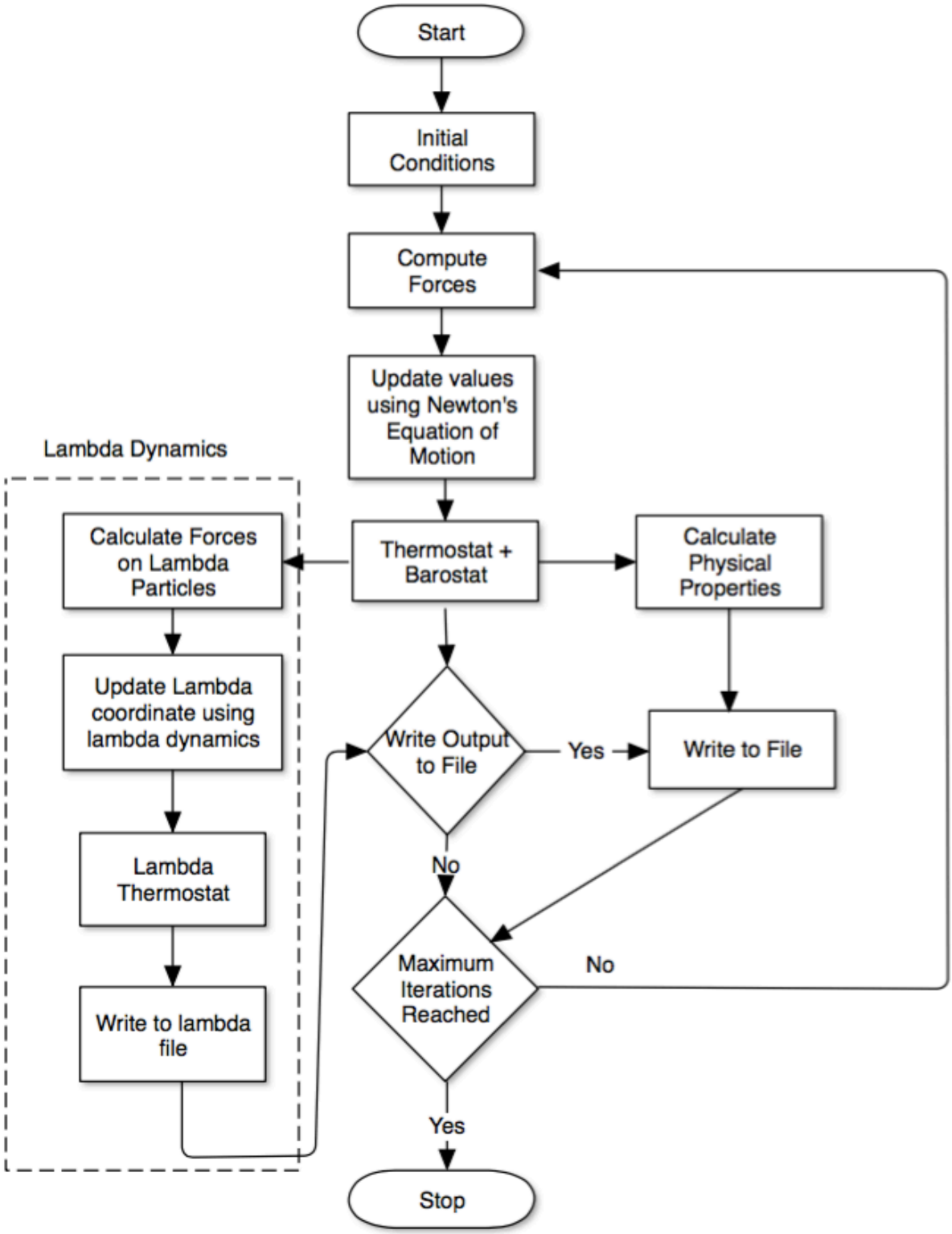
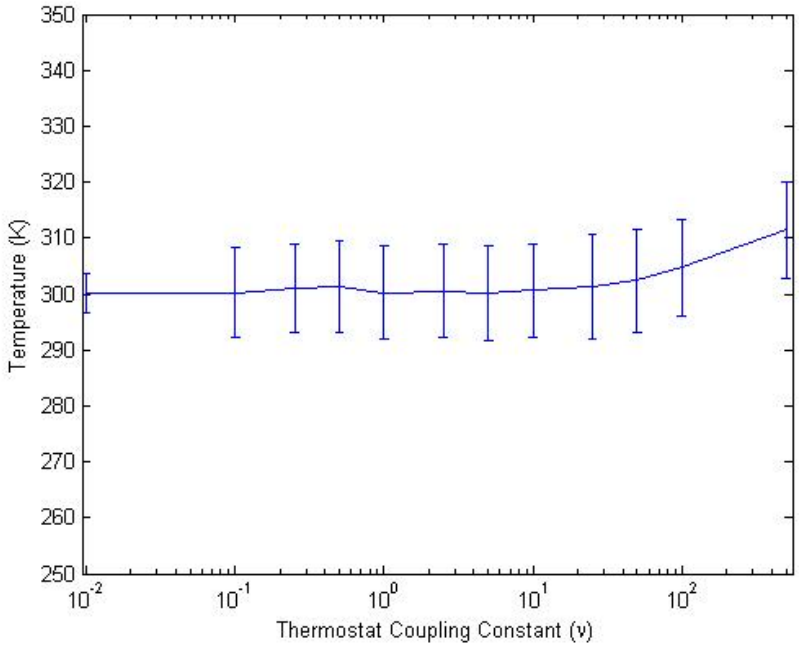


Figure. 9 Flowchart for constant pH molecular dynamics based on λ dynamics

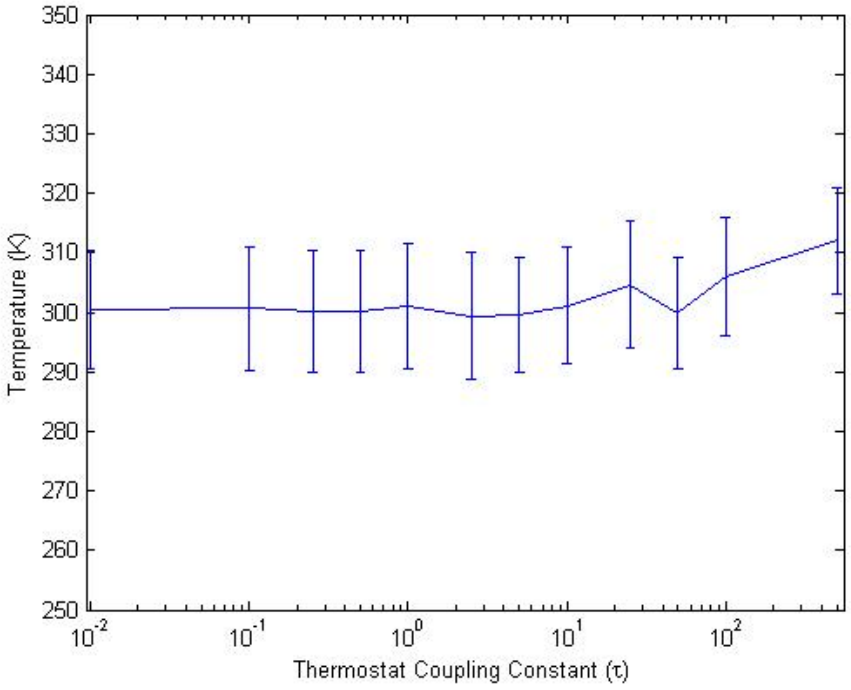
We have used a Verlet integrator for integrating the equations of motion with a timestep of 1fs. Cut-off schemes are used for Non-Bonded interactions in the system with the cut-off being set at half the box length.

The temperature of the λ thermostat is kept constant by using an external heat bath. For this purpose, we will be using two thermostats:

i. Berendsen Thermostat for the real particles because it provides us with stricter temperature coupling as compared to Anderson Thermostat
ii. Berendsen Thermostat for λ. We had some problems with our Anderson coupling scheme so we preferred Berendsen here.



(a)



(b)

Figure. 10 Mean Temperature along with standard deviation for different values of coupling constants. (a) Anderson Thermostat (b) Berendsen Thermostat. (x axis is on logarithmic scale)

The mass of lambda particles is set at 20u as it gives most stable trajectories[12].

3.8.Simulations on Pure Water

We ran some simulations on water to check the validity of our code. We used 216 molecules of SPC (Single Point Charge) water with a density of 0.996 g/cc in a cubic box with periodic boundary conditions. From the data generated we calculated number of hydrogen bonds formed per molecule of water molecule which came close to be 4 which is equal to the experimentally observed results. For calculating number of hydrogen bonds a geometrical criterion was used.

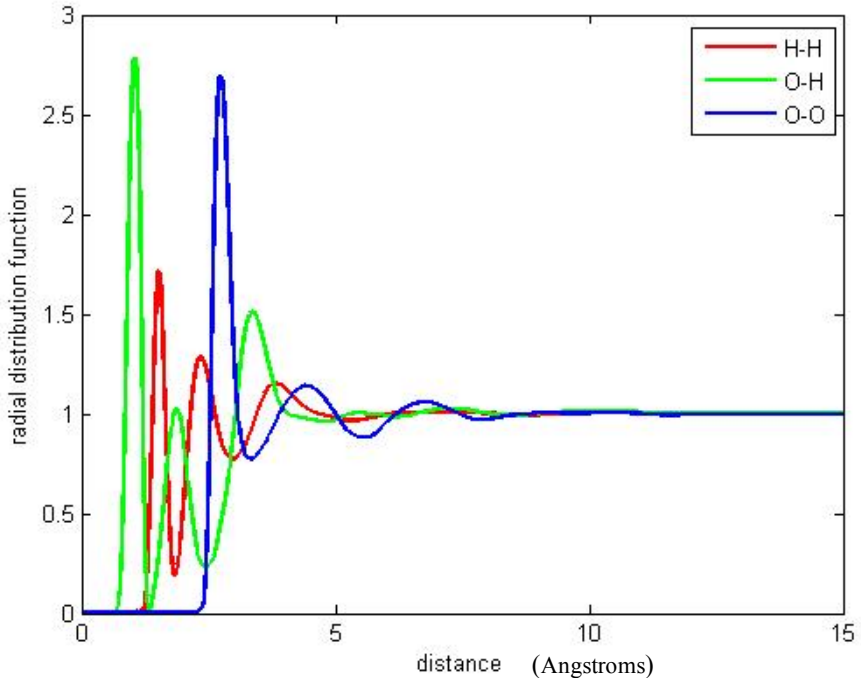


Figure. 11 Radial Distribution Function for water

The peaks at various distance in this RDF plot shows the probability of finding two atoms at that distance. One can see that after some distance all the plots approaches 1 which indicates that there is no long range order which is true for water. At short distances RDF is zero which is due to strong repulsive forces.

We have also made a patch on GROMACS 3.3 which is an open source library for doing molecular dynamics which presently lacks doing constant pH simulations. The rationale behind this patch was to use some advanced and efficient methods of doing conventional molecular dynamics in our simulations which are already present in GROMACS but would have taken time to incorporate in our library. For simple molecule HF we have used our custom library and for simulations on more complex acrylic acid we have used our patch on GROMACS.

4. RESULTS AND DISCUSSIONS

As a test case, so as to verify that our code is working correctly, we chose hydrogen fluoride as the initial molecule to work with. This choice was made since running simulations directly on a big molecule like polymer would take very long time. Therefore, before moving on to bigger molecules we wanted to verify that our code is working well. Since it's a simple molecule and small system, it makes finding errors and removing them easier. Also it allows us to do quantify the effect of various parameters in our code.

4.1. Simulation of Hydrogen Fluoride

Reference Free Energy Simulations

As mentioned earlier, reference free energy simulations were performed in GROMACS 5.0.2. The simulation box consisted of one hydrogen fluoride molecule solvated with 278 molecules of SPC water in 2.1nm box.

Unlike in *Donini et. al*[12] and other papers where they have used a cubic fit to $dH/d\lambda$ plot, we are using a cubic spline fit in our code as in our case cubic fit fails to incorporate the effect of free energy fully specially near $\lambda = 0$ which cubic spline captures more perfectly.

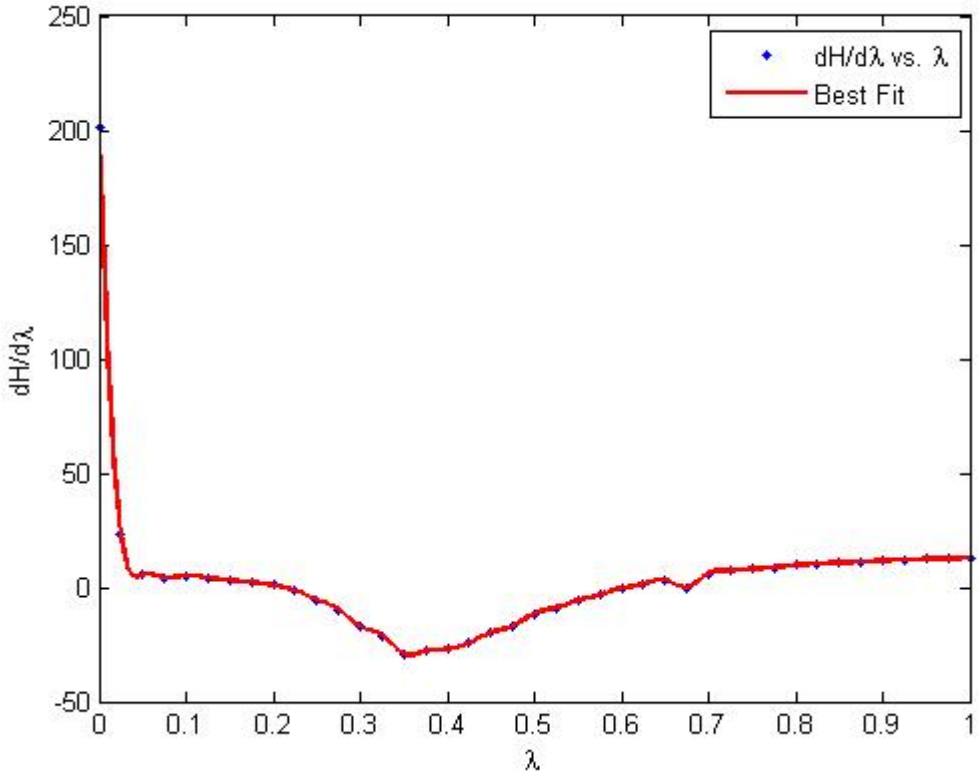


Figure. 12 Plot of $dH/d\lambda$ vs λ

Constant pH simulations

The exact same system and simulation conditions were used in these simulations as used in reference free energy simulation. Any change in system conditions in these two cases can affect the results adversely.

Effect of Barrier Potential

Firstly, we analysed the effect of barrier potential on our system. Adjusting barrier potential is essential to ensure that sufficient sampling is achieved in our simulation so that statistical errors can be minimised. For this we ran various simulation with different barrier heights at $pH = pK_a$ for a simulation time of 400ps.

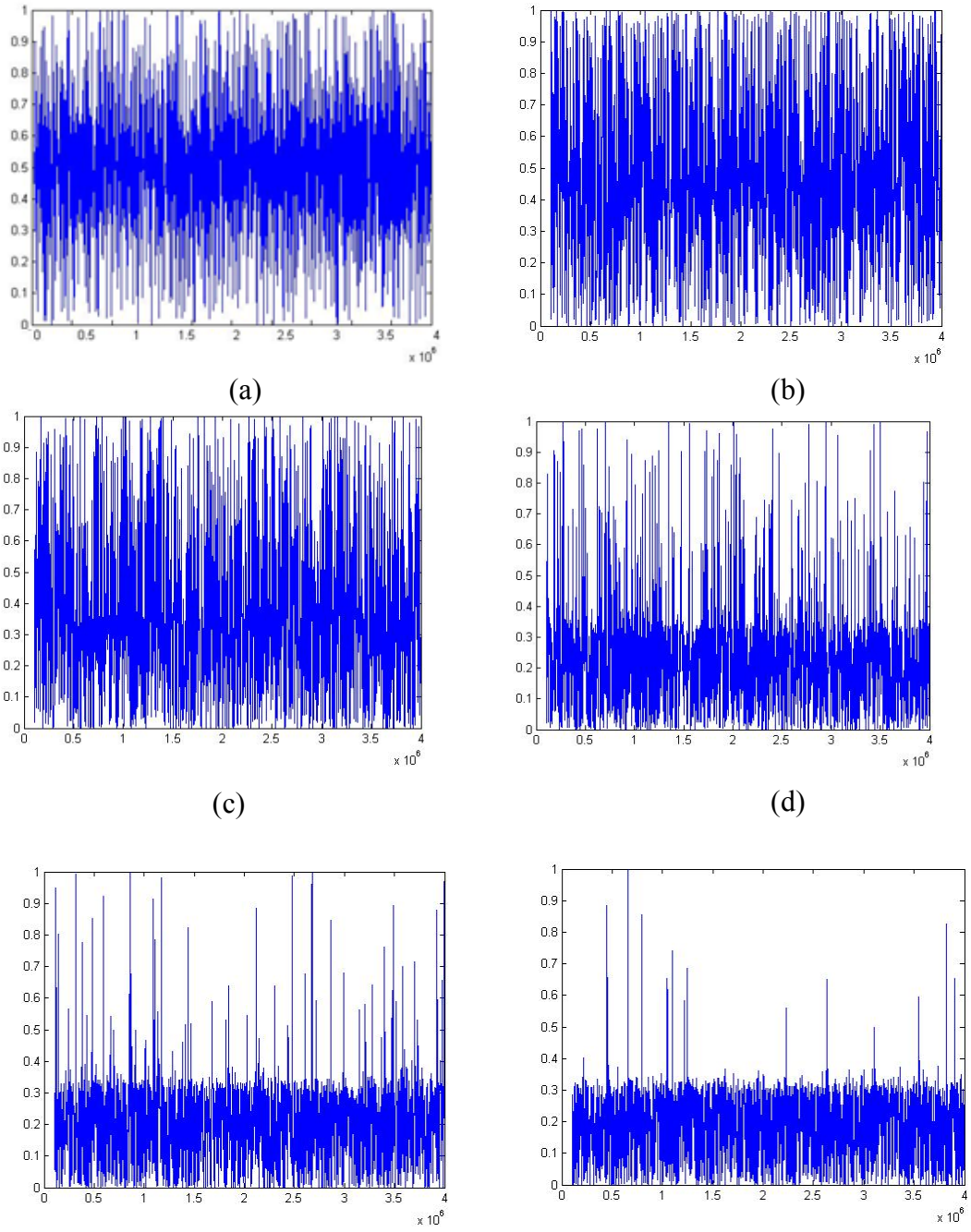
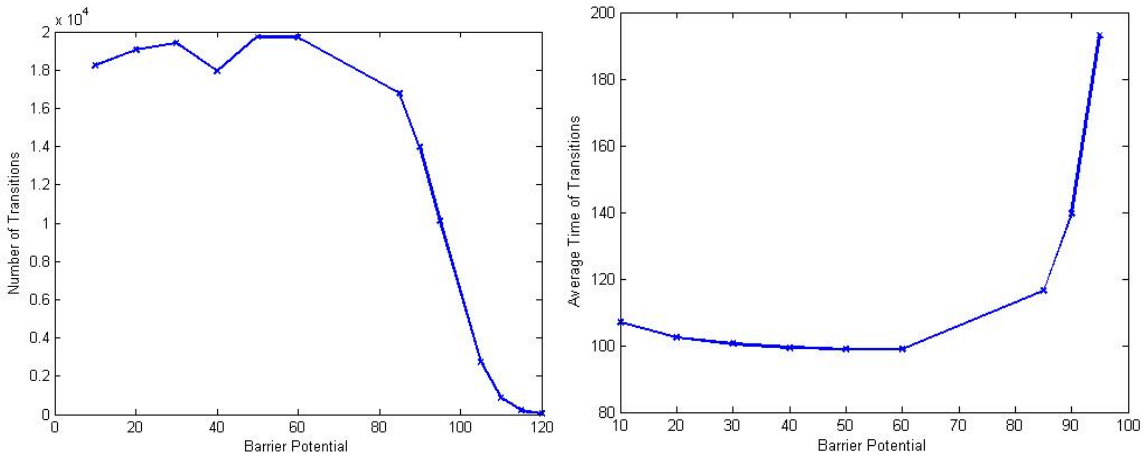


Figure. 13 Effect of barrier potential on λ variable (a)- 60, (b)-85,(c)-95,(d)-105,(e)-110, (f)-115 (all units in kcal/mole)

With an increase in barrier potential one thing is evident that number of transition decreases and we have to run longer to get sufficient number of transitions. Also, the residence time increase with increasing barrier potential. So we have two opposing things at work and we have to choose a middle way, as we want high residence time but also significant number of transitions



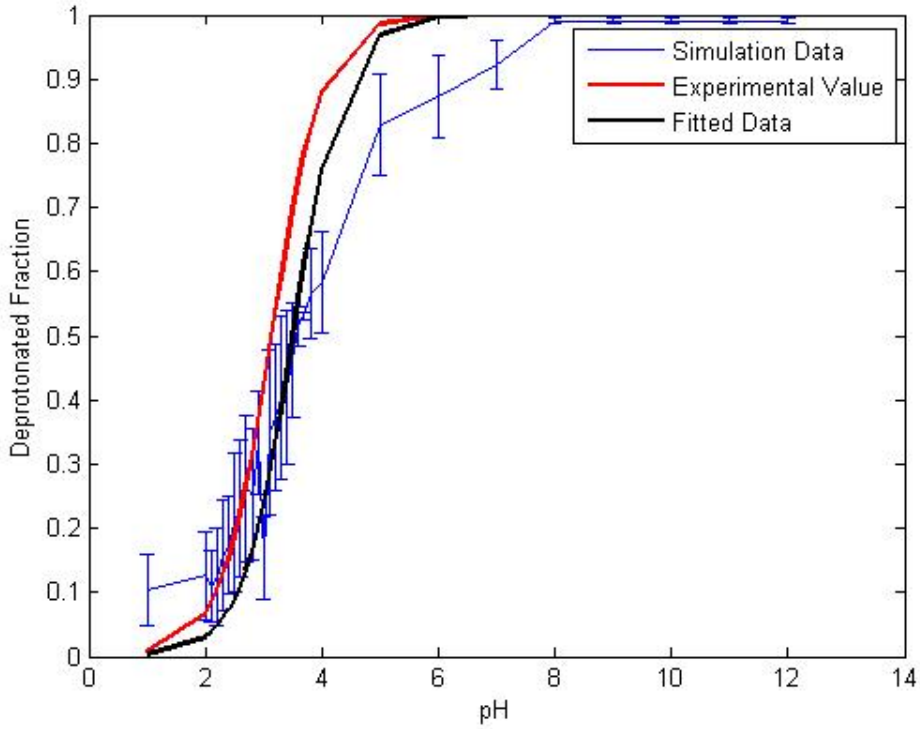
(a) (b)

Figure. 14 Effect of barrier potential on (a) number of transitions and on the (b) residence time of lambda states/average time between transitions.

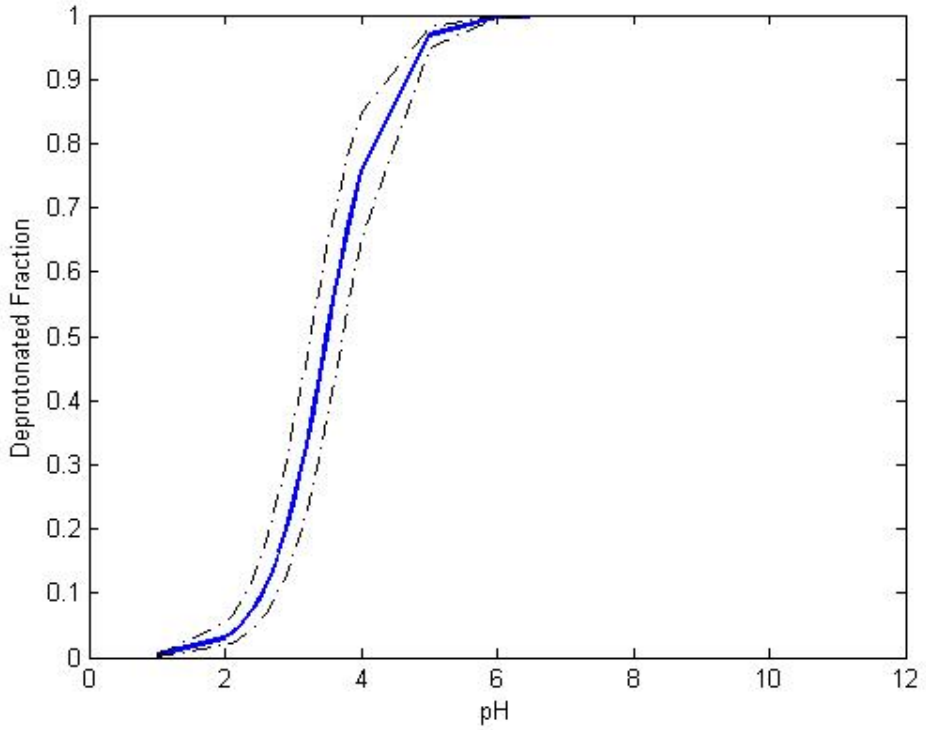
pK_a calculation

Our first concern here was to reproduce the shape of titration curve for HF and calculate its pK_a. For calculating pK_a of compounds we performed constant pH simulations at different pH just like titration. Initially we performed 12 simulations at pH range from 1-12 at every integer value. Then we also performed simulations between pH 2 and 4 at interval of 0.1. In all we ran simulations for 29 pH values. Each simulation ran for 15ns of production run after equilibrating the system for 1ns. From those simulations we calculated the fraction of the deprotonated acid which was calculated from the values of our λ parameter. For our purpose we chose $\lambda < 0.1$ as protonated and $\lambda > 0.9$ as deprotonated. This deprotonated acid fraction was then fitted to Henderson-Hasselbalch equation to find out the theoretical pK_a.

$$f^{deprot} = \frac{1}{1 + 10^{(pK_a - pH)}} \quad (6.135)$$



(a)



(b)

Figure. 15 Henderson-Hasselbalch curve for HF (a) shows fitted data, simulation data and the actual curve for HF, (b) shows the fitted curve along with the error margins in dashed lines.

For the chemists, we note that in a titration experiment, the pH is usually measured as a function of the volume of a strong acid/base solution added to the analyte solution. In contrast, in these constant pH simulations, pH is a fixed parameter, whereas the equivalents of analyte (i.e. how much of the analyte reacts with one mole of hydrogen ions) is the quantity to be estimated. Therefore, the titration curves in Figure. 15 are to be read as inverted titration curves, with respect to a typical experimental titration curve.

The calculated pK_a was equal to 3.5027 ± 0.2241 as against the experimentally obtained value of 3.14. The calculated pK_a deviates considerably from the experimentally determined value. This is possibly due to following reasons:

1. First and foremost, we have made significant assumptions in chosing the force field for HF, as it's standard force field is not available.
2. There might have been statistical errors:
    a. The value of λ is still for most of the simulation time in the intermediate range i.e. $0.1 < \lambda < 0.9$ which could lead to sampling errors as these states are physically meaningless
    b. The above point also implies that we need to run our simulations longer to ensure sufficient sampling and avoid these errors.

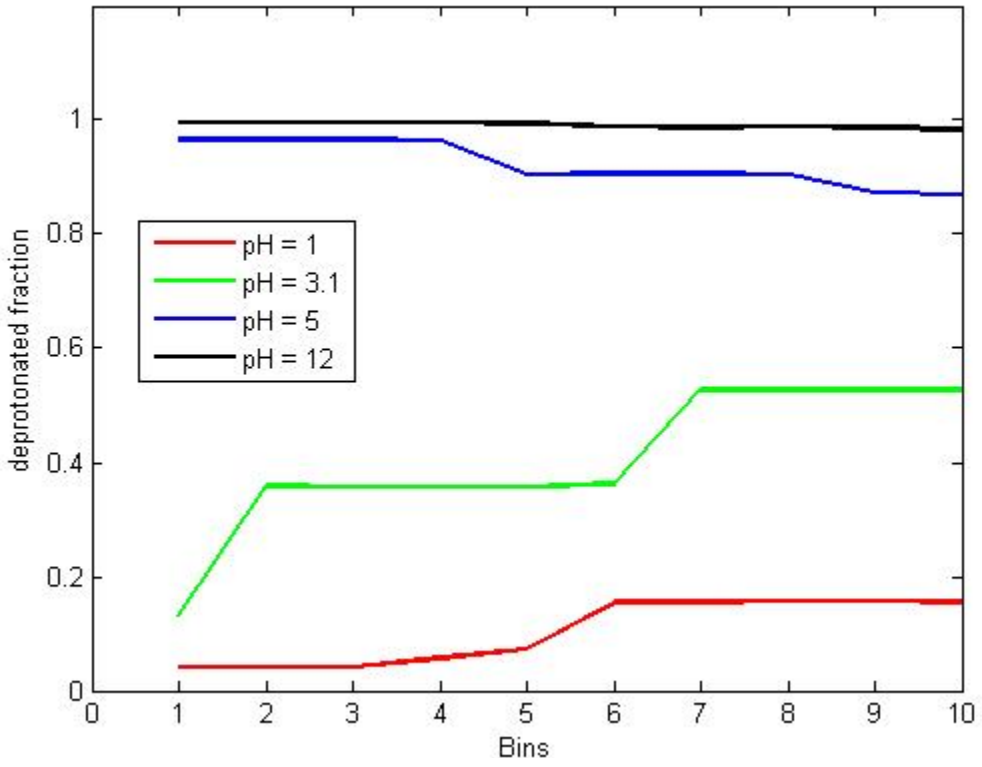


Figure. 16 Convergence of deprotonation for different pH values.

The above plot shows the convergence issue in our results. The entire simulation data was divided into 10bins of 1.5ns each and then deprotonated fraction was calculated for each bin separately. This fraction was then plotted against the bin number. For some values like pH=3 (represented by green line) we see that it doesn't converge that well which induces a sampling error in our results. For others like pH=5 and pH=12 (represented by blue and black lines respectively), we see that it converges fairly well.

Nevertheless, it still provides us with a strong foundation to build our future work.

4.2. Simulation of Acrylic Acid

For acrylic acid we have used a patch on Gromacs 3.3. The patch includes the method detailed in Chapter 2 for constant pH and uses the original gromacs features for conventional MD. The force field for acrylic acid was obtained from SwissParams.

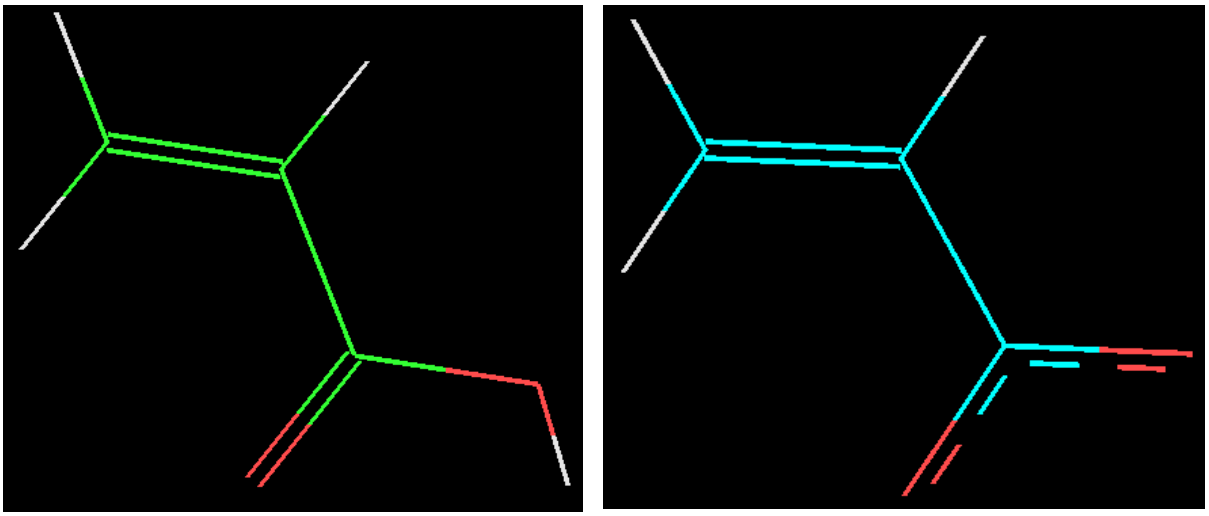


Figure. 17 Acrylic acid molecule protonated(on left) and deprotonated(on right).

Reference Free Energy Simulation

As for hydrogen flouride, reference free energy simulations were done using GROMACS 5.0.2 using 278 molecules of SPC water with 1 molecule of acrylic acid in a 2.1nm size cubic box.

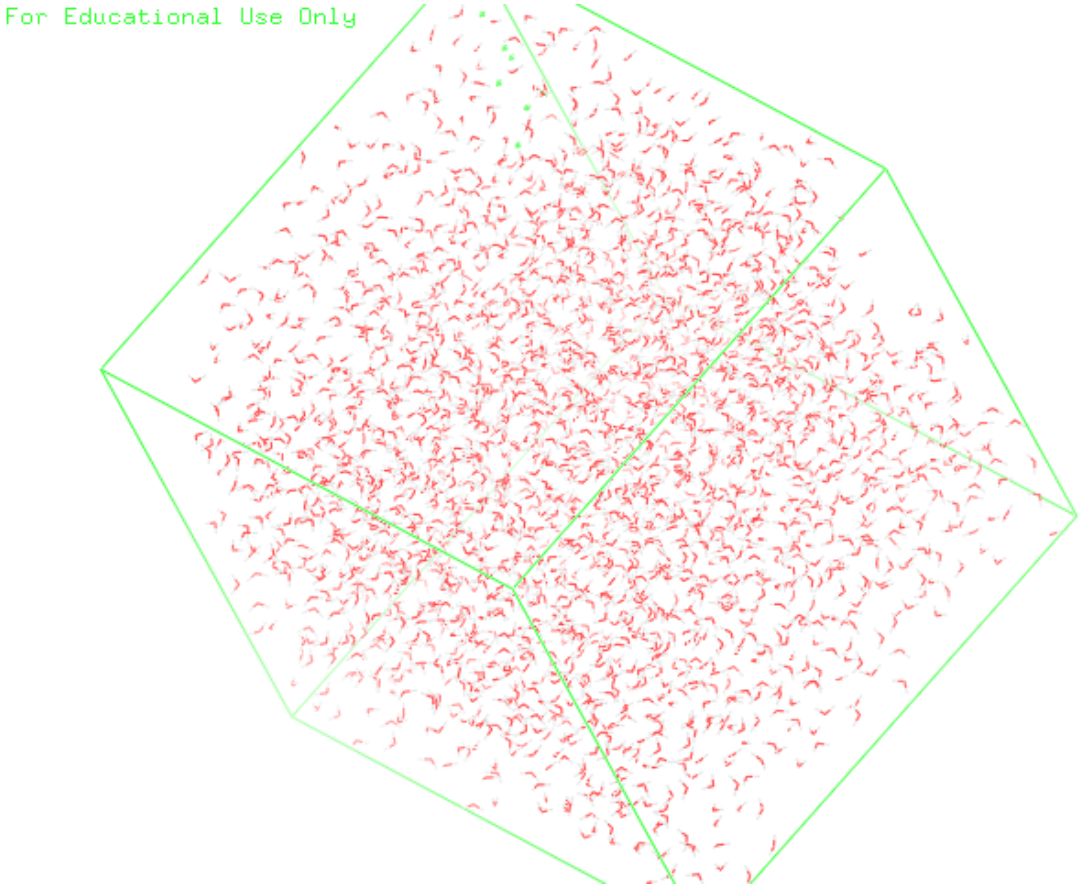


Figure. 18 Simulation box consisting of water molecule and acrylic acid at the centre(not visible in the image)

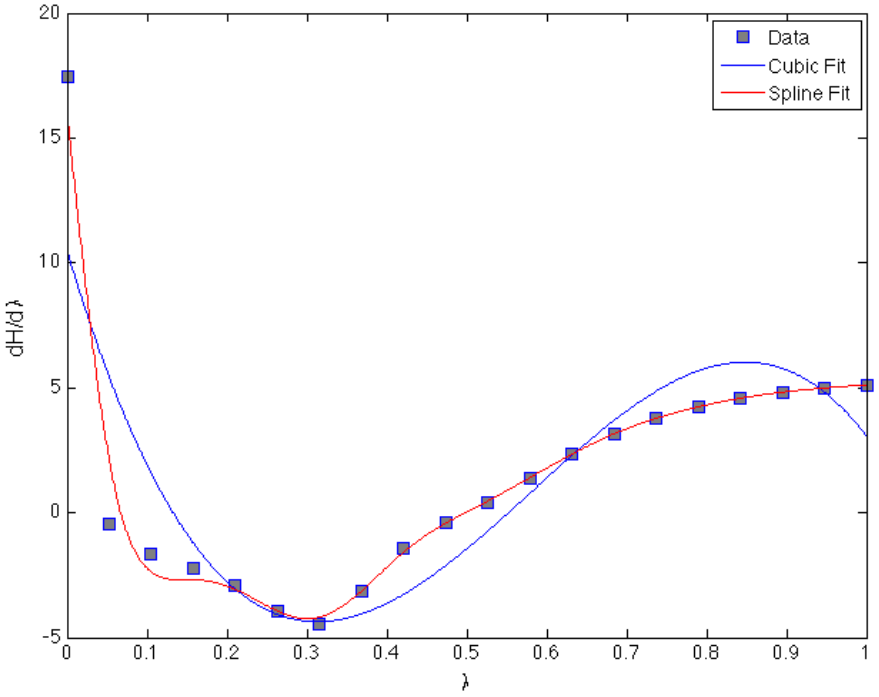


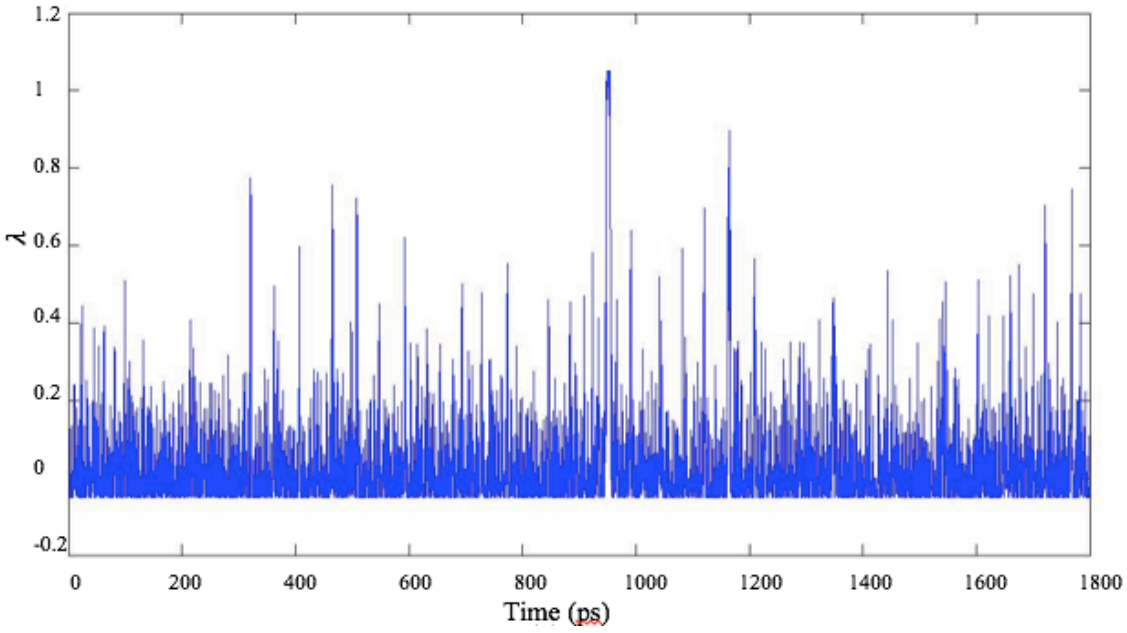
Figure. 19 Plot of $dH/d\lambda$ vs λ

The values obtained from reference energy simulations were fitted to a spline curve whose coefficients were used in our constant pH simulation.

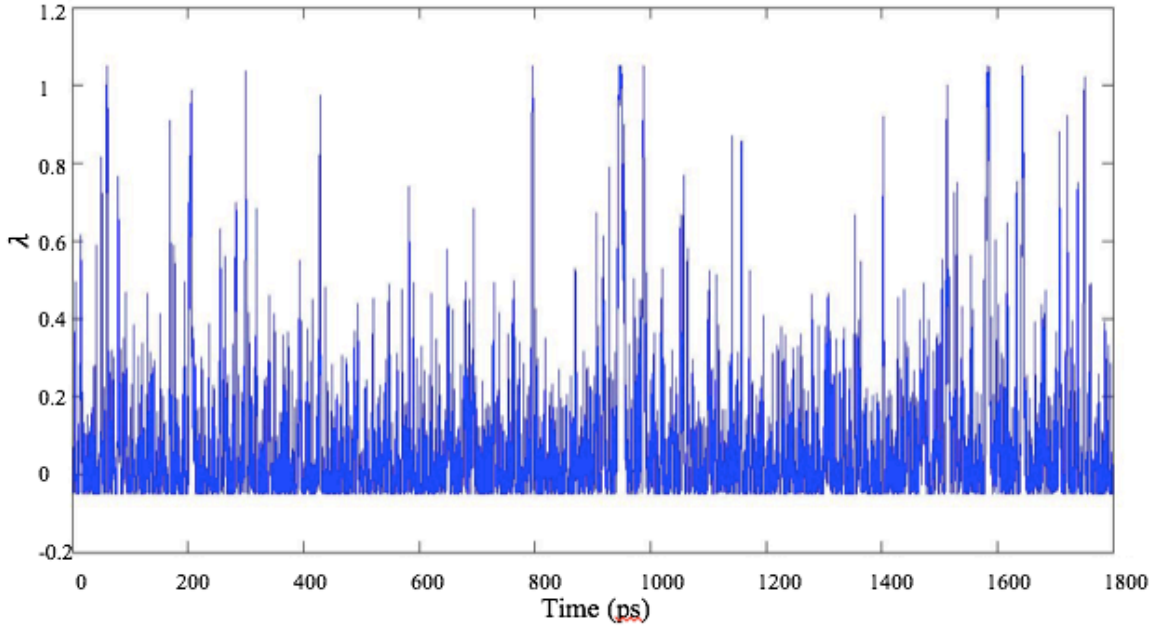
Constant pH Simulations

These simulations were done using patch on GROMACS 3.3 using the same box size as above and the same parameters in mdp file. Simulations were carried out for 8 pH values between 2 and 6(details in table below). Each simulation was repeated 3 times so as to get estimates of error. Unlike in the above case where we have used Berendsen thermostat for lambda particle, here we are using Anderson thermostat which gives us more stable trajectories.

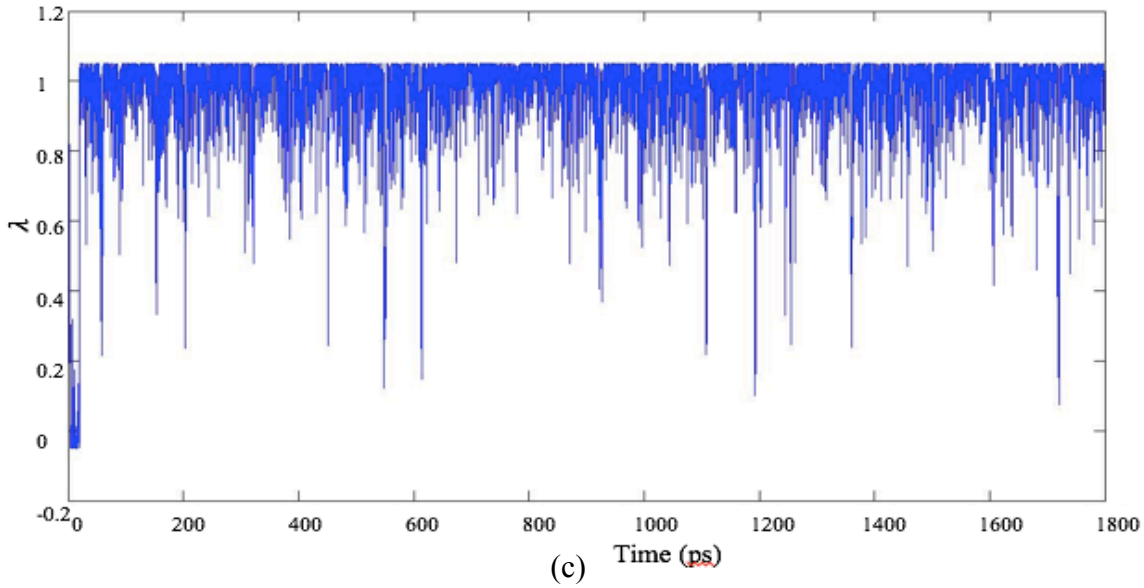
pH	#protonated states $\lambda < 0.1$	#deprotonated states $\lambda > 0.9$	Total states	$\frac{deprotonated}{total}$
2	14913	87	15000	.0058
3	14376	837	15213	0.055018734
4	9252	4982	14234	0.350007025
4.25	6756	7030	13786	0.509937618
4.5	5326	9890	15216	0.649973712
4.75	3343	11189	14532	0.769955959
5	1819	11167	12986	0.859926074
6	287	14040	14327	0.979967893



(a)



(b)



(c)

Figure. 20 Plot of λ vs time for constant pH simulations at (a) pH = 2, (b) pH=3 and (c)pH= 6

In Figure. 20 we can see the plot of λ vs time. For pH=2 and 3, it clearly shows that the value of λ stays below 0.1 for most of the time which implies acrylic acid remains protonated for most of the simulation. But for pH=6 it stays above 0.9 implying deprotonated acid for most of the simulation run. This is as expected in real experiments.

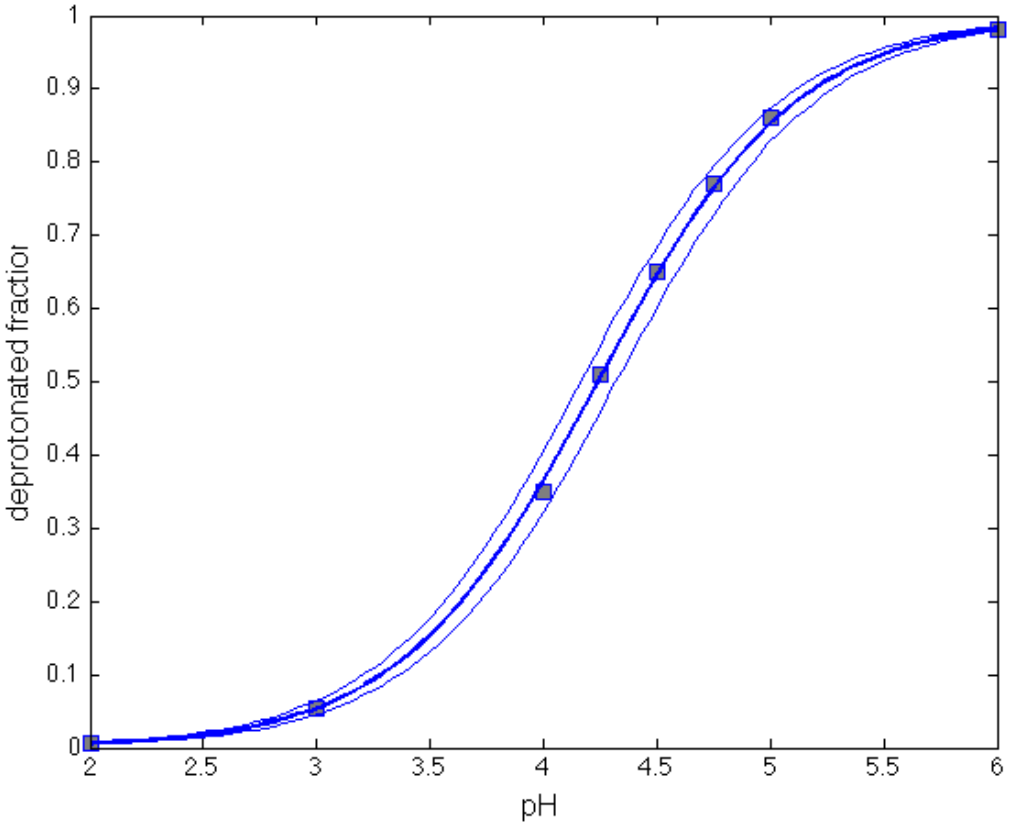


Figure. 21 Henderson-Hasselbalch curve for Acrylic Acid with error margins

The calculated pK_a was equal to 4.1490 ± 0.0789 as against the experimentally obtained value of 4.25.

In the case of acrylic acid we get a more accurate measure of the pK_a as compared to earlier case of hydrogen fluoride. This is because

i. We have a more accurate force field in this case
ii. We are using GROMACS which helps us in faster convergence and gives more accurate results
iii. We can see in Figure. 22 that in this case the value of deprotonation fraction is converging more readily as compared to the earlier case. Though still for pH=5 and pH=6 we see that the value of deprotonation factor haven't converged yet as it is still rising.

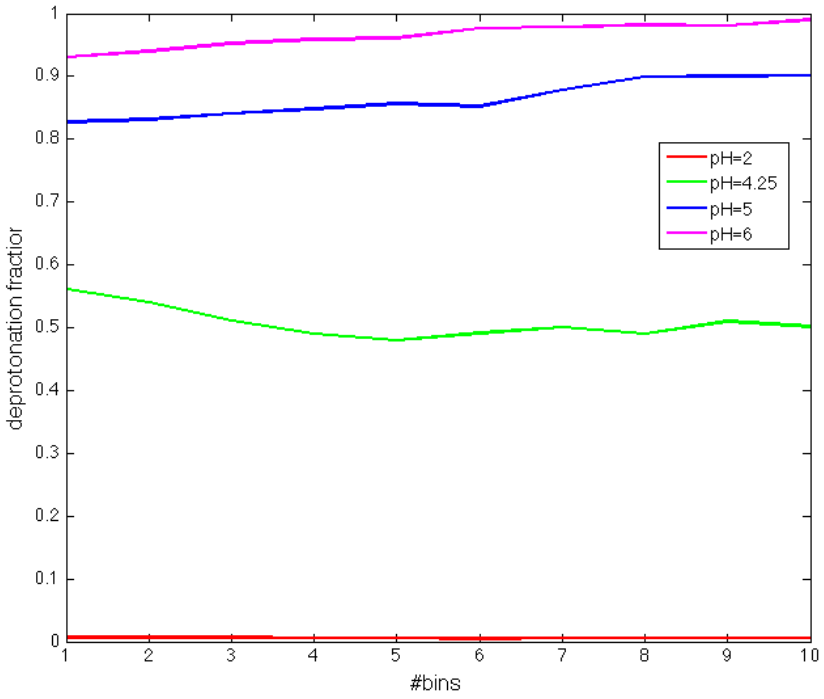


Figure. 22 Convergence of deprotonation fraction for various pH values

4.3. Extension to higher systems

Using this method we tried to simulate a 5-monomer polyacrylic acid molecule using GROMACS patch, but until now we have not got any significant results for that. One of the major bottleneck in these simulations is requirment of computational power and memory.

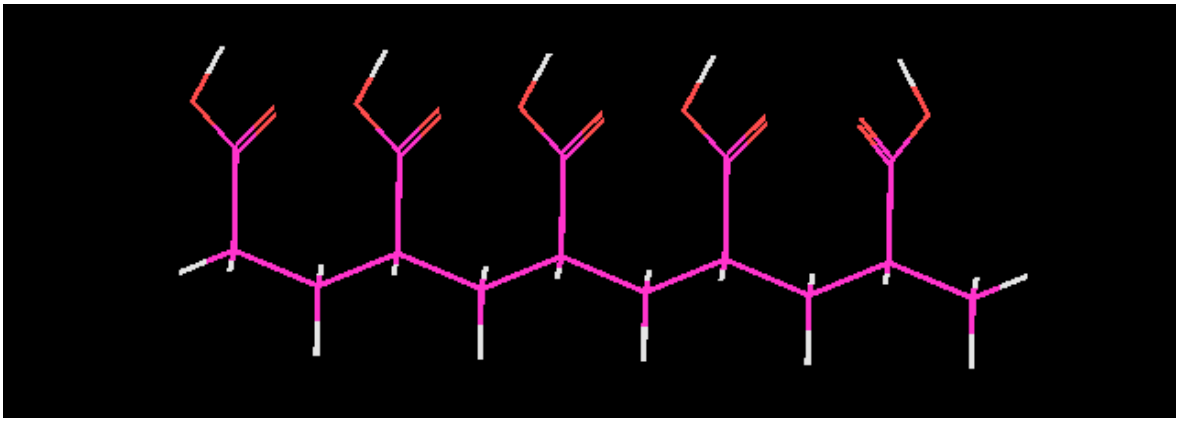


Figure. 23 5 monomer polyacrylic acid in iso form.

When moving from an acrylic acid monomer to polyacrylic acid polymer the length of the cubic box doubles atleast. This is because that the minimum box size required should be sufficient to cover the contour length of the polymer and also the molecule should not interact with it's own image. Also, for larger molecules a single solvent molecule should

not be able to see both sides of the polymer. This means that the length of each box vector must exceed the length of macromolecule in the direction of that edge plus two times the cut-off radius. This amounts to nearly 8 times more water molecules in our box so as to maintain the density of water and to completely solvate our molecule. Since these simulation scales as $O(N^{1.5})$, this leads to simulation requiring 23 times more time to complete. To put this into perspective, with the computational hardware at our disposal, it takes around 65 hours to run 1ns of simulation. If we run it for 20ns it amounts to 1300hours of runtime which equals 54 days roughly for 1 simulation. Also, since polymers have complex structure they require longer equilibriation runs to relax the molecules which further adds to computational expense. With polymers having more than one titratable site, we cannot say at the moment how this method will work as there may be issues of charged sites repelling each other which may lead to unstable system. There are methods which consider the cooperative effects of one titratable site on the other[12] which are more accurate for multiple sites as compared to single site independent model as ours. Presently our code doesn't have this method. Nonetheless, this study provides a strong foundation and a detailed parametric study to move on to polymers with one titratable site.

5. CONCLUSION

Present work portrays the λ dynamics method for doing constant pH molecular dynamics in explicit solvent. We have written our code to implement the above algorithm. The code is validated first for conventional MD by running simulations over pure water. Two systems have been analysed in detail: (i) hydrogen fluoride and (ii) acrylic acid.

1. With an increase in barrier potential the number of transition decreases and we have to run longer to get sufficient number of transitions.
2. We also observed that the residence time increases with increasing barrier potential.
3. So there is a trade off between residence time and number of transitions, as we want high residence time but also significant number of transitions.
4. Barrier potential is an important parameter as it affects the sampling significantly.
5. We were able to reproduce the shape of titration curve for a typical acid, though the calculated pK_a varied considerably from the experimental value but that can be attributed to approximations in our chosen force field and also to the problems of convergence as shown in Figure 16.
6. Spline fit to reference free energy gave us more accurate value of pK_a as compared to the cubic fit.
7. We were able to reproduce the titration curve for acrylic acid and the calculated pK_a was within the experimental limits.

Appendix A: Thermodynamic Integration

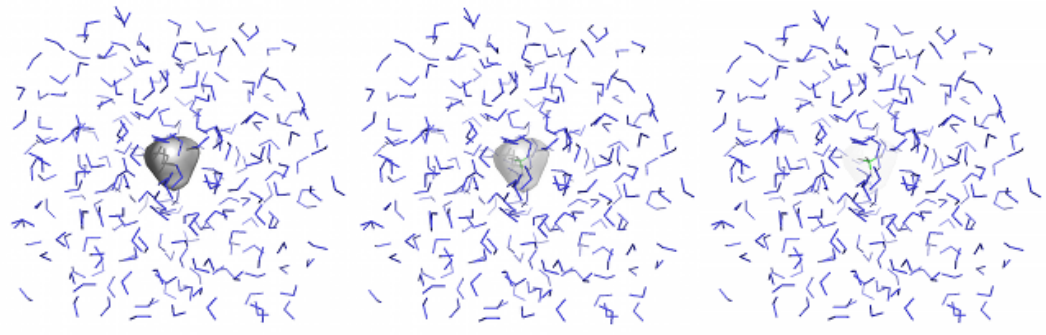


Figure. 24 transformation from state A (λ = 0) to state B (λ = 1)

This method requires two things to be specified apart from other simulation parameters like box size, time step, etc.

1. The two end states i.e. protonated and deprotonated state in our case
2. The pathway connecting the two which is specified in a form of a lambda vector

This lambda value is a sort of coupling parameter that indicates the level of change that has taken place between the two states i.e. the extent to which our Hamiltonian has been perturbed. We conduct simulations at different lambda values which allows us to plot a $\partial H/\partial\lambda$ curve from which ΔG can be calculated.

$$\Delta G = \int \langle \nabla H \rangle . \overrightarrow{d\lambda} \qquad (A.1)$$

which takes the following form if we use a finite difference scheme for numerical integration

$$\Delta G \approx \int \sum \langle \nabla H \rangle . \Delta\lambda \qquad (A.2)$$

GROMACS has a built in function to calculate the value of this integral (g_bar) which uses a method knows as BAR or Bennett Acceptance Ratio method.

GROMACS also allows to de-couple the system by using one parameter at a time like first decouple the electrostatic part and then the lennard jones part or all the parameters at once.

We have specified only the initial and final states so it uses linear interpolation to calculate the values of various parameters at intermediate lambda values as specified in our path way

Harmonic Potential

We use harmonic potential for bond and angle stretching terms.

$$V_b = \frac{1}{2}[(1-\lambda)k_b^A + \lambda k_b^B][b - (1-\lambda)b_0^A - \lambda b_0^B]^2 \qquad (A.3)$$

$$\frac{\partial V_b}{\partial \lambda} = \frac{1}{2}(k_b^B - k_b^A)[b - (1-\lambda)b_0^A + \lambda b_0^B]^2 + (b_0^A - b_0^B)[b - (1-\lambda)b_0^A - \lambda b_0^B][(1-\lambda)k_b^A + \lambda k_b^B] \quad (A.4)$$

The above expression is for bond stretching, though similar expression can be obtained also for bond-angle stretching

Coulombic Interaction

Between any two particles in the system

$$V_c = \frac{1}{4\pi\epsilon}\left[(1-\lambda)q_i^A q_j^A + \lambda q_i^B q_j^B\right] \quad (A.5)$$

$$\frac{\partial V_c}{\partial \lambda} = \frac{1}{4\pi\epsilon}[-q_i^A q_j^A + q_i^B q_j^B] \quad (A.6)$$

Lennard Jones Interaction

L-J interaction between any two particles is given by

$$V_{LJ} = \frac{(1-\lambda)C_{12}^A + \lambda C_{12}^B}{r_{ij}^{12}} - \frac{(1-\lambda)C_6^A + \lambda C_6^B}{r_{ij}^6} \quad (A.7)$$

$$\frac{\partial V_{LJ}}{\partial \lambda} = \frac{C_{12}^B - C_{12}^A}{r_{ij}^{12}} - \frac{C_6^B - C_6^A}{r_{ij}^6} \quad (A.8)$$

Though there are other interactions that can be modelled in this way but presently they are not part of our system so we will omit them here.

Appendix B: Our Library

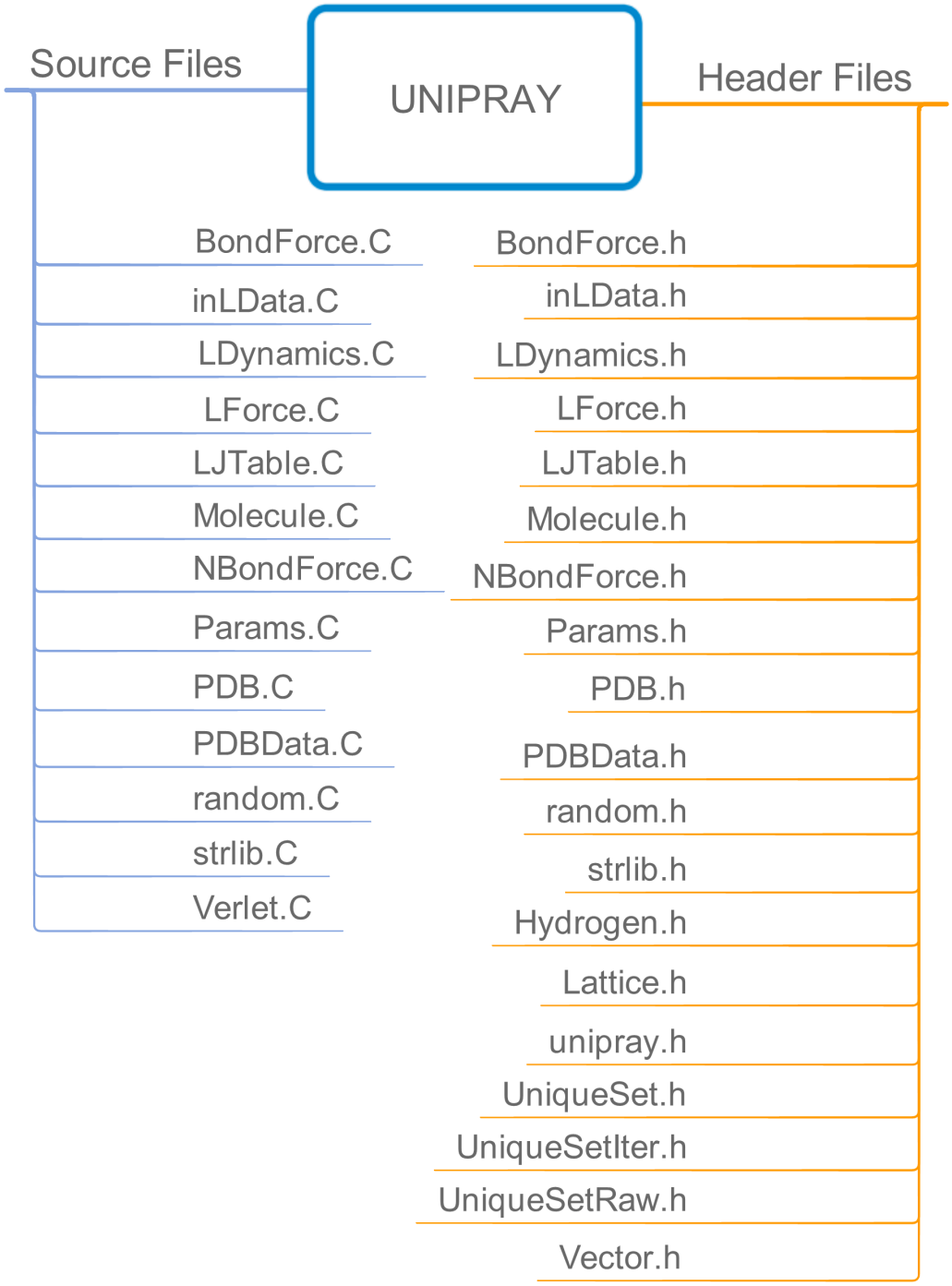


Figure. 25 Various Source and Header files included in our library.

In this appendix we provide a brief structure of our molecular dynamics library. We list the files (in alphabetical order) that are contained in our library and summary about what they do.

1. BondForce – It calculates all the bonded interactions of our system except the bonded interaction of our lambda particle. It contains separate function for calculating bond bending force and the angle-bending force. Presently we haven't implemented other bonded interactions but they will be added soon.
2. Hydrogen – It contains routines to check whether an atom is hydrogen atom or not. Presently, it's of no use, but it can be used later on when we want to constrain all the bond lengths involving hydrogen atom.
3. inLData – It reads the data regarding our titratable residue like the name of residue, number of atoms perturbed, pH of the system, etc. and it returns a linked-list containing all that information. This linked list also stores the value of our lambda parameter.
4. Lattice – It contains some helper functions which are used for doing various transformation on Vectors.
5. LDynamics – It contains the lambda dynamics part of our code. It advances value of our lambda variable over time using force values. It also initializes our data regarding titratable residue using functions from inLData. It can be considered as the main workhorse for our lambda dynamics code.
6. LForce – It returns the force on our lambda particle which is used by LDynamics::do_lambda_dynamics() to advances the value of lambda.
7. LJTable – It calculates and stores the value of Lennard Jones parameters C_{12} and C_6 from the values of ϵ and σ provided in PARAMS file so that they need not be calculated at every time.
8. Molecule – It reads in our PSF File and stores data in a molecule structure which contains detail about every molecule in our system, which atom is connected to which atom, etc.
9. NBondForce – It returns the Non Bonded forces for our system i.e. the Coulombic interaction and the Lennard Jones interaction.
10. Params – It reads in our parameters file containing force field information.
11. PDB and PDBData – These two files read in the PDB file using various routines implemented in them.
12. random – This is a custom random number generator which is used for generating velocity distribution and to draw uniform random numbers to be used with Andersen Thermostat.

13. strings and strlib – These file contain various helper functions related to string handling and processing. These help us in reading data from various files provided as input to our system.
14. unipray – This header file contains definition of all the structures and data types used in our library. It also contains macro defined constants used in our code.
15. UniqueSet, UniqueSetIter and UniqueSetRaw – These files contain helper functions for handling various structures and linked list used in our code.
16. Vector – It is also a helper header files which along with Lattice header files contains useful Vector operations and transformations.
17. Verlet – It houses the main function of our code. It contains the integrator, the thermostat, all the output routines implemented in it. It can be considered as the heart of our library.

Appendix C: File Types

This appendix provides a detail about the input file used and output file generated by our code.

lambda_groups file

Here all the titratable groups are located

name: The name of the titratable group. It must be of 4 letters though it can be different from that in PDB file.
residue_number: number of the residue.
initial_lambda: value of lambda between 0 and 1.
number_of_atoms: atom numbers as in PDB file, that are perturbed in our simulation

pK_a data file

In this file we list values of various parameters for each of the titratable group

residue: name of the titratable residue. Must be same as in the lambda_groups file
barrier: value of the barrier potential for this residue
const_a, ph_a, const_2, const_3: these corresponds to coefficient of the polynomial fit to the free energy data vs lambda obtained from the reference free energy simulation
const_b: $\ln 10 * R * T * \ pK_{a\_ref}$
ph_b: $-\ln 10 * R * T$

input.dat file

It contains following user specified parameters for simulation
nsteps: number of time steps to run for
timestep: the value of timestep in femto seconds
thermostat: Select Thermostat - 0 for Andersen and 1 for Berendsen
box_length: length of our simulation box in Angstroms

sampling_interval: number of time steps after which to save position and energy to output file

stepTcouple: number of steps after which to do a Berendsen Thermostat

switchdist: switching distance for non-bonded force

rvdw: cut-off distance for non-bonded interactions

tau: value of temperature coupling constant for thermostat

andersen_seed: random seed value for the random number generator

nu_T_lambda: number of time steps after which to couple temperature of lambda particle with thermostat

T_lambda: Temperature of lambda particle

m_lambda: mass of lambda particle

Tref: reference temperature of our system

equilSteps: number of equilibration steps to run for

stepCOMRemove: number of steps after which to remove centre of mass drift of the system

Appendix D: Analysis

Radial Distribution Function

The fluid state is characterized by the absence of any permanent structure, still there are some structural correlations that are measured to get some details about the molecular organisation.

In case of spatially homogenous system,

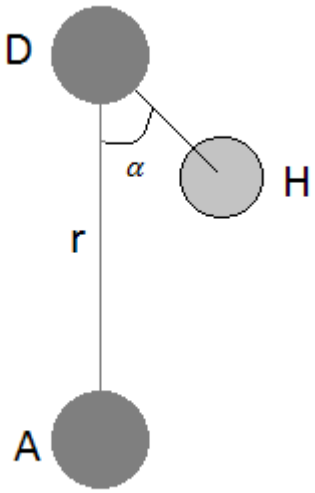
$$g(\boldsymbol{r}) = \frac{2V}{N_m^2}\langle \sum_{i<j} \delta(r - r_{ij})\rangle$$

$g(\boldsymbol{r})$ is the radial distribution function(RDF) which describes the spherically averaged local organisation around an atom.

Algorithm

- Pick a value of dr
- Loop over all values of r that you care about:
    1. Consider each particle you have in turn. Count all particles that are a distance between r and $r + dr$ away from the particle you're considering. You can think of this as all particles in a spherical shell surrounding the reference particle. The shell has a thickness dr.
    2. Divide your total count by N, the number of reference particles you considered -- probably the total number of particles in your data.
    3. Divide this number by $4\pi r^2 dr$, the volume of the spherical shell. This accounts for the fact that as r gets larger, for trivial reasons you find more particles with the given separation.
    4. Divide this by the particle number density.

Hydrogen Bonds



The value of r and α are calculated for every donor – acceptor pair and the calculated value is compared with following criterion to check for the existence of a hydrogen bond between them.

$$r \leq r_{HB} = 0.35nm$$
$$\alpha \leq \alpha_{HB} = 30°$$

Constant pH Simulations

For analysing data generated for constant pH simulations MATLAB® was used.

References

1. Smit, D. F. and B. Understanding Molecular Simulations. (2002).

2. Von Ballmoos, C., Wiedenmann, A. & Dimroth, P. Essentials for ATP synthesis by F1F0 ATP synthases. Annu. Rev. Biochem. 78, 649–72 (2009).

3. Murakami, S. Multidrug efflux transporter, AcrB--the pumping mechanism. Curr. Opin. Struct. Biol. 18, 459–65 (2008).

4. Hong, M. & DeGrado, W. F. Structural basis for proton conduction and inhibition by the influenza M2 protein. Protein Sci. 21, 1620–33 (2012).

5. Zhou and Ri-Bo Huang, G.-P. The pH-Triggered Conversion of the PrP^{c} to PrP^{sc}. Curr. Top. Med. Chem. 13, (2015).

6. Creighton, T. E. Proteins: Structures and Molecular Properties. (W.H. Freeman and Company, 1993).

7. Dobson, C. M. Protein folding and misfolding. Nature 426, 884–90 (2003).

8. J. Haas, E. Vöhringer-Martinez, A. Bögehold, D. Matthes, U. Hensen, A. Pelah, B. Abel, H. G. Primary Steps of pH-dependent Insulin Aggregation Kinetics Are Governed by Conformational Flexibility. ChemBioChem 10, 1816–1822 (2009).

9. Schmaljohann, D. Thermo- and pH-responsive polymers in drug delivery. Adv. Drug Deliv. Rev. 58, 1655–70 (2006).

10. Zhang, J.-L., Zheng, Q.-C., Chu, W.-T. & Zhang, H.-X. Drug Design Benefits from Molecular Dynamics: Some Examples. Curr. Comput. - Aided Drug Des. 9, 532–546 (2013).

11. H. Takahashi, S., M. Lira, L. & I. Córdoba de Torresi, S. Zero-Order Release Profiles from A Multistimuli Responsive Electro-Conductive Hydrogel. J. Biomater. Nanobiotechnol. 03, 262–268 (2012).

12. Donnini, S., Tegeler, F., Groenhof, G. & Grubm, H. Constant pH Molecular Dynamics in Explicit Solvent with λ -Dynamics. J. Chem. Theory Comput. 1962–1978 (2011).

13. Day, T. J. F., Soudackov, A. V., Čuma, M., Schmitt, U. W. & Voth, G. A. A second generation multistate empirical valence bond model for proton transport in aqueous systems. J. Chem. Phys. 117, 5839 (2002).

14. Warshel, A. & Weiss, R. M. An empirical valence bond approach for comparing reactions in solutions and in enzymes. J. Am. Chem. Soc. 102, 6218–6226 (1980).

15. Maupin, C. M., Wong, K. F., Soudackov, A. V, Kim, S. & Voth, G. A. A multistate empirical valence bond description of protonatable amino acids. J. Phys. Chem. A 110, 631–9 (2006).

16. Lill, M. A. & Helms, V. Proton shuttle in green fluorescent protein studied by dynamic simulations. Proc. Natl. Acad. Sci. U. S. A. 99, 2778–81 (2002).

17. Baptista, A. M., Teixeira, V. H. & Soares, C. M. Constant-pH molecular dynamics using stochastic titration. J. Chem. Phys. 117, 4184 (2002).

18. Wallace, J. a. & Shen, J. K. Continuous constant pH molecular dynamics in explicit solvent with pH-based replica exchange. J. Chem. Theory Comput. 7, 2617–2629 (2011).

19. Khandogin, J. & Brooks, C. L. Toward the Accurate First-Principles Prediction of Ionization Equilibria in Proteins †. Biochemistry 45, 9363–9373 (2006).

20. Khandogin, J. & Brooks, C. L. Constant pH molecular dynamics with proton tautomerism. Biophys. J. 89, 141–57 (2005).

21. Lee, M. S., Salsbury, F. R. & Brooks, C. L. Constant-pH molecular dynamics using continuous titration coordinates. Proteins Struct. Funct. Genet. 56, 738–752 (2004).

22. Wallace, J. a. & Shen, J. K. Charge-leveling and proper treatment of long-range electrostatics in all-atom molecular dynamics at constant pH. J. Chem. Phys. 137, (2012).

23. Chen, W., Wallace, J. a., Yue, Z. & Shen, J. K. Introducing titratable water to all-atom molecular dynamics at constant pH. Biophys. J. 105, L15–L17 (2013).

24. Baptista, A. M., Martel, P. J. & Petersen, S. B. Simulation of protein conformational freedom as a function of pH: constant-pH molecular dynamics using implicit titration. Proteins 27, 523–44 (1997).

25. Börjesson, U. & Hünenberger, P. H. Explicit-solvent molecular dynamics simulation at constant pH: Methodology and application to small amines. J. Chem. Phys. 114, 9706–9719 (2001).

26. J.E, M. & M.B, P. Molecular dynamics at a constant pH. Int. J. High Perform. Comput. Appl. 8, 47–53 (1994).

27. Kong, X. & Brooks III, C. L. λ-dynamics: A new approach to free energy calculations. J. Chem. Phys. 105, 2414–2423 (1996).

28. Warshel, A., Sussman, F. & King, G. Free energy of charges in solvated proteins: microscopic calculations using a reversible charging process. Biochemistry 25, 8368–72 (1986).

29. Kollman, P. Free energy calculations: Applications to chemical and biochemical phenomena. Chem. Rev. 93, 2395–2417 (1993).

30. Andersen, H. C. Molecular dynamics simulations at constant pressure and/or temperature. J. Chem. Phys. 72, 2384–2393 (1980).

31. Berendsen, H. J. C., Postma, J. P. M., van Gunsteren, W. F., DiNola, a & Haak, J. R. Molecular dynamics with coupling to an external bath. J. Chem. Phys. 81, 3684–3690 (1984).