Chapter 2

Molecular dynamics of proteins

2.1 Introduction

Theoretical description is the basis for comprehension of natural phenomena, as they are observed in experiments. Nevertheless in practice exact solutions for these descriptions are rather the exception than the rule and most theoretical results rely heavily on analytical and numerical approximations. Samples typically measured in experiments contain large numbers of molecules and cannot be theoretically studied by exactly enumerating all energy minima.

Computer simulation methods can be used instead of analytical approximations to generate representative conformations of a molecular system in equilibrium. Also in case of complex time dependent events, a simulation can provide a picture of the way in which a molecular system changes from one configuration to another.

Computer simulation is entirely based on physical theory, but the use of approximations is replaced by a more elaborate computational effort. The computation is not merely intended to generate an expected result, it is rather a virtual laboratory in which the behavior of a system can be described and predicted. In this respect computer simulation represents an intermediate level between experiment and theory [26, 27] .

Molecular dynamics (MD) is a computational methodology aimed at the solution of the N body problem, based on the classical analytical mechanics of Hamilton and Lagrange. A system of N interacting atoms is studied by solving the Newtonian equations of motion. Rigid molecules are described by the Euler angles. The system under study is then entirely classical, this means the quantum nature of atomic and molecular degrees of freedom is neglected. Therefore, molecular dynamics can be applied only to describe phenomena where quantum effects are not relevant or can be included as semi-classical corrections. Also relativistic effects are not taken into account, which means that the speed of light is infinite and all interactions are instantaneously propagated.

During an MD simulation, starting from arbitrary initial conditions, a trajectory in phase space is generated by numerically integrating the equations of motion. Each point in phase space

represents a set of positions and velocities of all N degrees of freedom forming the molecular system. The dynamic trajectory is a sequence of points generated at following time steps that entirely describes the motion of the molecular system. Time is the discrete independent variable used in the integration of the equations of motion.

Standard equilibrium MD corresponds to the micro canonical ensemble in statistical mechanics, but in certain cases properties at constant temperature or pressure are required. Thereby the equations of motion are modified and the trajectories are no longer solutions of the original Newton's equations, but are derived from more complex Hamiltonian functions (see section 2.6).

2.2 Ensemble and time averages

Equilibrium statistical mechanics provides the theoretical framework for many body systems, like for instance macromolecules. The concept of ensemble average is crucial, since it relates the microscopical energetics of a system in equilibrium to physically measurable quantities. A classical system of N interacting particles in equilibrium at constant temperature T is described in the canonical ensemble by the Boltzmann distribution. The ensemble average of a quantity $G(\vec{r^N})$, that is solely a function of the position of all N atoms $\vec{r^N}$, is expressed as phase space integral involving the potential energy $U(\vec{r^N})$:

$$\langle G \rangle = \frac{\int G(\vec{r^N}) \exp(-\beta U(\vec{r^N})) d\vec{r^N}}{\int \exp(-\beta U(\vec{r^N})) d\vec{r^N}} \tag{2.1}$$

Velocities are integrated out and $\beta = \frac{1}{k_B T}$ where k_B is the Boltzmann's constant.

Under certain conditions one can prove that a system in equilibrium is exploring the whole phase space when evolving with time. Therefore, following a single trajectory long enough is sufficient to visit the entire set of positions and velocities over which an ensemble average like the one in eq. (2.1) is calculated. This is the ergodic theorem [28] and implies that an ensemble average is equivalent to a time average over an infinitely long trajectory, which is defined as:

$$\langle G \rangle_t = \lim_{M \to \infty} \langle G \rangle_M = \lim_{M \to \infty} \frac{1}{M} \sum_{\mu=1}^{M} G(\vec{r^N}(t_\mu)) \tag{2.2}$$

where $\langle G \rangle_M$ is the average over M measurements of quantity G at subsequent time steps $\{t_\mu\}_{\mu=1}^{M}$. For a sufficiently large M the time average $\langle G \rangle_M$ is then according to the ergodic theorem a good approximation to the equilibrium ensemble average. Such time averages of energies and other quantities related to conformational properties are typically evaluated and analyzed during a molecular dynamics simulation.

2.3 Molecular dynamics of biomolecules

A theoretical study of structural and function properties of biological molecules at the atomic level is one of the goals of biocomputing. Although classical mechanics is in principle not suited for the simulation of such systems because of their quantum nature, a quantum mechanical treatment is still not feasible for molecules containing more than a few atoms. Therefore, one turns to classical MD simulation and uses empirical potential energy functions, which cannot take into account quantum effects. In this respect a crucial point is given by the parametrization of the energy function: parameters are adjusted in order to give agreement with quantum mechanical calculations on small compounds. Typical quantum phenomena like bond formation or disruption cannot be modeled via molecular dynamics: they require a more accurate treatment involving quantum calculations, which can be embedded in so called QM/MM simulations.
In the MD simulation of proteins and nucleic acids most used programs are CHARMM [29], AMBER [30], GROMOS [31]. In the following the main features of CHARMM, which was used throughout this work, are discussed. Most of the properties presented here are however common to all programs. The major packages have similar capabilities in terms of what molecular systems can be studied, the kind of simulations they allow to perform, and the tools for simulation data analysis [32].

2.3.1 Force field

The potential energy function describing the interaction between N atoms is a sum of energies, which depend on the atomic positions $\{\vec{r^N}\}$. One can distinguish between bonded and non-bonded interactions, so that the potential energy can be written as:

$$U(\{\vec{r^N}\}) = V_{\text{bonded}}(\{\vec{r^N}\}) + \sum_{i,j} V_{\text{non-bonded}}(\vec{r^i}, \vec{r^j}) \tag{2.3}$$

The bonded energy term describes the interaction among atoms involved in covalent bonds and consists of three different contributions. These are:

1. the bond energy, which for each pair of covalently bound atoms (i, j) describes deviations from the ideal bond length $b_0^{i,j}$ by means of a harmonic potential:

$$E_{\text{stretch}} = \sum_{(i,j)} K_B^{i,j}(\|\vec{r^i} - \vec{r^j}\| - b_0^{i,j})^2 \tag{2.4}$$

2. the bond angle energy defined for three subsequent atoms covalently bound and forming an angle which deviates from the ideal value θ_0. This is a harmonic potential in the angle:

$$E_{\text{angle}} = \sum_{(i,j,k)} K_\theta^{i,j,k}(\theta(i,j,k) - \theta_0^{i,j,k})^2 \tag{2.5}$$

3. the torsion angle potential which models the presence of steric barriers between four atoms separated by three covalent bonds. The motion associated with this term is a rotation, described by a dihedral angle and coefficient of symmetry n, around the middle bond. This potential is assumed to be periodic and is often expressed as a cosine function:

$$E_{\text{torsion}} = \sum_{(i,j,k,l)} K_{\phi}^{i,j,k,l} \left(1 - \cos(n\phi)\right) \tag{2.6}$$

So, the final expression for the bonded energy term is:

$$V_{\text{bonded}}(\{\vec{r^N}\}) = E_{\text{stretch}} + E_{\text{angle}} + E_{\text{torsion}} \tag{2.7}$$

Notice that all force constants and equilibrium values in the previous expressions depend on the atom species involved, therefore they are indexed.

The non-bonded potential energy is a sum of pair energy terms. Each one of the $N(N-1)$ pairs given in a system of N atoms provides two such terms: one is given by the van der Waals interaction energy and the other one by the electrostatic interaction. Both terms are function of the atom-atom distance $r_{i,j} = \|\vec{r^i} - \vec{r^j}\|$.

The van der Waals interaction between two atoms arises from a balance between repulsive and attractive forces. The repulsive force is present at short distances where the Pauli exclusion principle is relevant. The attractive force arises from fluctuations in the charge distribution in the electron clouds, which generates instantaneous dipoles. The attractive interaction is longer range than the repulsion, whereas at short distances the repulsive interaction becomes dominant. This produces a minimum in the energy, which is defined as follows:

$$E_{\text{vdw}}(\vec{r^i}, \vec{r^j}) = \sum_{i,j=1}^{N} \left(\frac{A_{i,j}}{r_{i,j}^{12}} - \frac{C_{i,j}}{r_{i,j}^{6}} \right) \tag{2.8}$$

Again the parameters depend on the chemical type of involved atoms.

From the electrostatic point of view, each atom i is considered to carry a point charge q_i located at the atomic position. Therefore, the electrostatic interactions between two atoms *in vacuo* is simply given by the Coulomb potential energy:

$$E_{\text{elec}}(\vec{r^i}, \vec{r^j}) = \sum_{i,j=1}^{N} \frac{q_i q_j}{\varepsilon_0 r_{i,j}} \tag{2.9}$$

and the total non-bonded energy function is then:

$$\sum_{i,j} V_{\text{non-bonded}}(\vec{r^i}, \vec{r^j}) = E_{\text{vdw}}(\vec{r^i}, \vec{r^j}) + E_{\text{elec}}(\vec{r^i}, \vec{r^j}) \tag{2.10}$$

Point charges are defined for each atom depending on the topology of the compound considered. A list of values concerning bond distances, angles, charges and atom types is given for all

amino acids and other relevant compounds in the topology file. All parameters involved in the potential energy function, like equilibrium distances and angles and van der Waals interaction parameters, are contained in the parameter set, which together with the topology file defines the force field. There are different versions of topologies and parameters available for biologically relevant molecules, like proteins, nucleic acids and carbohydrates. Each molecular dynamics program is usually equipped with its own force field, although force fields can be used independently with different programs. In the case of proteins and nucleic acids the most recent topology file developed for CHARMM is the CHARMM22 set [33], characterized by an explicit representation of all atoms, including non polar hydrogens. The atomic charges were derived from *ab initio* calculations of interactions between water molecules and small model compounds. In the preceding CHARMM19 topology file [34, 35] polar hydrogen atoms are explicitly represented, while non polar hydrogens are implicitly included in the binding partners: these are carbon atoms whose radius, mass and charge are modified in order to take into account the presence of hydrogens. Such a description, called extended model, has the advantage that the number of degrees of freedom and therefore the computation time are reduced.
Energy parameters are crucial in ensuring a correct representation of interactions. They are adjusted in order to give agreement with quantum *ab initio* calculations and tested on a number of different peptides and proteins in vacuo and crystals. The CHARMM22 parameter set is associated with the CHARMM22 topology file. The two files completely define the force field and are loaded together at the beginning of a computation.

2.3.2 Treatment of long range interactions

The most time consuming part of a molecular dynamics simulation is the calculation of the non bonded terms in the potential energy function. Namely a system of N atoms provides $N(N-1)$ pairs for which electrostatic and van der Waals forces should be computed. To speed up the computation, a cutoff distance can be introduced, such that all interactions between two atoms separated by a distance greater than the cutoff are ignored. There are several methods to achieve this, while respecting the smoothness of the energy function. For instance the shift cutoff method modifies the potential energy terms at all distances, such that at cutoff distance the interactions are zero. A drawback of this method is that all equilibrium distances are reduced. The switch cutoff method instead changes the interaction profile only over a predefined range of distances. The potential takes its usual value up to a first cutoff and is then switched to zero between the first and the second cutoff distance. This method has the disadvantage that strong forces arise in the transition region and is therefore not recommended, when using short cutoff distances.
By itself, the usage of cutoff distances may not strongly reduce the computation time during dynamics. This is because all distances between atom pairs should be calculated before deciding which ones are within the cutoffs and thus relevant for the calculation. The *non-bonded neighbor list* is a device by which atoms to be included in the non-bonded calculations are

automatically given at each step and only a periodical check and updating is required. In this way one avoids the computation of $N(N-1)$ distances at each step. The list contains all atoms within the cutoff distance of any other atom, together with all neighbor atoms, that are slightly further away than this cutoff. The distance used to calculate the neighbors, or neighbor cutoff, defines a sort of reservoir of atoms. It should be large enough to contain all atoms that in the time between two checks can enter the interaction range of a given atom. The list is regularly updated during the dynamics. A typical updating frequency is every 5 steps, which is a compromise between accuracy and efficiency.
Although the elimination of forces at large distances provides a significant improvement in calculation efficiency, one should point out that long range electrostatic interactions are indeed very important in biological molecules, as demonstrated in a number of experimental studies [36, 37]. Therefore, in many cases instead of using cutoff distances it is preferable to adopt one of the algorithms that have been developed in order to take into account the contribution of long range forces. The Ewald summation method is typically used for periodic systems like protein crystals [38], whereas for non periodic systems such as a molecule in solution a multipole expansion [39, 40] can be applied: this procedure distinguishes between a short range component of the electrostatic interactions, which is treated in the usual pairwise fashion, and a long range part approximated by a multipole expansion.

2.3.3 Boundary conditions

Boundary effects can play an important role in MD simulations especially in case of small numbers of atoms, with a large ratio between surface and volume parts [27]. The correct treatment of boundaries is crucial to MD simulations because it enables the calculation of bulk properties, which characterize a macroscopic molecular system. In the simulation of a protein in solution, it is of great importance that water molecules completely surround the protein, as it is in real systems. This configuration can be achieved either using a very large number of solvent atoms surrounding the protein and being confined by a cavity potential function, or by applying periodic boundary conditions. The latter procedure consists in putting the molecular system into a box, usually of cubic shape, and then generating a number of identical copies or images of the system adjacent to each face of the central box. The central molecular system evolves according to the Newtonian equations and interacts with the images, whose motion is a replica of the central dynamics. This method has the advantage that all atoms forming the real system, including those on the boundary, are surrounded by neighbors and the total number of degrees of freedom is still manageable. Moreover, the number of atoms in the central box is conserved, since for each particle leaving the box on one side the corresponding image enters the same box on the opposite side. In order to further reduce the number of degrees of freedom and therefore the computer time, other geometries than the cubic can be employed, like for instance the truncated octahedron, which was used in the longest protein folding simulation published so far [41].

2.4 Integrators: Verlet, leap-frog, velocity Verlet

Under the influence of a continuous potential energy function, the atoms moving according to a dynamics simulation give rise to a many-body problem that cannot be solved analytically. Therefore the equations of motion are integrated numerically using a finite difference method. The basic idea is that the integration is broken down into many small stages, each separated in time by a fixed time step δt. The total force on each atom in the configuration at time t is calculated as sum of all interactions with other atoms. Using the Newtonian equation the acceleration is obtained from the force. Then, combining it with the positions and velocities at time t, the new positions and velocities at the subsequent time step $t+\delta t$ are computed, forces are also calculated and the procedure is repeated to get new values at time $t+2\delta t$. The force is assumed to be constant during each time step.

There are different algorithms for integrating the equations of motion. Some of them assume that positions and dynamical quantities can be expressed as Taylor series expansions:

$$\vec{r}(t+\delta t) = \vec{r}(t) + \delta t\vec{v}(t) + \frac{1}{2}\delta t^2\vec{a}(t) + \dots \tag{2.11}$$

$$\vec{v}(t+\delta t) = \vec{v}(t) + \delta t\vec{a}(t) + \frac{1}{2}\delta t^2\vec{b}(t) + \dots \tag{2.12}$$

$$\vec{a}(t+\delta t) = \vec{a}(t) + \delta t\vec{b}(t) \tag{2.13}$$

where $\vec{r}(t)$, $\vec{v}(t)$, $\vec{a}(t)$ and $\vec{b}(t)$ indicate respectively position, velocity, acceleration and first derivative of the acceleration, all evaluated at time t. Acceleration is obtained at all times by means of the Newton's equation. In a system of N interacting atoms described by the potential energy function $U(\{\vec{r^N}\})$ defined as in eq. (2.3), the acceleration of i-th atom with mass m_i is

$$m_i\vec{a}_i = -\frac{\partial}{\partial\vec{r}_i}U(\{\vec{r^N}\}) \tag{2.14}$$

The most employed integration algorithm is the *Verlet integrator* [42], which uses positions and accelerations at time t, as well as positions at the previous step $t-\delta t$, to calculate the new positions at $t+\delta t$. From the Taylor expansion in eq. (2.11) follows for each particle:

$$\begin{aligned}\vec{r}(t+\delta t) &= \vec{r}(t) + \delta t\vec{v}(t) + \frac{1}{2}\delta t^2\vec{a}(t) + \dots \\ \vec{r}(t-\delta t) &= \vec{r}(t) - \delta t\vec{v}(t) + \frac{1}{2}\delta t^2\vec{a}(t) - \dots\end{aligned} \tag{2.15}$$

Adding these two equations gives, approximated to the fourth order in t:

$$\vec{r}(t+\delta t) = 2\vec{r}(t) - \vec{r}(t-\delta t) + \delta t^2\vec{a}(t) \tag{2.16}$$

Velocities do not appear explicitly in the integration algorithm. They can be estimated from the

positions:

$$\vec{v}(t) = [\vec{r}(t+\delta t) - \vec{r}(t-\delta t)]/2\delta t \tag{2.17}$$

One drawback of this integrator is that the positions at time $t+\delta t$ are obtained by adding a small term, $\delta t^2 \vec{a}(t)$, to the difference of two much larger quantities, leading to a loss of precision. In addition, at the starting step $t = 0$, the positions $\vec{r}(-\delta t)$ must be estimated using the Taylor expansion.

The *leap-frog* algorithm includes explicity the velocities and does not require the calculation of differences of large numbers. It uses the following relationships:

$$\begin{aligned} \vec{r}(t+\delta t) &= \vec{r}(t) + \delta t \vec{v}(t+\frac{1}{2}\delta t) \\ \vec{v}(t+\frac{1}{2}\delta t) &= \vec{v}(t-\frac{1}{2}\delta t) + \delta t \vec{a}(t) \end{aligned} \tag{2.18}$$

Here the velocities are always calculated at intermediate time steps between two subsequent position calculations. The disadvantage of this method is that positions and velocities are not synchronized, which for instance makes it impossible to calculate the total energy at a given time step.

Finally, the *velocity Verlet* algorithm gives positions, velocities and accelerations at the same step, thus with less efficiency but higher precision than the previous methods. Following equations are used:

$$\begin{aligned} \vec{r}(t+\delta t) &= \vec{r}(t) + \delta t \vec{v} + \frac{1}{2}\vec{a}(t) \\ \vec{v}(t+\delta t) &= \vec{v}(t) + \frac{1}{2}\delta t[\vec{a}(t) + \vec{a}(t+\delta t)] \end{aligned} \tag{2.19}$$

All these algorithms are based on equations derived from a hamiltonian function, so that the total energy, sum of potential and kinetic part, is in principle conserved and fully determined by the initial conditions when running a simulation. The stability of the integrator is measured in terms of deviation of the numerical trajectory from a reference analytical trajectory, where energy is fully conserved. Therefore, the stability is directly related to the ability of conserving energy and momentum throughout the dynamics. The stability also depends on the choice of the time step size: a stable algorithm permits a time step of 1-2 fs for protein simulations, which provides an efficient sampling of the phase space.

2.5 Set up of a (N, V, E) protein simulation

The standard molecular dynamics simulation is performed under conditions of constant energy E, number of particles N and volume V. This corresponds to the microcanonical ensemble in statistical mechanics. The trajectory propagation used for data analysis -so called production run- follows a series of preliminary operations.

The choice of the correct initial configuration is crucial in determining the success of a simulation. For simulations of systems at equilibrium it is recommended to choose a starting conformation that is close to the state which is desired to simulate. If an experimentally determined configuration is known, like for instance an X-ray or NMR structure published in the Protein Data Bank [43], this should be used. In case of a crystal structure some modeling is required, namely the missing hydrogen atoms must be correctly placed, using the information contained in the topology file.
Proteins are usually solvated, and the solvent must be modeled. In the next chapter the use of implicit solvent representations will be discussed. More often one chooses an explicit solvent model, namely a set of water molecules entirely surrounding the solute, filling a box and subjected to some boundary conditions, as described in the previous sections. Various models of water molecules are available in CHARMM force field, the TIP3 model [44] is widely used. The water molecules also need a starting configuration reasonably close to a realistic liquid at the desired temperature. This is usually obtained by running a preliminary molecular dynamics simulation of water alone at the required temperature, until equilibrium is reached.
After inserting the protein into the water box and deleting all water molecules that happen to be closer than a hydrogen bond distance, 2.6 Å, to the protein, the interactions between solute and solvent must be correctly modeled, in order to obtain a good starting configuration. This is achieved by means of an *energy minimization* according to the following scheme:

1. solvent molecules are energy minimized first while the solute is kept fixed;
2. the protein energy is minimized while constraining the solvent, in order to remove hot spots within the molecule;
3. finally both sets are minimized simultaneously, in order to reach a mutually optimized configuration.

At this point the dynamic integrator is invoked, time step size is defined, non-bonded interaction options are given and the dynamics can start. A (N, V, E) simulation is made of three steps, each one producing a segment of dynamic trajectory. After each step the dynamics is stopped and restarted, which means that information on positions and velocities of all particles at the end of one step is stored and used to initialize the subsequent step. The steps are listed here:

heating At time zero velocities are assigned according to a Maxwell-Boltzmann distribution corresponding to a temperature close to 0 K. After a number of steps, temperature is increased by 5 or 10 K, by rescaling the velocities (see next section). After a number of cycles the desired simulation temperature is reached.

equilibration At the beginning of this phase the system has reached the final temperature but it is usually far from equilibrium. During equilibration energy can flow within the system, bonded and non-bonded interactions are relaxed and hot spots are removed. Temperature is kept constant at the final value by means of a thermostat (see next section)

which rescales velocities if the temperature change exceeds a threshold. When the observed quantities like total energy and temperature assume a behavior characterized by small fluctuations and constant average (typically after tens or hundreds picoseconds) equilibrium is reached.

production At this point any temperature control is removed, the system is isolated and becomes strictly microcanonical. Now energy is conserved, while temperature is fluctuating. Various quantities can be monitored during time evolution. Velocities and coordinates can be stored for later analysis, when thermodynamic quantities are evaluated.

2.6 Temperature control

In (N, V, E) MD simulations it is required to control the temperature, namely during heating and equilibration. There may also be other situations where temperature control plays a fundamental role in MD simulation, like for instance (N,V,T) or constant temperature simulations, related to the canonical ensemble, or (N,p,T) simulations, known as constant pressure simulations.

The temperature of a molecular system is related to the average kinetic energy according to the equipartition theorem:

$$\langle K \rangle_{NVT} = \frac{1}{2} \sum_{i=1}^{N} m_i \langle \vec{v}_i^2 \rangle = \frac{3}{2} N k_B T \tag{2.20}$$

for a system of N atoms. Given the proportionality between squared velocities and temperature, the simplest way to control the temperature is to scale the velocities using an appropriate factor [45]. For instance, if temperature at time t is T_{curr}, and the required temperature is T_{req}, it is sufficient to multiply all velocities by:

$$\lambda = \sqrt{\frac{T_{req}}{T_{curr}}} \tag{2.21}$$

to obtain the required temperature. This procedure is known as *velocity rescaling*. Other scaling procedures use more complicated time-dependent scaling factors, to obtain an exponential decay of the molecular system towards the desired temperature [46]. These methods do not generate canonical averages in general: velocity scaling may artificially delay temperature equilibration differences between different parts of the system.

The *extended system* method, introduced to perform constant temperature MD simulations by Nosé [47] and Hoover [48] generates a canonical ensemble. In this approach a thermal reservoir is defined as an additional degree of freedom, in the molecular system. A brief description of the method, using the canonical representation, is given here.

Given a system of N interacting atoms, defined by the canonical variables $\{\vec{q}_i, \vec{p}_i\}$ that evolve

in time t according to equations of motion derived from the Hamiltonian:

$$H = \sum_{i=1}^{N} \frac{p_i^2}{2m_i} + U(\vec{q}_1, \ldots \vec{q}_N) \tag{2.22}$$

one defines a transformation into an extended virtual system $\{\vec{\rho}_i, \vec{\pi}_i\}$ dependent on time τ and containing an additional degree of freedom, s, by means of the following relationships:

$$\begin{aligned} s &= \frac{d\tau}{dt} \\ \rho_i &= q_i \\ \pi_i &= s p_i \end{aligned} \tag{2.23}$$

$$\pi_s = \frac{ds}{d\tau} \tag{2.24}$$

A new Hamiltonian for the virtual system, including s and its virtual mass M_s, is written as:

$$H^* = \sum_{i=1}^{N} \frac{\pi_i^2}{2m_i s^2} + U(\vec{\rho}_1, \ldots \vec{\rho}_N) + \frac{\pi_s^2}{2M_s} + (3N+1) k_B T \ln s \tag{2.25}$$

One can show that this Hamiltonian leads to a canonical ensemble. The equations of motion of the virtual system:

$$\begin{aligned} \frac{d\pi_i}{d\tau} &= -\frac{\partial H^*}{\partial \rho_i} \qquad \frac{d\pi_s}{d\tau} = -\frac{\partial H^*}{\partial s} \\ \frac{d\rho_i}{d\tau} &= \frac{\partial H^*}{\partial \pi_i} \qquad \frac{ds}{d\tau} = \frac{\partial H^*}{\partial \pi_s} \end{aligned} \tag{2.26}$$

can be written as functions of the real variables and of time t, and the coupling with the thermostat is then explicit. Defining

$$\zeta = \frac{1}{s}\frac{ds}{dt} \tag{2.27}$$

the equations of motion in terms of the real system become:

$$\begin{aligned} \frac{dp_i}{dt} &= -\frac{\partial U}{\partial q_i} - \zeta p_i \qquad \frac{d\zeta}{dt} = \frac{1}{M_s}\left[\sum_{i=1}^{N} \frac{p_i^2}{2m_i} - (3N+1) k_B T\right] \\ \frac{dq_i}{dt} &= \frac{p_i}{m_i} \qquad \frac{d\ln s}{dt} = \zeta \end{aligned} \tag{2.28}$$

Each state visited by the virtual system during the dynamics given by equations (2.26) corresponds uniquely to a state of the real system described in eq. (2.28). The virtual mass M_s is a parameter, which controls the energy flow between the real molecular system and the reservoir s. If M_s is very large the energy flow is slow and in the limit of infinite M_s no energy exchange takes place, thus conventional molecular dynamics is regained.

2.7 Stochastic molecular dynamics

The effect of solvent molecules on a solute consists of two contributions: on one hand the solvent molecules randomly interact with solute molecules by means of collisions in which energy is exchanged. On the other hand, the net effect of these collisions with solvent is a friction mechanism which slows down dynamics of solute and solvent. At thermodynamic equilibrium energy between solute and solvent flows such that both parts of the system fluctuate around the equilibrium temperature.

When performing a molecular dynamics simulation one can replace the explicit solvent molecules with a set of forces accomplishing the two effects described above. This provides an efficient way of controlling temperature. In fact such a solvent model plays the role of a thermal bath exchanging energy with the molecular system under consideration.

The above method [49, 50] relies on the stochastic *Langevin equation,* which governs the solute motion including solvent effects and therefore replaces the Newton equation. For a system of N interacting atoms with interaction given by the potential energy function $U(\{\vec{r^N}\})$ as in eq. (2.3), the Langevin equation for the generic i-th atom states:

$$m_i \frac{d^2\vec{r}_i}{dt^2} = -\frac{\partial}{\partial \vec{r}_i} U(\{\vec{r^N}\}) - \gamma_i m_i \frac{d\vec{r}_i}{dt} + \vec{R}_i(t) \tag{2.29}$$

The first term on the right is the deterministic force derived from the underlying potential energy function. To simplify the dynamics by eliminating uninteresting degrees of freedom, this potential function can be replaced by a *potential of mean force,* whose derivatives are obtained from ensemble averages. The second term represents the friction force, proportional to the velocity, which describes the damping effect of solvent. The coefficient γ_i represents the frequency of collisions and is the inverse of the velocity relaxation time. This time describes how long a particle moves before loosing memory of its initial velocity due to collisions. The intensity of the damping effect depends on the collision frequency: the more frequent are the collisions, the stronger is the damping effect. The third term on the right side is the stochastic force $\vec{R}_i(t)$, which describes the force, given by random interactions. The instantaneous value of each component of $\vec{R}_i(t)$ is taken from a Gaussian distribution $w(R_i)$ with zero mean, that is:

$$\langle R_i \rangle = 0 \quad ; \quad w(R_i) \propto \exp\left(-\frac{R_i^2}{2\langle R_i^2 \rangle}\right) \tag{2.30}$$

Friction and random force are related. Friction is dissipative. The average stochastic force vanishes, but in second order it increases the kinetic energy. Thus, cooling by friction compensates heating by noise, as governed by the *fluctuation-dissipation theorem* [51]:

$$\langle R_i(0) R_j(t) \rangle = 2k_B T \gamma_i \delta_{i,j} \delta(t) \tag{2.31}$$

where the correlation function of random force, given as ensemble average of the product

of two components $R_i(t)$ at different times, is related to temperature T and to the friction coefficient γ_i. The delta function in time indicates that random forces at different times are uncorrelated, as well as different components. K_B is Boltzmann's constant. The algorithm used to solve eq. (2.29) numerically in a MD simulation depends on the relationship between the size of the time step Δt and the collision frequency γ [49]. In case of mild interactions between solvent and solute, that is small friction coefficient, or, in other words, a propagation velocity relaxation time much longer that the step used for time:

$$\gamma \Delta t \ll 1 \tag{2.32}$$

the equation is integrated as follows, under the assumption that the force is constant during the time step:

$$\begin{aligned} \vec{r}(t+\Delta t) &= \vec{r}(t) + \vec{v}(t)\Delta t + \frac{1}{2}\vec{a}(t)\delta t^2 \\ \vec{v}(t+\Delta t) &= \vec{v}(t) + \vec{a}(t)\Delta t \\ \vec{a}(t) &= -\gamma\vec{v}(t) + \frac{1}{m}\left(-\frac{\partial}{\partial\vec{r}}U(\{\vec{r^N}\}) + \vec{R}(t)\right) \end{aligned} \tag{2.33}$$

For values of $\gamma\Delta t$ up to about 0.3 the Langevin algorithm implemented in CHARMM produces a stable dynamics and the fluctuation-dissipation theorem is satisfied.

2.8 The origins of CHARMM

The basis of modern physical research on biomolecules was set in the 1960s at the Weizmann Institute in Israel, in the group of Shneior Lifson [52]. Under his direction Ariel Warshel as PhD student and Michael Levitt, as a visiting student in 1967, wrote together the first computer program for simulating the properties of a molecular system from a simple potential energy function. There was considerable interest in developing empirical potential energy functions for small molecules. The new idea was to use a functional form that was able not only to calculate vibrational spectra, but also to determine the minimum energy structure. This program was called CFF [53] and provided energy and forces of a molecule. Levitt applied it subsequently to proteins, eventually obtaining the first energy minimization of an entire protein structure [54]. As Martin Karplus joined Lifson's group at the Weizmann Institute in the late 1960s, Chris Anfinsen was a regular visitor to the Weizmann, and many discussions on protein folding in solution inspired Karplus' work on this topic, for instance the *diffusion-collision* model for protein folding [18, 53]. With Szabo's molecular model on hemoglobin cooperativity [55], new questions arose about the energetics of structural transition from the unliganded to the liganded state of hemoglobin. It was time to develop a program that would make it possible to take a given amino acid sequence (e.g, that of hemoglobin) and a set of coordinates (e.g those obtained from the x-ray structure of deoxy hemoglobin) and to use this information to

calculate the energy of the system and its derivatives as function of the atomic positions. This program, developed by Bruce Gelin and Martin Karplus and inspired to the work of Michael Levitt [56, 54], was in fact PreCHARMM, although it did not have a name. It was successfully used in a range of applications [57, 58].

Given this program, the next step was to use the forces, originally computed for energy minimization, to solve Newton's equation and therefore calculate the dynamics. This task was performed by Andrew Mc Cammon when he joined Karplus' group.

Molecular dynamics was historically related to two kinds of problems. One direction concerned the study of simple chemical reactions, which finally led to modern semiclassical, QM/MM and quantum methods. The other was devoted to the computation of thermodynamical quantities in many particle systems, like the hard spheres liquid [59], the soft Lennard-Jones spheres representing liquid argon [60] and liquid water [61]. A molecular dynamics program suited for proteins had to base on sufficiently accurate potential functions, able to represent long enough (at least 10-100 ps) the features of an inhomogeneous system and sample the neighborhood of the native state. This was first achieved in 1975 with the simulation of BPTI [62], a small and stable protein, for which a relatively accurate x-ray structure was available. It was simulated *in vacuo* for 9.2 ps. This was enough to show that proteins are not rigid, but flexible structures, where internal motions can play a functional role. After this conceptual breakthrough, the evolution of molecular dynamics has been concerned with more detailed potentials, longer MD runs for improved statistics and description of more complex systems. The primary limitation of simulation methods yet remains, and it is that they are approximate. Experiment plays an essential role in validating the simulation methods, in other words the comparison with experimental data serves to test the accuracy of the calculated results and provides criteria for improving the methodology.