THESIS TITLE

SEBASTIAN SALASSI

Supervisor: Dr. Giulia Rossi

Advisor: Prof. Annalisa Relini

LIST OF ACRONYMS

FF Force Field

CG Coarse–Grained

COM Center of Mass

DOF Degree of Freedom

PW Polarizable Water

ESM Ewald Summation Method

PME Particle Mesh Ewald

FFT Fast Fourier Transform

PEF Potential Energy Function

PBC Periodic Boundary Condition

MD Molecular Dynamics

RMS Root Mean Square

INTRODUCTION

CONTENTS

1 INTRODUCTION TO MOLECULAR DYNAMICS

For macroscopic bodies, the motion of a system in time and space is governed by the classical equations of motion, say the Newton's law, while reducing time and space scales, quantum mechanics kicks in. Despite the latter statement, classical laws of motion have proved to be a good approximation also at the molecular level, as long as atoms are massive enough.

In order to predict the time evolution of a complete system, such as the biomolecular system we will treat in this thesis, Newton's equations of motion need to be integrated numerically. The necessity of a numerical integration arises from the complexity of the interactions involved in realistic systems, often nonlinear functions of positions and momenta of the particles making up the system, which makes it impossible to obtain an analytical solution for the equations of motion.

In the first part of this Chapter the laws of classical and statistical mechanics will be briefly summarized. Then we introduce the computational Molecular Dynamics (MD) method and analyze the main aspect of this technique.

1.1 REVIEW OF CLASSICAL AND STATISTICAL MECHANICS

1.1.1 Classical mechanics

The classical behavior of a system of N particles with mass m_i and coordinates $\vec{r}_1, \cdots, \vec{r}_N$, is described by the Newton's second law. Each particle in the system will experience the total force $\vec{F}_i$ and the then the second law for particle i yield

$$m_i \ddot{\vec{r}}_i = \vec{F}_i(\vec{r}_1, \cdots, \vec{r}_N) \tag{1.1.1}$$

The total force is defined as

$$\vec{F}_i(\vec{r}_1, \cdots, \vec{r}_N) = \vec{f}_i^{(e)}(\vec{r}_i) + \sum_{i \neq j}^{N} \vec{f}_{ij}^{(i)}(\vec{r}_i - \vec{r}_j) \tag{1.1.2}$$

where $\vec{f}_i^{(e)}$ is the external force acting on particle i; while $\vec{f}_{ij}^{(i)}$ is the inter-particle force that i exerts on j and *vice-versa*, in general it depends only on the distance between the particles. The equations (1.1.1) is refereed to as *the equations of motion* of the system; integrating them, with the sets of the *initial conditions* at start time t_0: $\vec{r}_1(t_0), \cdots, \vec{r}_N(t_0)$ and $\dot{\vec{r}}_1(t_0), \cdots, \dot{\vec{r}}_N(t_0)$, the positions and the velocities of all the particles in the system at any time it is known.

An other way to write the equations of motion is to use the particles momenta $\vec{p}_i = m_i \dot{\vec{r}}_i$ and then the equations (1.1.1) became

$$\frac{d\vec{p}_i}{dt} = \vec{F}_i(\vec{r}_1, \cdots, \vec{r}_N) \tag{1.1.3}$$

Obtain the full set of $6N$ functions $\{\vec{r}_1(t), \cdots, \vec{r}_N(t), \vec{p}_1(t), \cdots, \vec{p}_N\}$ give us a full description of the dynamics of the N–particle system. The set of functions above can be arranged in an ordered $6N$–dimension vector

$$\vec{x}_t = (\vec{r}_1(t), \cdots, \vec{r}_N(t), \vec{p}_1(t), \cdots, \vec{p}_N) \tag{1.1.4}$$

called *phase space vector* or the *microstate* of the system at a time t. All the possible microstate of a system generate a $6N$–dimension space called *phase space* of the system, indicated with Ω. $\vec{x}_t$ describe a particular trajectory of the phase space, i.e. the system's evolution is the motion of a phase space's point.

Let us suppose that all the forces acting on the N–particle system are conservative; this means that must exist a scalar function $U = U(\vec{r}_1, \cdots, \vec{r}_N)$ called Potential Energy Function (PEF), for witch

$$\vec{F}_i(\vec{r}_1, \cdots, \vec{r}_N) = -\partial r_{i\alpha} U(\vec{r}_1, \cdots, \vec{r}_N)\ \hat{e}_\alpha = -\vec{\nabla}_i U(\vec{r}_1, \cdots, \vec{r}_N) \tag{1.1.5}$$

so we have only to know the PEF of the system at any time and the initial conditions for solving Newton's Law.

The kinetics energy of the system, instead, is defined as

$$K(\dot{\vec{r}}_1, \cdots, \dot{\vec{r}}_N) = \sum_{i=0}^{N} \frac{1}{2} m_i \vec{r}_i \cdot \vec{r}_i \tag{1.1.6}$$

Supposing the system to be conservative, using the PEF and the kinetics energy, we can define a scalar function, called *Lagrangian* of the system

$$\mathcal{L}(\vec{r}_1, \cdots, \vec{r}_N, \dot{\vec{r}}_1, \cdots, \dot{\vec{r}}_N) = K(\dot{\vec{r}}_1, \cdots, \dot{\vec{r}}_N) - U(\vec{r}_1, \cdots, \vec{r}_N) \tag{1.1.7}$$

such that

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{r}_{i\alpha}}\right) - \frac{\partial \mathcal{L}}{\partial r_{i\alpha}} \tag{1.1.8}$$

this is a set of 3N equations: for each i, $\alpha = 1, 2, 3$. This equations are called *Euler–Lagrange equations of motion*. It is easy to show that substituting the definition of $\mathcal{L}$ we obtain the Newton's second law. The Euler–Lagrange equations is a sort of generator of the equations of motion.

With the definition of the Lagrangian we have

$$p_{i\alpha} = \frac{\partial \mathcal{L}}{\partial \dot{\vec{r}}_{i\alpha}} = m_i \dot{\vec{r}}_{i\alpha} \tag{1.1.9}$$

so we can express particles velocities as a function of particle momenta. Equations (1.1.6) and (1.1.9) let us to express the kinetics energy in the form

$$K = \sum_{i=0}^{N} \frac{\vec{p}_i \cdot \vec{p}_i}{2m_i} \tag{1.1.10}$$

For describing the system we can define an other scalar function, called *Hamiltonian* of the system

$$\begin{aligned}\mathcal{H}(\vec{r}_1, \cdots, \vec{r}_N, \vec{p}_1, \cdots, \vec{p}_N) = &\sum_{i=0}^{N} \vec{p}_i \cdot \dot{\vec{r}}_i(\vec{p}_i) + \\ &- \mathcal{L}(\vec{r}_1, \cdots, \vec{r}_N, \dot{\vec{r}}_1(\vec{p}_1), \cdots, \dot{\vec{r}}_N(\vec{p}_N))\end{aligned} \tag{1.1.11}$$

substituting (1.1.7) and using (1.1.10) the Hamiltonian of the system is nothing that

$$\mathcal{H} = K + U \tag{1.1.12}$$

or *the total energy of the system*. To obtain the equations of motion we have to solve the *Hamilton's equations*

$$\begin{aligned}\dot{r}_{i\alpha} &= \frac{\partial \mathcal{H}}{\partial p_{i\alpha}} \\ \dot{p}_{i\alpha} &= -\frac{\partial \mathcal{H}}{\partial q_{i\alpha}}\end{aligned}$$

Describing the system with the Hamiltonian formalism, in some cases, is most useful than Lagrangian one, first of all because the Hamiltonian of a system is directly related to a physical quantity well know: the total energy.

1.1.2 Statistical mechanics

With the classical mechanics described above we have a good and sophisticated machinery that allow us, knowing some informations about the system in exams, i.e. initial positions and velocities of all particles and how it interact each other, to solve completely the equations of motion in order to get the dynamics of the system at every time. So classical mechanics encode all the informations about the *microscopic* view of a system and, in principle, we can extract all the informations we want about the *macroscopic* proprieties of such system. The main task of such process is to obtain the thermodynamics proprieties of a system (temperature, pressure end so on) from the complete sets of positions and velocities of all particles and thus it is necessary to have a link between microscopic and macroscopic world. In principle this can be done, but if we consider a real system we should solve a set of 6N equations where N is of the order of the Avogadro number ($\mathcal{N}_A = 6.022 \cdot 10^{23}$ mol^{-1}); we can not think to solve a such number of equations analytically even if we consider to solve it numerically: that it is almost impossible. Thus the problem is to extract the macroscopic informations from the classical mechanics and to establish a well computable link between microscopic and macroscopic for obtain "easily" the thermodynamics informations required.

The solution of that problems comes from the *statistical mechanics* developed, principally, by Boltzmann and Gibbs. Statistical mechanics involves all the rules and methods through witch the microscopic world and macroscopic one are related to each other; this make also a stable derivation of thermodynamics from the microscopic proprieties: without that thermodynamics wold be only a phenomenological theory. The main contribution at the solution of the problems is to recognize that *a macroscopic observable of a system do not strongly depend on the complete dynamics of every particles in the system, but rather on an average that cancel out all the details of the microscopic features*. Now it is intuitively true; if we consider to set up an experiment, in principle, we can prepare the system in a specific microscopic state that generate a specific macroscopic state; certainly we can do the contrary and for sure, if the system is real, we do not find the same microscopic state! Then we can iterate the experiment and we find that for a specific macroscopic state of a system there exist some number of microscopic state that yield the same properties.

The most important idea, that make this concept practicable, is the concept of *statistical ensemble*. Based on the previous story a general definition of an ensemble is *a collection of systems subject all to a set of common interactions and sharing all the same macroscopic proprieties*. That concept make a rational basis of thermodynamics and a procedure for computing many macroscopic

observable. In more detail a N–particle system in a specific microscopic state is described by its microstate: $\vec{x} = (\vec{r}_1, \cdots, \vec{r}_N, \vec{p}_1, \cdots, \vec{p}_N)$ and hence each systems is describe as a point in the phase space, then *an ensemble is a set of points in the phase space that are subject to the constrain to be a part of the ensemble itself*. Each system evolve in time with the equations of motion, so the time evolution of an ensemble is described by the flow of this sets of points in the phase space according to the classical mechanics. Defined an ensemble we are able to compute, at every time, the macroscopic observable simple doing its average over all the systems in the ensemble. For doing this we have to know, at every time, witch microstates of the phase space are part of that ensemble. For this purpose we define the *ensemble distribution function* $\tilde{\rho} = \tilde{\rho}(\vec{x}, t)$; if $dx = dr_1 \cdots dr_{3N}, dp_1, \cdots, dp_{3N}$ is the infinitesimal phase space volume, then

$$\frac{1}{\mathcal{N}}\tilde{\rho}(\vec{x}, t)\ dx = \rho(\vec{x}, t)\ dx$$

where $\mathcal{N}$ is the total number of microstate in that ensemble; it is the probability that the microstate $\vec{x}$ at a time t is part of the ensemble. The function $\rho(\vec{x}, t)$ is instead the more convenient normalized distribution function. For definition of probability density must be

$$\int_\Omega \rho(\vec{x}, t)\ dx = 1, \qquad \rho(\vec{x}, t) \geqslant 0$$

Giving the ensemble distribution function, the ensemble average of a observable $A = A(\vec{x})$, at every time, is defined as

$$\langle A \rangle(t) = \int_\Omega A(\vec{x})\rho(\vec{x}, t)\ dx$$

For an ensemble at thermodynamic equilibrium the macroscopic state is fixed and so, if A is an equilibrium observable, it must be time–independent: this let us to define a scalar function of the Hamiltonian of the system such that

$$\langle A \rangle = \frac{1}{\mathcal{Z}} \int_\Omega A(\vec{x}, t) f(\mathcal{H}(\vec{x}))\ dx \tag{1.1.13}$$

where $\mathcal{Z}$, known as *partition function*, is specific for the ensemble in exams and it is defined as follow

$$\mathcal{Z} = \int_\Omega f(\mathcal{H}(\vec{x}))\ dx \tag{1.1.14}$$

In order to compute the partition function we need to specified the thermodynamic observables, called *control variables*, that characterize the ensemble it self; for definition of an ensemble at thermodynamic equilibrium that

control variables must be constant in time. The main ensembles used in statistical mechanics and the related control variables, are summarized as follow

- *microcanonical ensemble*: constant–NVE;
- *canonical ensemble*: constant–NVT;
- *isothermal–isobaric ensemble*: constant–NpT;
- *grand–canonical ensemble*: constant–μpT.

The averages computed with a different ensembles are equivalent in the so called *thermodynamic limit*, this is the *equivalence of ensembles*. Thus, it must be possible to change from one ensemble to another leaving averages unchanged.

We have defined the ensemble averages and how to compute it but we need also a link between that and the experimental values. When we measure a macroscopic observable A we prepare an experiment with *one only* system in a specific macroscopic state and we study its evolution in time. A is a function of time and phase space vector and it fluctuate over time due to the particles interactions. The measurement itself requires long time intervals compared to microscopic time scales, thus, when we measure an observable we take an *average over time*. If the time of average, in principle, is infinity then we have the "real" mean value of the observable

$$\overline{A} = \lim_{\tau \to +\infty} \frac{1}{\tau} \int_{t_0}^{\tau} A(\vec{x}_t)\, dt$$

In order for a comparison to be made, an identity between ensemble and time averages must be established. This link is provided by *ergodic theorem* and the *ergodic hypothesis*. A system is says to be ergodic if, for long period of time, all the microstate in the phase space with the same energy are equiprobablly accessible. Then the ergodic theorem says that the time and ensemble averages are equal *almost everywhere* in the space space. So we can write the follow identity

$$\overline{A} = \lim_{\tau \to +\infty} \frac{1}{\tau} \int_{t_0}^{\tau} A(\vec{x}_t)\, dt = \int_{\Omega} A(\vec{x}) \rho(\vec{x}, t)\, dx = \langle A \rangle (t) \tag{1.1.15}$$

The most important ensembles are the microcanonical and the isothermal–isobaric. In the follow we describe it briefly with particular attention to the isothermal–isobaric one, the most relevant for this thesis work.

1.1.3 Microcanonical ensemble

The microcanonical ensemble is composed by the systems whose number of particle (N), volume (V) and energy (E) are constant. Due to the constant energy it describe an Hamiltonian system so those for which

$$\mathcal{H}(\vec{x}) = E$$

this let us to define the partition function as follow

$$\mathcal{Z}_{NVE} = \frac{1}{N!h^{3N}} \int_{\Omega} \delta(\mathcal{H}(\vec{x}) - E)\ dx$$

where the normalization factor N! take into account the particle's indistinguishability and h^{3N} is for compatibility of statistical mechanics with quantum mechanics, it come from Heisenberg's uncertainty principle and it is the smallest phase space volume element.

1.1.4 Isothermal–isobaric ensemble

The isothermal–isobaric ensemble contains those systems with constant particle's number (N), pressure (p) and temperature (T). This is useful in many chemical, biological and physical systems since its proprieties are reported in conditions of standard temperature and pressure. For maintain the system at constant temperature and pressure it is necessary to couple it with an external *temperature bath* and a *pressure bath*. The first one can be considered simply a very big system at constant temperature with an high thermal capacity. The second can be idealized like a piston connected to the system: changing the volume for adjusting the pressure. The instantaneous work done by the system against the external piston is defined by pV, where V is the instantaneous system's volume. Then we have to correct the Hamiltonian of the system: $\mathcal{H}(\vec{x}) \to \mathcal{H}(\vec{x}) + pV$. The partition function is then defined considering the Boltzmann ensemble distribution function

$$\mathcal{Z}_{NPT} = \frac{1}{N!h^{3N}} \int_{0}^{+\infty} dV \int_{\Omega} e^{-\beta(\mathcal{H}(\vec{x})+pV)}\ dx \tag{1.1.16}$$

where $\beta^{-1} \equiv k_B T$, $k_B = 1.3806505(24) \cdot 10^{-23}$ JK^{-1} is the Boltzmann constant and the normalization factor is like above. The right thermodynamic potential for obtain all the macroscopic observable is the Gibbs free energy G defined by

$$G = -k_B T \ln \mathcal{Z}_{NpT} \tag{1.1.17}$$

it describe the maximum reversible work that may be performed by the system. The other thermodynamic quantities can be obtained by derivative

$$\mu = \left(\frac{\partial G}{\partial N}\right)_{pT} \qquad \langle V \rangle = \left(\frac{\partial G}{\partial P}\right)_{NT} \qquad S = \left(\frac{\partial G}{\partial T}\right)_{Np} \tag{1.1.18}$$

where S is the system's entropy.

1.2 MOLECULAR DYNAMICS SIMULATION

Molecular Dynamics (MD) is a set of techniques that allow us to prepare a "computer experiments" in with we have a virtual system set with some initial conditions, i.e. some microstate of all system's particles. Then, solving numerically the classical equations of motion we will able to know the time evolution of that system. Obviously that experiment is carried out using a model that approximates a real physical, chemical or biological system. Such virtual experiment approach has the advantage that many experiments can be easily set up with different initial condition and/or with different control parameters, such as temperature or pressure, in a straightforward and safer manner. The system's model contain all the informations for obtaining an approximation of the interactions between all system's particles or, most commonly, to compute the PEF in which the forces are derived by equation (1.1.5). With a numerical integrator, solving the equations of motion, a MD simulation generate a set of phase space vectors, a *trajectory*, at discrete times that are multiples of the fundamental time discretization parameter, called *MD time step*: Δt. Staring from an initial phase space vector $\vec{x}_0$, at each step, the forces are computed from the PEF, then the equation of motion are integrated and a new phase space vector $\vec{x}_{\Delta t}$ are generated, thus a new set of forces is computed and so on. In order to compute time averages we need to discretize equation (1.1.2) and the time integration is substitute with a summation over the collected data at certain time step $\Delta\tau = i\Delta t$, $i = 1, 2, 3, \cdots$. If $i > 1$ only a subset of the collected data is used to compute time averages. The formula become

$$\langle A \rangle = \frac{1}{M} \sum_{n=1}^{M} A(\vec{x}_{n\Delta\tau}) \tag{1.2.1}$$

where $M\Delta\tau$ is the total averaged time, of course must be $M \leqslant D/i$ if D is the total number of MD steps. An MD simulation can be summarized in the scheme in figure (1).

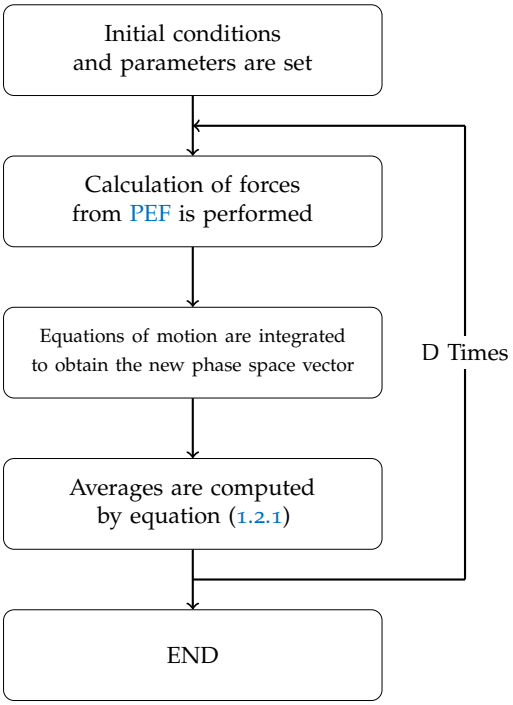


Figure 1: Schematic representation of an MD simulation

1.2.1 Initial configuration

Before a MD simulation can be performed it is necessary to select an *initial configuration*. The choice of it can be nontrivial and it depend on the complexity of the system, moreover, all the successive simulations depend on that initial configuration, so if that are wrong, maybe even the simulations produce wrong results. Then, carefully attention must be used to setting up the initial configuration.

Setting up an initial configuration mean to prepare a N–particle system and give all the particle's positions and velocities, i.e. give all the 6N coordinates of the initial phase space vector $\vec{x}_0$. The initial velocities can be extracted randomly from the Maxwell–Boltzmann distribution function at a specific system's temperature

$$f(v_i) = \sqrt{\frac{m_i}{2\pi k_B T}} e^{-\left(\frac{m_i v_i^2}{2k_B T}\right)}$$

moreover the random assignment algorithm must rescale all the velocities such that the total system's momentum $\vec{P} = \sum_{k=1}^{N} m_k \vec{v}_k$ is zero, this is equivalent to a Center of Mass (COM) motion removal. That is done because,

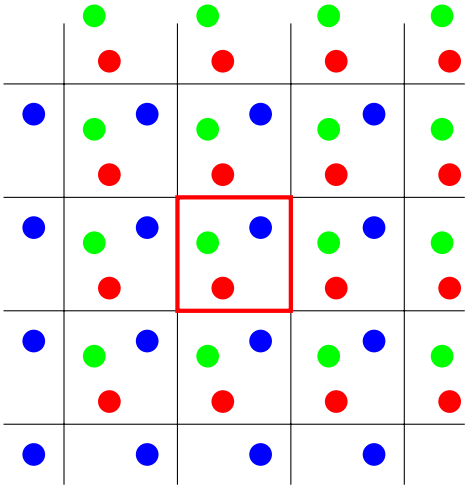


Figure 2: Schematic view of a two–dimensional box with PBC imposed. The central, red contoured, box is the simulation box and it si replicated along each side.

in general, the total force acting on the system $\vec{F} = \sum_{k=1}^{N} F_i$ is zero, then the COM motion is constant and to avoid a constant drift of the system in space this can be removed. Of course this is a constraint on the system and it must be take into account because it reduce the system's Degree of Freedom (DOF) by 3.

1.2.2 Periodic boundary conditions

In all experiment our systems is necessarily confined in to a box with some initial size. Even in a MD simulation the sample system is inserted into a *simulation box* whose shape can be differently chosen to better reproduce the symmetry of the simulated system. That box give us the trivial possibility to introduce a well defined reference system of coordinate in which respect the particles positions are assigned. Obviously we must not forget to correctly treat the *boundary conditions*. In order to avoid the treatment of surface effect and for considering only a infinite bulk system, a Periodic Boundary Condition (PBC) are imposed to the simulation box. This, let us also the possibility to simulate the system's bulk proprieties without considering a large number of particles. To give a better idea, in figure (2) is shown an example of a two–dimensional box with the PBC imposed. The central, red contoured, box is the simulation box, the idea is to replicate that box in space along each sides so that there are no surface particles nor walls in the central box. When a particle moves in the central box, all its images virtually move the same way in the copies of the box so that if a particle leave the virtual boundary of the central box, then, its nearest image enters the box and the number density of particles in the simulation box is conserved. This virtual movement of image particles is achieved adjusting the positions of the sim-

ulation box particles which have left the main box. For example, is one use a cubic box and a particle crosses its boundary in one direction, say the x direction, then its coordinate is corrected by subtracting (if leave the box in the positive direction) or adding the box side length parallel to x direction. Even if the most used shape is the cubic one, not all the shapes are accessible to impose the PBC, the most useful example is a spherical shape. When it is possible one have to use the most appropriate shape due to better describe the symmetry of the system, otherwise a closest approximation, compatible with PBC, must be used.

Even if PBC are used in a wide range of applications, it must be taken into account that imposing periodicity to a system may affect its properties. A clear limitation of the periodic cell is that it is not possible to achieve fluctuations that have a wavelength greater then the cells's length. This cause, obviously, the impossibility to sampling that vibrating modes. An other problem arises with the range of the inter–particles interactions: one have to choose carefully the size of the simulation box, or the number of particles if an NPT ensemble is used, to ensure that the smallest simulation box length is greater then the interaction range. This can be made easily for example with the Van der Waals interaction. On the contrary is a difficult and time consuming task to do the same with the electrostatic interaction that are treated with more sophisticated methods (this will be better explained in the section 1.3.4).

1.2.3 Numerical integrators

As we have seen above we need to solve numerically the equations of motion. Since the PEF is a continuos function of the phase spaces vector at a time t, the simplest way is to use the so called *finite difference* method. The basic idea is to expand the Newton's law in a Taylor series as follow

$$\vec{r}_i(t+\delta t) = \vec{r}_i(t) + \vec{v}_i(t)\ \delta t + \frac{1}{2m_i}\vec{f}_i(t)\ (\delta t)^2\ +\ o((\delta t)^3) \tag{1.2.2}$$

where we used the identities $\vec{v}_i(t) = \dot{\vec{r}}_i(t)$ and $m_i\ddot{\vec{r}}_i(t) = \vec{f}_i(t)$.

From this point, different algorithms have been developed. In the following we will describe in detail the most important, the *Verlet algorithm*, and an its implementation the *leap–frog algorithm*, which is the default used in our MD tools for this thesis work.

Verlet algorithm

The Verlet algorithm required the positions and the forces at a time t and the positions at a time $t - \delta t$ for calculate the positions at a time $t + \delta t$. Starting from equation (1.2.2) we can write

$$\vec{r}_i(t + \delta t) \simeq \vec{r}(t) + \vec{v}_i(t)\delta t + \frac{1}{2m_i}\vec{f}_i(t)\ (\delta t)^2 \tag{1.2.3}$$

$$\vec{r}_i(t - \delta t) \simeq \vec{r}(t) - \vec{v}_i(t)\delta t + \frac{1}{2m_i}\vec{f}_i(t)\ (\delta t)^2 \tag{1.2.4}$$

take its sum give us the new positions at a time $t + \delta t$

$$\vec{r}_i(t + \delta t) \simeq 2\vec{r}_i(t) - \vec{r}_i(t - \delta t) + \frac{1}{2m_i}\vec{f}_i(t)\ (\delta t)^2$$

The velocities does not compare in the equation above and can be obtained taking the difference of equation (1.2.3) and (1.2.4)

$$\vec{v}_i(t) \simeq \frac{\vec{r}_i(t + \delta t) - \vec{r}_i(t - \delta t)}{2\delta t}$$

Since positions are computed as differences this is a fourth order algorithm and the precision is up to $(\delta t)^4$, but are also a sum of a small term (of the order $(\delta t)^2$) to the differences of two larger terms, this may cause a loss of precision.

The main disadvantage is that the velocities at a time t are an output of the the calculation and not a part of the algorithm itself; moreover it is not self–starting because the algorithm required the positions at a time $t - \delta t$. At $t = 0$ we need a trick for obtain the previous inexistent positions. The trick is to use the equation (1.2.4) truncated at the first order: $\vec{r}_i(-\delta t) \simeq \vec{r}(0) - \vec{v}_i(0)\delta t$.

Leap–Frog algorithm

The leap–frog algorithm is a variant of the Verlet one and it is commonly implemented in many MD tools, that is our case. It compute the positions at a time t and the velocities at a time $t + \delta t/2$ from the forces at a time t and the velocities at a time $t - \delta t$. The main advantage, respect to the Verlet algorithm, is that it is self–starting because not require the positions at a time $t - \delta t$.

First it calculate the velocities at a time $t + \delta t/2$ as follow

$$\vec{v}_i(t + 1/2\delta t) \simeq \vec{v}_i(t - 1/2\delta t) + \vec{a}_i(t)\delta t$$

then the positions at a time $t + \delta t$ are computed

$$\vec{r}_i(t + \delta t) \simeq \vec{r}_i(t) + \vec{v}_i(t + 1/2\delta t)\delta t$$

The velocities at a time t can be calculated by

$$\vec{v}_i(t) \simeq \frac{\vec{v}_i(t + 1/2\delta t) + \vec{v}_i(t - 1/2\delta t)}{2} \tag{1.2.5}$$

An other advantage is that the velocities are part of the algorithm itself and that it does not require the calculation of the difference between two large number, with an increasing of precision. The obviously disadvantage is that the positions and velocities are not synchronized and it require equation (1.2.5) for calculating the velocities at a time t. The need to have velocities at the same time of positions, as for the Verlet algorithm, derived from the calculation of the kinetics energy contribution at the total energy: it must be computed with positions and velocities at the same time.

1.2.4 Neighbor list

1.2.5 Thermostats algorithms

1.2.6 Barostats algorithms

Berendesen algorithm

Parrinello–Rahman algorithm

1.3 EMPIRICAL FORCE–FIELD MODEL

As we have seen in the previous section MD provides a variety of tools for solving the time evolution of a N–particle system to obtain its dynamics. Due to the possibility to capture different length and time scales MD simulations can be used in a variety of systems, such a set of atoms, molecules or more complex system such as protein and macromolecules systems. In each of it systems, depending on the *interaction model* and its *parametrization*, we will be able to describe crucial molecular–level processes, such as hydrogen bond formation in organic molecules, which happen on the picoseconds time scale; or study slow processes such as the diffusion of massive colloidal particles, taking place on time scales of milliseconds if not seconds. When we study a soft or condensed matter system or, in general, a system composed by a large number of atoms (of the order of $N \gg 1000$), a crucial role play the *Born–Oppenheimer approximation*. It says that we can separate the motion

of the electrons by the motion of the atomic nuclei. That is done for integrating out the high frequency electrons' motions in order to remove some DOF. Moreover the main interesting processes of soft and condensed matter, ranging from protein folding to glass transitions, from surface diffusion to ligand–receptor binding take place on longer time scales and involve larger number of atoms. Father if we want to know precisely the dynamics of the electrons in the system we have to introduce quantum mechanical methods that are, even for a small number of particles (of the order of $N \sim 100$), to much computationally time consuming, thus the Born–Oppenheimer approximation is indispensable. In the following, when we speak about atoms or chemical moieties we refer to it as for nuclei coordinates only without considering electrons at all.

Nevertheless atoms or molecules interactions, such as bond formation, is mediated by the electrons interactions. Thus, for describe the dynamics of such a system with a classical MD tools and the Born–Oppenheimer approximation, it is necessary to develop an *empirical model of the inter–atoms interactions* that mimic correctly the "real" interactions. Since forces are derived from the PEF we need a model composed by the set of the simplest pairwise addictive potentials that mimic the inter–particles interactions. The model, the set of simulation parameters, such as the time step, the set of functional forms of the inter–particles interactions potential and its parameterizations are collected in to the so called empirical Force Field (FF). The meaning of *empirical* is that most of the functional forms of the inter–atoms interaction has no "first principle" justification and they are only an approximation to reality: there is not a correct way it are chosen as a compromise between accuracy and computational efficiency. Father it is necessary to stress out that a FF is a well defined single entity containing the simulation parameters, the functional models of the interactions and also its parameterizations (and the way to obtain it). All the parameters of a FF are in harmony to each other, change some parameters without retesting all the FF is not allowed because, maybe, one can destroy the whole FF.

For biomolecular applications exist two main classes of FFs: the *atomistic* FFs which the basic particles are atoms, and the *coarse–grained* FFs with the basic particles represent atom groups or a small chemical moieties. In this case, even the way to do the corse–graining of the atoms in the molecules, called *mapping*, is part of the FF itself. Different Coarse–Grained (CG) FFs can use different mapping method even with the same functional forms.

In the latter we add some other informations about FFs and describe the principal functional forms for modeling the inter–particles interactions and how to treat it in a MD simulation. For a more complete discussion about FFs the reader is addressed to the book by A. R. Leach [5]. While in the next

section we focus to the main CG FF used in this thesis work: the MARTINI CG FF developed by Marrink *et al.* [7].

PARAMETERIZATION In general the functional forms for potential interactions are common to all particles in the system, then the FF is completed by a set of empirical parameters that characterize the interaction between different types of particles, whether they are atoms or whole chemical groups. Interaction parameters are empirical in the sense that they are assigned based on the reproduction of a small set of a target properties of a small group of systems. These target properties can be derived from experimental measurements or from finer–level calculations or simulations. Nowadays, atomistic and CG biomolecular FFs come as "packages" of parameters and functional forms appropriate for the description of a large variety of chemical compounds in the liquid and solid phases.

TRANSFERABILITY As described above the parameterization of a FF involves a small set of test systems for which some set of target proprieties are reproduced. The main characteristic of a FF is the *transferability* that means the ability of the model to describe more different situations that differ from those used at the parameterization stage. Of course one would expect to be able to make some predictions for a bigger variety of systems and for other proprieties not used in the parametrization stage. Common faults of organic FFs concern, for example, phase transitions of organic compounds and phase transitions temperatures.

1.3.1 Inter–particles interactions

For biomolecular applications the inter–particles interactions potential are divided in to two main classes: the *bonded interactions* involving particles within the same molecules and the *non–bonded interactions* engaging all particles in the system and commonly are only the Van der Waals and the electrostatic interactions. The most common and general functional form for the PEF is the following one

$$\begin{aligned} U(\vec{r}_1, \cdots \vec{r}_N) = & \frac{1}{2} \sum_{\text{bonds}} \frac{1}{2} k_i^b (l_i - l_{i0})^2 + \frac{1}{2} \sum_{\text{angles}} k_i^a (\theta_i - \theta_{i0})^2 + \\ & + \frac{1}{2} \sum_{\text{torsions}} V_n (1 + \cos(n\omega - \gamma)) + \\ & + \sum_{i=1}^{N} \sum_{j>i} \left(4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right) + \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} \right) \end{aligned} \tag{1.3.1}$$

The first two terms in equation (1.3.1) are harmonic potentials which model respectively the energy contribution due to deviation from reference bond length l_{i0} and bond angle θ_{i0}. Together with the bond and angle elastic constants, k_{bi} and k_{ia} respectively, they constitutes the set of parameters for bond and angle contributions. The angle contribution involve a set of three particles in the same molecule. The middle line of equation (1.3.1) concern the energy contribution due to the bond's torsional change where ω is the torsional angle. It involve four particles in the same molecule and mimic the energy barrier needed to rotate the bond angle along the bond axis. γ is a phase factor, V_n qualitatively describe the energy barrier for each n–th components and n is defined as the number of minima for each components. The last line in equation (1.3.1) contains the energy contribution due to the non–bonded interactions: the Van der Waals modeled by a Lennard–Jones $12-6$ potential, fully characterized by the constants σ_{ij} and ϵ_{ij} proper for each particles' pair; and the electrostatic potential described by the particles' charge q. The non–bonded interactions involve obviously all particles in the system, but for particles belonging to same molecule they are computed only if they are separated by at least three bonds, i.e. if their interactions are not described by bonded terms. The various contributions described above are schematically represented in the figure (3).

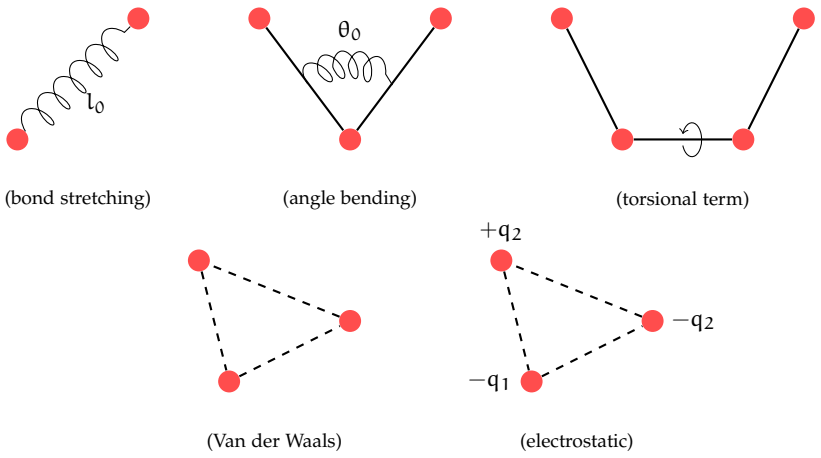


Figure 3: Schematic representation of the common inter–atoms interactions for biomolecular applications: bond stretching, angle bending, torsional term, Van der Waals and electrostatic interactions.

1.3.2 Non–bonded interactions

The bonded interactions, as we can see in equation (1.3.1), are at *fixed range*, meaning that it depended, for example, on equilibrium bond's length that is fixed. That is not for the non–bonded interactions because they depends on the inter–particles distance r_{ij} and they decay to zero as a power of r_{ij}^{-d}. Depending on the power order d compared to the system's size they are split into *short range* and *long range* interactions. Obviously the Lennard–Jones potential decay more rapidly then the electrostatic one; then, it is a short range interactions, while the electrostatic is a long range interactions.

Cut–off and shift method

Calculations of the non–bonded energy contributions is the most time consuming part of an MD simulations. Moreover, they are pairwise interactions and scale as N^2. Especially for the short range interactions, various method are developed in order to speed–up the simulations. The *cut–off* method is the most used to treat the short range interactions, but in same cases even the long range. Taking one particle, the general idea, is to evaluate the non–bonded interactions with all other particles that are closer to the first for a distance of r_c, called *cut–off*, otherwise the interactions is set to 0. That means that the new potential is of the form

$$v^*(r) = \begin{cases} v(r) & r \leqslant r_c \\ 0 & r > r_c \end{cases}$$

Doing that generate a discontinuity in potential itself and in the first derivative, i.e. in the forces; this is bad for energy conservation. A trick for solving the discontinuity of the potential and to improve the energy conservation is to apply even a *shift* of the potential value at r_c, since it is a constant it not affect the forces. We have

$$v^*(r) = \begin{cases} v(r) - v(r_c) & r \leqslant r_c \\ 0 & r > r_c \end{cases}$$

For solving even the discontinuity of the forces, that can cause some instability in a simulation, we need to consider a linear term proportional to the first derivative of the potential, such as

$$v^*(r) = \begin{cases} v(r) - v(r_c) - \left.\dfrac{dv(r)}{dr}\right|_{r_c} (r - r_c) & r \leqslant r_c \\ 0 & r > r_c \end{cases}$$

Although, the shift methods make the potential quite different from the "true" and this make difficult to retrieve the correct thermodynamics proprieties. Thus, even it can solve some instability, it must be carefully used.

An other powerful method is the *switch* method. The general idea is to consider two cut–off r_{c1} and r_{c2}. If $r \leqslant r_{c1}$ the "true" forms are used; while for $r > r_{c2}$ it is set to zero. For $r_{c1} < r \leqslant r_{c2}$ a *switching functions* is considered in order to *smoothly* switch the potential to 0.

It is important to stress out that even the method used to treat the short range interaction, as the cut–off radii and eventually the switching function are part of the simulation parameters that are still part of the FF. So they are interdependent with the model parameterization, and should not never changed without retesting some target properties. Obviously the same apply to the treatment of the long range interactions and the method used.

1.3.3 Van der Waals interactions

The Van der Waals forces are a set of interactions that are divided in to two main contribution: an attracting interactions and a repulsive one. The main contribution to both is due to quantum dynamics effect of the electrons cloud interactions through the Pauli exclusion principle; and the instantaneous electrostatic interactions, even if both atoms are neutral, such as dipole–dipole, induced dipole–dipole and induced dipole–induced dipole interactions, in a more rigorous description even they should be treated quantum mechanically. To the attractive contribution there are also the London dispersion forces that involves polar and non–polar atoms and is due to the instantaneous multipoles interactions, and the hydrogen–bonding that is due to quantum effect, instantaneous electrostatic interactions and an entropic contribution.

The common model to treat the Van der Waals interactions is to use a Lennard–Jones potential, the $12-6$ is the most common but even the $9-6$ is used depending on the system. The general forms for a $12-6$ Lennard–Jones potential is the following

$$v(r) = 4\epsilon\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right) = \frac{C_{12}}{r^{12}} - \frac{C_6}{r^6} \tag{1.3.2}$$

where $C_{12} = 4\epsilon\sigma^{12}$, $C_6 = 4\epsilon\sigma^6$ and r is the pairwise particles distance. ϵ is related to the absolute value of minimum while σ is relented to distance at the minimum: $r_{min} = 2^{1/6}\sigma$. That constant are proper for each particles pair type. The attractive contribution is due to the negative part proportional to

r^{-6}; while the repulsive one is due to the positive part proportional to r^{-12}. In figure (4) there is a example plot of the function (1.3.2) with $\epsilon = \sigma = 1$.

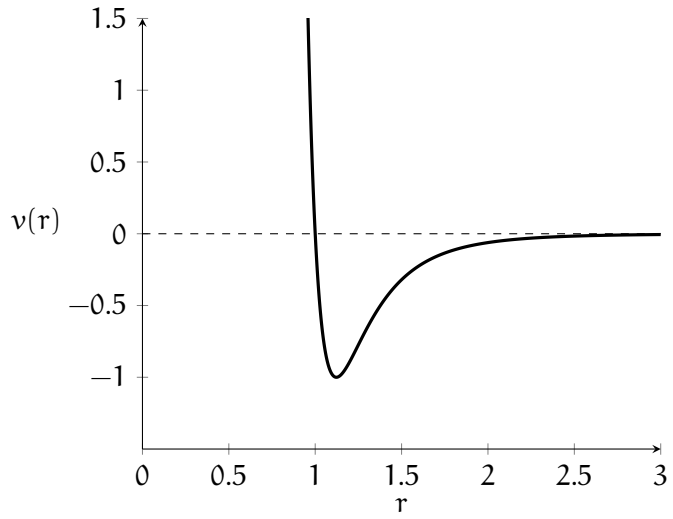


Figure 4: Example plot of a Lennard–Jones function with $\epsilon = \sigma = 1$.

The simplest and computational efficient way to treat a Lennard–Jones function, and in general all the short range interactions, is to use the cut–off method together with the shift or switch methods in order to obtain a continuous potentials and/or a continuous forces. As we can see from figure (4) the Lennard–Jones potential go to zero rapidly with distance: at $r \sim 2\sigma$ its value is less then 1% of the value in $r \sim \sigma$. A good choose for the cut–off is then of the order of $r_c \sim 2\sigma \div 3\sigma$.

1.3.4 Electrostatic interactions

One of the most important long range interaction is the electrostatic forces. Despite the long range characteristic, for purely computational efficient reason, most of the FFs for biomolecular applications treat them in the same way as a short range interactions by a cut–off method[1]. Of course this is an approximations and can lead to a serious issue in that proprieties or systems that strongly depend on the electrostatic interactions. Most issue due to a non well treatment of the Coulomb interactions are related to, for example, develop of a good polar solvent model (a good treatment of the electrostatic proprieties of water is really important or biological application), consider the interactions of charged particle with polar solvent, transport processes

1 In general, for computational reason, it is a common choice to consider the some cut–off for Van der Waals and electrostatic interactions.

of charged moieties, calculations of the electrostatic potential inside a macromolecules and so on. The loss of computational efficiency in the calculations of the electrostatic energy contribution is that it needs to take into account *all particles* in a system, but we can not forget the PBC: even all the infinity images of all particles need to be taken.

If we consider only the simulation box the energy contribution is

$$U = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\frac{1}{4\pi\varepsilon_0}\frac{q_i q_j}{r_{ij}} \tag{1.3.3}$$

where q_i and q_j are the charge, respectively, of particles i and j and r_{ij} is the distance between i and j. But we need also all image box. Supposing, for simplicity, that the box is a cubic shape of size L, then we can define a tern of integer numbers $(n_x,\ n_y,\ n_z)$, $n_i = 0, 1, 2, \cdots$ so that the position of all other image box, respect to the central simulation box, is $\vec{n} = L(n_x,\ n_y,\ n_z)$. Then the energy contribution became

$$U = \frac{1}{2}{\sum_{n_x,n_y,n_z}^{+\infty}}' \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{4\pi\varepsilon_0}\frac{q_i q_j}{\|\vec{r}_i - \vec{r}_j + \vec{n}\|} \tag{1.3.4}$$

where the prime indicate that for $\vec{n} = 0$ i.e. the energy contribution of the simulation box, we need to exclude the self interaction term: so in third sum must be $j \neq i$.

As describe above, a cut–off method is a good easy way solution for solving equation (1.3.3) and some time it reproduce good results. However, the increasing of computers power can lead to develop more rigorous methods to solve equation (1.3.4), even for very large system. The main problem is that the summation in equation (1.3.4) is *conditionally convergent*[2] and converges extremely slowly and need too terms that became too time consuming, especially for large system (of the order of $N \sim 3 \cdot 10^4$). The most important methods developed for solving that problem are based on the *Ewald Summation Method* (ESM). We shall describe that used in this thesis work: the Ewald summations method itself and the *Particle Mesh Ewald* (PME) method. For a more complete discussion about the advanced methods developed to treat the electrostatic interactions for biological applications the reader is addressed to the Review by Cisneros *et al.* [1] and for more technical detail to the book by D. Frenkel and B. Smit [4].

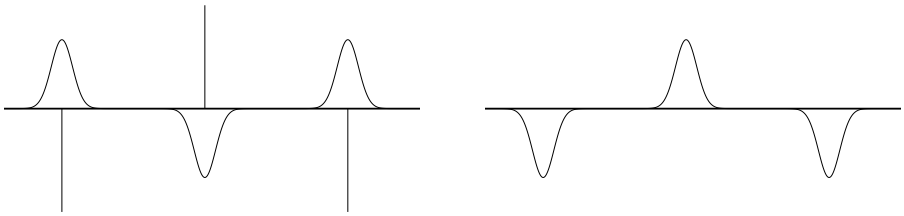
2 A conditionally convergent series contains both positive and negative terms such that the positive or negative term alone form both a divergent series. The sum of a conditionally convergent series depends on the order in which the positive and negative terms are considered.

Ewald summation method

The Ewald Summation Method (ESM) is the first method, introduced by Ewald *et al.* for a correct treatment of the electrostatic energy contribution in an ionic crystal that ca be extended in the electrostatic interactions of a periodic charge density. The basic idea is to split the summation in equation (1.3.4) in two series both rapidly convergent. The method is based on the following identity

$$\frac{1}{r} = \frac{f(r)}{r} + \frac{1 - f(r)}{r} \tag{1.3.5}$$

the trick is to choose a function $f(r)$ that will deal the rapid variation of the $1/r$ term for small r and the slow decay at long r; in that case the two series can rapidly converge.



(a) Real space **(b)** Reciprocal space

Figure 5: Schematic illustration of the Ewald Summation Method charge distribution: in (a) point charges (represented by vertical lines) and the neutralizing Gaussian charge distribution; in (b) the counteracts Gaussian distribution.

The ESM for electrostatic interactions is based, as illustrated in figure (5), considering each point–like charges in the system surrounded by a neutralizing charge distribution of equal magnitude but opposite sign that decay rapidly to zero. Simplify the notation for a one dimensional system, the simplest functional form is a Gaussian distribution centered in the position r_i of the point–like charge q_i, of the form

$$\rho_i(r) = \frac{q_i \alpha^3}{\pi^{3/2}} e^{-\alpha^2 (r - r_i)^2} \tag{1.3.6}$$

that obey the relation

$$\frac{q_i \alpha^3}{\pi^{3/2}} \int_{r_i - \epsilon}^{r_i + \epsilon} e^{-\alpha^2 (r - r_i)^2} \, dr \simeq q_i$$

where $(r_i - \epsilon; r_i + \epsilon)$ is a small interval around r_i. The energy contribution due to this set up, the point–like charge *and* the gaussian charge distribution, is given by

$$U_r = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}{\sum_{n_x,n_y,n_z}}' \frac{q_i q_j}{4\pi\varepsilon_0}\frac{\operatorname{erfc}(\alpha\|\vec{r}_i - \vec{r}_j + \vec{n}\|)}{\|\vec{r}_i - \vec{r}_j + \vec{n}\|} \tag{1.3.7}$$

where $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x)$ is the complementary error function and $\operatorname{erf}(x)$ is the error function given by

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^{+\infty} e^{-t^2}\,dt, \qquad \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^{x} e^{-t^2}\,dt \tag{1.3.8}$$

The point is that the summation involving the complementary error function in equation (1.3.7) is rapidly convergent and it need very few terms so that a cut–off method can be safety used. The rate of convergence depends on the α parameter, bigger is α more rapidly converge and shorter can be the cut–off. Thus the ESM use the erfcr as a $f(r)$ function in the equation (1.3.5). Of course since we have add a non physical neutralizing charge distribution in the system we must consider an other distribution of equal magnitude but opposite sign in order to restore the real charge distribution of the system. In view of the identity in the equation (1.3.5) that is done considering a distribution of the form $(1 - f(r))/r$ so, using equation (1.3.8), it is of the form $\operatorname{erf}(r)/r$. While the former it is compute in the *real space*, an other trick, is to consider the latter energy contribution due the counteracts charge distribution of the neutralizing charge, in the *reciprocal space*, thus considering its Fourier transform. That energy contribution is given by

$$U_f = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k_x,k_y,k_z} \frac{1}{4\pi\varepsilon_0}\frac{4\pi}{L^3 k^2} e^{-k^2/(4\alpha^2)} e^{i\vec{k}\cdot(\vec{r}_i - \vec{r}_j)} \tag{1.3.9}$$

where $\vec{k} = 2\pi\vec{n}/L$ are the reciprocal vectors. Even this reciprocal sum converge rapidly as the summation in equation (1.3.7); then a cut–off method can be safety used. Nevertheless, as opposite to the former, smaller is the α shorter can be the cut–off. Clearly it need a proper *balance* between the real and reciprocal space summation.

Since in equation (1.3.7) even the self interaction with each Gaussian is included we need to add another item for cancel out it; that is done by the self–term

$$U_{self} = -\frac{\alpha}{\sqrt{\pi}}\sum_{i=1}^{N}\frac{q_i}{4\pi\varepsilon_0} \tag{1.3.10}$$

Summarizing, the energy contribution of the electrostatic interactions by the ESM, is computed summing the equations (1.3.7),(1.3.9) and (1.3.10) to obtain

$$\begin{aligned} U = & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}{\sum_{n_x,n_y,n_z}}' \frac{q_i q_j}{4\pi\varepsilon_0}\frac{\operatorname{erfc}(\alpha\|\vec{r}_i-\vec{r}_j+\vec{n}\|)}{\|\vec{r}_i-\vec{r}_j+\vec{n}\|} + \\ & + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k_x,k_y,k_z} \frac{1}{4\pi\varepsilon_0}\frac{4\pi}{L^3k^2}e^{-k^2/(4\alpha^2)}e^{i\vec{k}\cdot(\vec{r}_i-\vec{r}_j)} + \\ & - \frac{\alpha}{\sqrt{\pi}}\sum_{i=1}^{N}\frac{q_i}{4\pi\varepsilon_0} \end{aligned} \tag{1.3.11}$$

The first line is the real space contribution while the second is the Fourier energy contribution. Since the last self–interaction term is constant it does not affect forces computations. The ESM offer a well defined method to properly treat the electrostatic interactions, nevertheless it is a quite expensive in term of computational resources. If α and the cut–off are constant, then the computation scales as $\sim N^2$; while if α and the cut–off are dynamically updated then it scales as $\sim N^{3/2}$ but this can lead to an incompatibility between the Van der Waals interactions cut–off and the electrostatic one compromising the efficiency gain. Despite we have described that method applied to the electrostatic interactions, it can be used, with some changes, to all long–range interactions and in general to all energy contributions that decay as r^{-d}, for example, even with the Van der Waals energy contributions.

For biomolecular applications most MD tools set an equal cut-off radius for both Van der Waals interaction the the real part of the Ewald summation (1.3.11) in order to achieve for both a scaling of the order $\sim N$. However in this way the computation of the reciprocal part in the Ewald summation (1.3.11) will be very inefficient and scales as $\sim N^2$. In order to increase the efficiency on the calculation of the Fourier transform a various of advanced methods can be used. It are all based on the use of the Fast Fourier Transform (FFT) methods. In this way the reciprocal part can scales as $\sim N \ln N$. Since FFT require discretize quantity, the idea of such method, called *Particle Mesh* is to consider the charge density spread on a mesh grid and then evaluate the electrostatic potential via solving the Poisson's equation using fast Poisson solver together with FFT method; that can be done, for example, exploiting the PBC in order to discretize and make periodic the Poisson's equation. Such algorithm include the *particle–particle particle–mesh* method, *Particle mesh Ewald* method, *Fast–Fourier Poisson* method and a recent methodology based on multi–scale mesh grid; the efficiency and accuracy of such mesh–based algorithms depend strongly on the way in which

the charges are attributed to mesh points, different for each methods. In the latter we describe the one used in this thesis work, the Particle Mesh Ewald (PME) method.

Particle mesh Ewald method

The Particle Mesh Ewald (PME) method developed by Darden *et al.* [2] are based on the ESM so the starting point is the equation (1.3.11). In which, as described above, the first part of the Ewals summation are computed in the real space together with the Van der Waals contributions using the same cut–off radius. While the reciprocal part are computed using FFT methods, in order to have a gain of performance. For doing this, first, we need to consider a grid mesh in which the Gaussian counteracts charge density are spread into. The basic idea, then, is to achieve the electrostatic energy solving the Poisson's equation through FFT methods. The efficiency and the accuracy depends on the way the charges are distributed into the grid. For doing this a *charge assignment function*, $W(r)$ is introduced such that, considering for simplicity a one dimensional system, the fraction of a charge at position r assigned to a grid point at position r_p is given by $W(r_p - r)$. Hence, if we have a charge density $\rho(r)$ then the charges at the grid point r_p are given by

$$q_M(r_p) = \int_0^L W(r_p - r)\rho(r)\, dr \tag{1.3.12}$$

where L is the box length and, if h is the grid spacing, $M = L/h$ is the number of mesh point. In figure (6) is schematically represented the charge assignment. The assignment function should have the following proprieties: should be an even function and should be normalized in such a way that the sum of the fractional charges equals the total charge of the system. Moreover the best accuracy is obtained with a dense grid in order to reduce as much possible the discretization of the charges density. However the computational cost increases as the number of grid points! So it clearly need a balance between efficiency and accuracy.

A nice way to solve the problem of charge assignment is to shift the problem in the discretization of the Fourier transform. It can be viewed as an interpolation problem. Consider the $e^{-i\vec{k}\cdot\vec{r}_j}$ term in the Fourier transform of the equation (1.3.9). In general $\vec{r}_j$ does not correspond to a mesh grid point, so that term is not part of a discrete Fourier transform. The idea, thus, is to interpolate it in terms of values of the complex exponential at the mesh points. Switching for simplicity to a one dimensional system, if the mesh grid has $M = L/h$ points, a particle coordinate r_j is located between mesh

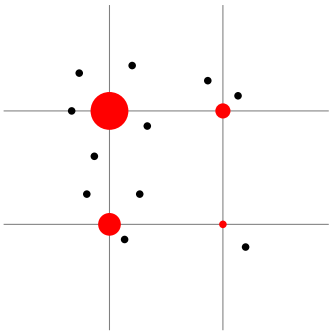


Figure 6: A schematic representation of the charge assignment. The black filled circles are a unit particle charge, while the red one, are the charge assigned to grid points. Bigger is the circle, more is the charge.

points $[r_j/h]$ and $[r_j/h]+1$ where $[]$ denote the integer part; thus a p–order interpolation of the exponential is of the form

$$e^{-ikr_j} \simeq \sum_{i=1}^{M} W_p\left(\frac{r_j}{h} - i\right) e^{-ikhi}$$

where W_p denote the interpolation coefficients. A p–order interpolation means that only the p mesh points nearest the r_j contributes to the sum. Assuming a point–like charge distribution the Fourier transform of the charge density is therefore

$$\rho_k \simeq \sum_i e^{-ikhi} \sum_j q_i W_p\left(\frac{r_j}{h} - i\right)$$

we can interpret the above expression as the discrete Fourier transform of the charge density

$$\rho(i) = \sum_j q_i W_p\left(\frac{r_j}{h} - i\right)$$

but using equation (1.3.12), it is nothing that the point–like charge distribution assigned to the mesh point i through the assignment function W_p.

We clearly see that the charge assignment problem is now shifted to the complex exponential interpolation. There are two main methods to make the interpolation: the *Lagrange interpolation method* and the *Euler SPLINE interpolation method*. The basic idea of the first is to use, as interpolating function, a polynomial function of degree $\leqslant (n-1)$ where n is the number of points to interpolate, that passes through all the n points, and it is constructed with a summation over the *Lagrange basis polynomials* as follow

$$P(x) = \sum_{i=1}^{n} y_i \prod_{\substack{k=1 \\ k\neq j}}^{n} \frac{x - x_k}{x_j - x_k}$$

where $(x_i; y_i)$ are the set of points to interpolate. The main disadvantage of that method is that, even if $P(x)$ is continuous everywhere, its derivative is not, thus can lead to some instability in MD simulations. The second method, that is the most used in MD tools, is based on the concept of *SPLINE interpolation*. Instead of using a unique interpolating function that passes to each points, the SPLINE method use a *piecewise polynomial function*, called SPLINE, in which each pieces are smoothly connected and optimized to interpolate a subset of the points. The Euler SPLINE method use the *exponential Euler SPLINE* that is constructed with the basis of the Euler n–degree polynomials $A_n(x; \lambda)$ generated by the following equation

$$\frac{\lambda - 1}{\lambda - e^z} e^{xz} = \sum_{n=0}^{+\infty} \frac{A_n(x; \lambda)}{n!} z^n$$

where λ is a complex parameter and z is a complex variable. The main properties of such SPLINE is that, it is $n - 1$ times analytic continuously differentiable and then can solve the instability problem of the Lagrange interpolation method. In literature the former Euler SPLINE method is refers to *smooth* Particle Mesh Ewald and the reader is addressed to the article by Essmann, Darder *et al.* [3] for more technical detail about the interpolation procedure.

Summarizing the PME method is implemented following the latter procedure

- By the interpolation of the complex exponential in the Fourier transform of the Ewald summation, the Gaussian counteracts charge distribution are spread into the mesh grid and a discretize Fourier transform are obtained;
- The Poisson's equations for the discretized charges are solved thought the FFT methods;
- The reciprocal energy contribution are obtained considering the inverse Fourier transform;
- Electrostatic forces are computed and assigned to the charged system's particles.

The main advantages of the PME algorithm is that the potential energy and forces are smooth functions of the particles positions, offers a good energies conservations, offers a very well balance between accuracy and computational efficiency since it scales as $\sim N \ln N$ and it is easily generalizable to interaction potentials that decay as r^{-d} such the Lennard–Jones potential. Nevertheless it not conserve very well the particles momentum due to Root

Mean Square (RMS) errors in the forces calculations that have to be cancel out by removing the forces averages.

1.3.5 Charge representation

Even if some methods are developed to speed up the electrostatic energy contribution such us the PME, one of the main problem, related to the electrostatic interactions, of the FFs for biomolecular applications remains the *charge representation*: the way that the charges of atoms or molecules are assigned to the system's particles. The problem arise from the representation of the negatively charged distribution due to the electrons cloud, that is, for instance a purely quantum effect, and the way that interactions of molecular electronic cloud can be achieved. Nevertheless that is crucial for a better description of the most electrostatic phenomena such us polarizability effect of molecules and polar solvent, solvation shell of charged ions, protein–ligands interaction, ion transport through polar and non–polar medium, self assembly processes and so on.

The mostly used solution is the *atom–centered "partial charge" approximation* in which the full charge density of the molecule is replaced by a fractional point–like charges assigned to every atom. But now, one has to decide how much of the molecular charge density should be assigned to each atom. Traditionally most FFs assigned to each atom of a molecule a fixed partial–charge. The most used procedure for extracting the partial–charges from molecular wave functions are based on fitting that atomic charges with the molecular electrostatic potential, computed with *ab initio* calculation such as *density functional theory*. The fitting procedure consist of minimizing the deviation between the electrostatic potential produced by the charge assigned and the molecular electrostatic potential. Such representation are believed to be a an important source of error in the electrostatic treatment. Moreover with fixed charge assignment it is more challenging to take into account such phenomena that involve a transfer of charge inside the molecule, as polarization effect. The use of off–centered charges and/or higher order atomic multiples can significantly increase the treatment of electrostatic but of course it is necessary a good balance between accuracy and performance since the electrostatic problem can arise rapidly a loss of efficiency sometimes without a really gain in the accuracy.

1.3.6 Polarization

Polarization refers to the redistribution of the electron density of a molecule in the presences of an external electric field, generated, for example, by

charged ions or an other molecule. The polarization effect generate non–addictive, attractive, inter– or intra–molecular interactions that rapidly became a many–body interactions with even the induced polarization effect. It is recognized as an important physical effect and an increasing number of studies shows that the lack this effect can be a serious limitation, particularly, for ionic systems and chemical process that involve different environment such us water–protein or water–lipid membrane environment. In MD simulations the polarization effect are included using either *implicit* or *explicit* method.

The implicit methods completely avoids the many–body calculation by including a mean polarization effect in the functional form of the interactions potential. The general idea is to surround all the simulation box by a transparent medium with a constant electrical permittivity ε_r. In this way the polarization effect is take into account considering a mean field theory and solving the Poisson's equation, for determine the electrostatic potential due to system's charges, by the substitution $\varepsilon_0 \to \varepsilon_0\varepsilon_r$. If that method give an incomparable gain in performance, must be carefully used. The main disadvantage is that the mean polarization effect is add to all system's particles. That can be interesting, for example, when our system is composed principally by water; but if the simulation box is composed by a different chemical environments such as water and lipids or other organic matters, the electrical permittivity can not be the same, that can lead to a serious incorrect results of the proprieties of the organic matters and maybe of the simulation etall.

The way to correct the above behavior is to use an explicit methods. As the name suggest, the polarization effect is take into account for every molecules or solvent in the system by an its proper model included in the FF. The general idea is to add to a molecule or atom some more internal DOF for take into account the movement of charges and/or split the point–like charge assigned, for example, to a chemical group, to a partial charge assigned to each particles of the chemical group itself. That can be done for every molecules or atoms in the system and thus it is an optimum for better describe systems with different chemical environments.

1.3.7 Coarse–Grained model

As we have introduced at the beginning of this section, for biomolecular applications exist two main classes of FFs: atomistic FFs and CG FFs. Since the atomistic model take into account all the atoms in a molecule it is obviously the most real and accurate FF. Nevertheless as the molecule's size also the DOF of the simulation increases and thus lead into a loss of performance.

Moreover, basically, the atomistic FFs are efficient until the physical proprieties can be properly sampled on a time scale of a few microseconds over a length scales of a few nanometers. As the time and length scales increase more and more time are need for carry out a complete simulation. Unfortunately many biological processes involving lipid membranes and other organic matters, including the interactions with synthetic compounds, take place on much longer time and length scales.

One possible solution is to *integrate out* some DOF, preserving those are relevant for the problem in exam: this procedure is so called *coarse–graining*. The basic units of a CG FFs are called *beads*, each representing a group of atoms or a well defined chemical moieties. The size of the group of atoms that is represented by a single bead determines the degree of coarsening of the FF. Even in this case, all the general features described above, apply: a functional forms need to be chosen and their parameterization need to be adjusted so as to reproduce the desired target properties. Moreover, in this case, even a *mapping* procedure should be define that is the first step in the development of a CG model: it establish a link between the atomistic model and the coarse–grain beads. There is not a unique or correct procedure to obtain the mapping because it depends on the desired coarse–graining level, on the time and length scales that one wants to correct sample and on the properties one wants to reproduce. For biological applications CG FFs are often designed to reproduced specific thermodynamics properties such surface tension, free energy of partitioning, free energy of hydration and so on, instead of, for example, the structural properties.

In general a CG FF are more computationally efficient then an atomistic FF for the following reason: the DOF of the system is reduced due to the CG procedure and a smaller number of interactions and forces has to be take into account; bead–bead interactions, which result from cancel out the fine structural details, are softer than the atom–atom interactions. Thus, vibrational modes are slower, and their sampling can be achieved using larger MD time steps than in atomistic simulations; softer interactions imply a smoother PEF, and this lead to a faster diffusion.

1.4 MARTINI: A COARSE–GRAINED FORCE–FIELD

MARTINI is a CG FF originally developed by Marrink *et al.* [7] for organic solvents and lipids and then extended to proteins [8], carbohydrates [6] and a broad class of polymers [9]. The original aim was to improve the description of the physics and chemical properties of the lipids membranes using a CG model. The power of the model is made immediately clear and soon the phi-

losophy behind was changed to developing a FF applicable to a broad range of organic applications providing a set of extensively calibrated building blocks for make a large variety of organic molecules without reparametrizing all the FF. That is possible, because, instead of focusing on the accurate reproducing of structural detail of a particular system, it is based on the reproducing accurately the interaction between polar and non–polar chemical compounds. That is the main target properties: the *partitioning free energy* between water and a large number of organic system, i.e. the free energy of transfer of these chemical moieties from polar and non–polar solvents. These building blocks are representative of the main chemical moieties in a organic system, that has been the guide for the mapping procedure.

1.4.1 Mapping

The mapping for the MARTINI beads, is based on a four–to–one scheme that groups for heavy atoms like C, S, O and so on, plus their associated hydrogen atoms, into a single interacting site. Consistently four water molecules is modeling with one MARTINI bead. There are four main beads type: polar (P), non–polar (N), apolar (A) and charged (Q). Each beads type has a number of subtypes for take into account a more accurate representation of the chemical nature of the moieties due to the specific atomistic structures. These subtype are distinguished by the hydrogen bonding capabilities: donor (d), acceptor (a), both donor and acceptor (da) and none (0) and/or by a degree of polarity: lowest polarity (1), $\cdots$, highest polarity (5).

1.4.2 Interactions potential

VAN DER WAALS INTERACTIONS The functional form describing the pairwise Van der Walls interactions is a Lennard–Jones $12-6$ potential as in equation (1.3.2). For the most beads the σ parameter is set equal to 0.47 nm except for the Q–C_1 and Q–C_2 interactions that are set to $\sigma = 0.62$ nm. This is consistent for reproducing the hydration shel when a charged beads (Q) is dragged into an apolar medium. The strength of the interactions is instead dived into ten levels, reported in table (1) The association of the interactions strength with the MARTINI beads is shown in figure (7).

ELECTROSTATIC INTERACTIONS The electrostatic charges are assigned using the atom–centered approximation, as described in 1.3.5. But, in this case, the charges are no more fractional and it are empirically assigned at the center of the beads trying to following as much possible the net charge of the associated chemical moieties. A special case is the water, modeled

Level	ϵ [kJ/mol]
O	5.6
I	5.0
II	4.5
III	4.0
IV	3.5
V	3.1
VI	2.7
VII	2.3
VIII	2.0
IX	2.0

Table 1: Interactions strength parameter (ϵ). The last one is for the special case of $\sigma = 0.62$ nm.

		Q				P					N				C				
	sub	da	d	a	0	5	4	3	2	1	da	d	a	0	5	4	3	2	1
Q	da	O	O	O	II	O	O	O	I	I	I	I	I	IV	V	VI	VII	IX	IX
	d	O	I	O	II	O	O	O	I	I	I	III	I	IV	V	VI	VII	IX	IX
	a	O	O	I	II	O	O	O	I	I	I	I	III	IV	V	VI	VII	IX	IX
	0	II	II	II	IV	I	O	I	II	III	III	III	III	IV	V	VI	VII	IX	IX
P	5	O	O	O	I	O	O	O	O	O	I	I	I	IV	V	VI	VI	VII	VIII
	4	O	O	O	O	O	I	I	II	II	III	III	III	IV	V	VI	VI	VII	VIII
	3	O	O	O	I	O	I	I	II	II	II	II	II	IV	IV	V	V	VI	VII
	2	I	I	I	II	O	II	II	II	II	II	II	II	III	IV	IV	V	VI	VII
	1	I	I	I	III	O	II	II	II	II	II	II	II	III	IV	IV	IV	V	VI
N	da	I	I	I	III	I	III	II	II	II	II	II	II	IV	IV	V	VI	VI	VI
	d	I	III	I	III	I	III	II	II	II	II	III	II	IV	IV	V	VI	VI	VI
	a	I	I	III	III	I	III	II	II	II	II	II	III	IV	IV	V	VI	VI	VI
	0	IV	IV	IV	IV	IV	IV	IV	III	III	IV	IV	IV	IV	IV	IV	IV	V	VI
C	5	V	V	V	V	V	V	IV	IV	IV	IV	IV	IV	IV	IV	IV	IV	V	V
	4	VI	VI	VI	VI	VI	VI	V	IV	IV	V	V	V	IV	IV	IV	IV	V	V
	3	VII	VII	VII	VII	VI	VI	V	V	IV	VI	VI	VI	IV	IV	IV	IV	IV	IV
	2	IX	IX	IX	IX	VII	VII	VI	VI	V	VI	VI	VI	V	V	V	IV	IV	IV
	1	IX	IX	IX	IX	VIII	VIII	VII	VII	VI	VI	VI	VI	VI	V	V	IV	IV	IV

Figure 7: Interactions strength associations matrix for the MARTINI beads type and subtype. Taken from [7].

as a P_4 bead, because it interact only with the Van der Waals interactions the polarizability effect is not very well described. For repair this lack it is used an implicit medium with a dielectric constant $\varepsilon_r = 15$. However, as we will see later in 1.4.5, for avoid the problems of the implicit medium described in 1.3.6, especially for the main aim of the MARTINI FF, the lipids membranes for that the dielectric constant in the hydroponic region is much more smaller, Yesylevskyy *et al.* [10] have developed a more sophisticated CG water model, called *polarizable water* (PW), for take into account a better water's polarization effect.

BONDED INTERACTIONS It include only a bond length and an angle harmonic contributions. The first use an harmonic potential as the first term in equation (1.3.1). With the same bond constant for all beads type: $k^b =$

1250 kJ/(mol nm^2) and an equilibrium distance of $l_0 = 0.47$ nm. Instead the angle contribution is modeled as a cosine–type harmonic potential

$$U = \frac{1}{2} k^a (\cos(\theta) - \cos(\theta_0))^2$$

and the parameters are: $k^a = 25$ kJ/mol and $\theta_0 = 180°$ for aliphatic chains; $k^a = 45$ kJ/mol and $\theta_0 = 120°$ for `cis` double bonds and $k^a = 45$ kJ/mol and $\theta_0 = 180°$ for `trans` unsaturated bonds. Moreover, specifically, for ring systems an improper dihedral angle harmonic potential can be used to prevent out of plane distortion. The form is

$$U = k_{id}(\theta_{ijkl} - \theta_0)^2$$

where θ_{ijkl} denotes the angle between the planes constituted between atoms i, j, k and j, k, l; k_{id} and θ_0 are, as usual, the force constant and the equilibration angle.

1.4.3 Simulation parameters

The MARTINI FF was originally developed using a group shifted cut–off scheme for both Lennard–Jones and electrostatic potentials with a cut–off radius of $r_c = 1.2$ nm. The Lennard–Jones was shifted from $r_s = 0.9$ nm to r_c while from $r_s = 0.0$ nm to r_c for the electrostatic potential. The neighbor list is constructed as $r_{list} = r_c$ and updated every 10 MD steps. Recently the more efficient Verlet cut–off scheme was tested and used with the MARTINI FF with a cut–off radius of $r_c = 1.1$ nm, a buffer tolerance of 0.005 kJ/(mol ps) and the neighbor list is set to be updated with a minimum value of 10 MD steps. Moreover, the treatment of the electrostatic interaction can be safety updated to the PME method together with the Verlet cut–off scheme. These was largely tested by Yesylevskyy *et al.* [10]. In this case the cut–off radius was set to $r_c = 1.2$ nm; the PME grid spacing was set to the lower bound of 0.12 nm and the interpolation was set to be a fourth-order. Moreover, with the use of the Polarizable Water (PW) the dielectric constant should be reduced to $\varepsilon = 2.5$. In all cases a time step up to 40 fs is suitable for a great number of applications, but 20 fs is the most powerful choice in terms of performance and accuracy balancing. It should be clear that changing that simulation parameters must be followed by a retest of the main properties of the MARTINI FF.

1.4.4 Parametrization

In order to parametrize the MARTINI CG FF a set of thermodynamics properties, obtained from MD simulations are compared and fitted among those experimentally measured. These properties are the *free energies of vaporization*, *hydration* and *partitioning* between water and a set of organic compounds such as hexadecane (H), chloroform (C), ether (E) and octanol (O). The free energy of hydration was obtained from the partitioning of the CG compounds between bulk water in equilibrium with its vapor. Similarity the free energy of vaporization was obtained considering a simulation box with the selected CG compounds in equilibrium with its vapor. From the equilibrium densities of the particles in both the phases the related free energy can be computed from

$$\Delta F = k_B T \ln\left(\frac{\rho_{vap}}{\rho_{bulk}}\right)$$

That simulations was performed in a canonical NVT ensemble.

Instead the partitioning free energy between water and an organic solvent was obtained in a NPT ensemble, considering a simulation box half filled by water and half by the organic solvents. Then a small fraction of the CG particles for that the partitioning free energy one want to compute, was placed in the simulation box. From the equilibrium densities of the particles in water ρ_{wat} and in organic solvent ρ_{oil}, the free energy of transfer can be computed from

$$\Delta G^{part}_{SW} = k_B T \ln\left(\frac{\rho_{oil}}{\rho_{wat}}\right)$$

where S indicate the organic solvent.

A summary of the results is reported in figure (**??**). As one can see the model has a bad performance for what concern the free energies of vaporization and hydration, which are too high with respect to the experimental data. Instead the partitioning free energy match very well. Thus the model is not very accurate for vapour–liquid system, but as long as one does not study that systems, the partitioning free energy is much important then the other.

2 | DUE

3 TRE

BIBLIOGRAPHY

[1] G. Andrés Cisneros et al. "Classical Electrostatics for Biomolecular Simulations". In: *Chemical Reviews* 114.1 (2014), pp. 779–814. DOI: http://dx.doi.org/10.1021/cr300461d. URL: http://dx.doi.org/10.1021/cr300461d.

[2] Tom Darden, Darrin York, and Lee Pedersen. "Particle mesh Ewald: An N · ln(N) method for Ewald sums in large systems". In: *The Journal of Chemical Physics* 98.12 (1993), pp. 10089–10092. DOI: http://dx.doi.org/10.1063/1.464397. URL: http://scitation.aip.org/content/aip/journal/jcp/98/12/10.1063/1.464397.

[3] Ulrich Essmann et al. "A smooth particle mesh Ewald method". In: *The Journal of Chemical Physics* 103.19 (1995), pp. 8577–8593. DOI: http://dx.doi.org/10.1063/1.470117. URL: http://scitation.aip.org/content/aip/journal/jcp/103/19/10.1063/1.470117.

[4] D. Frenkel and B. Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Computational science series. Elsevier Science, 2001. ISBN: 9780080519982. URL: https://books.google.it/books?id=5qTzldS9ROIC.

[5] A. R. Leach. *Molecular Modelling: Principles and Applications*. Pearson Education. Prentice Hall, 2001. ISBN: 9780582382107. URL: https://books.google.it/books?id=kB7jsbV-uhkC.

[6] Cesar A. López et al. "Martini Coarse-Grained Force Field: Extension to Carbohydrates". In: *Journal of Chemical Theory and Computation* 5.12 (2009), pp. 3195–3210. DOI: 10.1021/ct900313w. URL: http://dx.doi.org/10.1021/ct900313w.

[7] S. J. Marrink et al. "The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations". In: *The Journal of Physical Chemistry B* 111.27 (2007), pp. 7812–7824. DOI: http://dx.doi.org/10.1021/jp071097f. URL: http://dx.doi.org/10.1021/jp071097f.

[8] Luca Monticelli et al. "The MARTINI Coarse-Grained Force Field: Extension to Proteins". In: *Journal of Chemical Theory and Computation* 4.5 (2008), pp. 819–834. DOI: http://dx.doi.org/10.1021/ct700324x. URL: http://dx.doi.org/10.1021/ct700324x.

[9] Giulia Rossi et al. “Coarse-graining polymers with the MARTINI force-field: polystyrene as a benchmark case”. In: *Soft Matter* 7 (2 2011), pp. 698–708. DOI: 10.1039/C0SM00481B. URL: http://dx.doi.org/10.1039/C0SM00481B.

[10] Semen O. Yesylevskyy et al. “Polarizable Water Model for the Coarse-Grained MARTINI Force Field”. In: *PLoS Comput Biol* 6.6 (June 2010), pp. 1–17. DOI: 10.1371/journal.pcbi.1000810. URL: http://dx.doi.org/10.1371/journal.pcbi.1000810.